TOWARDS AN INTRINSIC DEFINITION OF ROBUSTNESS FOR A CLASSIFIER

Théo Giraudon, Vincent Gripon1 Matthias Löwe2 Franck Vermet3

1IMT Atlantique 2University of Münster 3University of Brest

ABSTRACT
Finding good measures of robustness – i.e. the ability to correctly classify corrupted input signals – of a trained classifier is an important question for sensitive practical applications. In this paper, we point out that averaging the radius of robustness of samples in a validation set is a statistically weak measure. We propose instead to weight the importance of samples depending on their difficulty. We motivate the proposed score by a theoretical case study using logistic regression. We also empirically demonstrate the ability of the proposed score to measure robustness of classifiers with little dependence on the choice of samples in more complex settings, including deep convolutional neural networks and real datasets.

Index Terms— robustness, deep neural networks

1. INTRODUCTION

Deep Learning is the golden standard for many challenges in the fields of machine learning and signal processing. Due to a large number of parameters, these architectures are able to absorb subtle dependencies from large datasets and to generalize decisions to previously unseen input signals. Among others, Deep Learning is state-of-the-art in classification in vision [1], playing complex abstract games [2], processing natural language [3], or decoding the brain activity [4]. However, despite achieving outstanding results, as typically measured by the accuracy on a validation or test set, Deep Learning architectures are likely to fail at correctly classifying corrupted input signals. As a matter of fact, in [5] it was shown first that it is possible to fool the system decision using a humanly imperceptible additive noise. This opened the way many contributions where increasingly efficient attacks have been introduced [6]. Such attacks could be considered artificial, since they require access to the network function. However, recent works [7, 8] have shown that some corruptions, where input signals are modified agnostically of the network function, are likely to dramatically lower the accuracy of the system as well.

In recent years, there has been an increasing number of works proposing mitigation strategies to enhance the robustness of trained deep classifiers. By robustness, most authors think of the ability of the network function to maintain a good decision for a certain radius around the training or test samples [9]. Examples of such strategies are listed in Section 2. Robustness is intrinsically tied with generalization: an ideal network function that would correctly predict the class of any input would by definition be robust. Let us point out that this statement implies that even in the case of a perfect classifier, robustness cannot be thought of uniformly for any input in a class domain. Indeed, a classifier defines a partition of the input domain in class regions. Hence some inputs are arbitrarily close to the boundary. It cannot be expected that such inputs have a large radius of robustness. On the contrary, inputs chosen at the center of class regions are likely to yield large such radii. This phenomenon is depicted in Figure 1.

In this work, we introduce a new definition of robustness which aims at simplifying the comparison of the intrinsic robustness of classifiers, by reducing the impact of the choice of samples it is estimated on. More precisely we introduce a new definition of robustness and show that it is independent of the choice of the samples in the case of the logistic regression. We also perform experiments using challenging vision datasets and corruption benchmarks to validate the interest of the proposed definition in a practical scenario.

2. RELATED WORK

Adversarial attacks as introduced by [5] consist of deformations with minimal norm, often imperceptible to the human eye, that causes the network to change its prediction with high confidence. To create these deformations, a complete knowledge about the decision function of the classifier, in particular its gradient, is needed. One could argue that these deformations need perfect knowledge about the network function and hence do not appear in practical life. However, [8] suggested black-box (i.e. agnostic of the network function) corruptions that considerably damage predictions of state-of-art neural networks on the CIFAR-10 and ImageNet datasets.

In order to reduce the efficiency of these attacks, several options have been studied. One of the most popular is adversarial training [10, 11, 12]. Apart from acting on the training set, another idea is to change the decision process, for instance through randomized smoothing [13, 14]. Finally, some authors choose to work on the structure of the network function by forcing its Lipschitz constant to be smaller than 1 ([15], [16]). We will study the impact of some of these strategies on our
Fig. 1: Figure (1a) represents the radii of robustness of \( x_1 \) and \( x_2 \) chosen in the same class region. Note that the maximum radius of robustness for \( x_1 \) is much larger than that for \( x_2 \) due to their respective distances to the boundary. This illustrates that not all samples may have the same radius of robustness. Figure (1b) shows two classes separated by two linear classifiers. Estimating robustness of the classifier as the mean of the radius of each sample, we obtain the orange dotted. Accounting for the difficulty of samples, using our definition 3.3, we obtain that the red solid line is the most robust. The latter is indeed more robust, as it yields a larger margin.

3. METHODOLOGY

Here we present our theoretical approach to robustness.

3.1. Properties of an ideal score of robustness

An ideal score of robustness \( R_s(f, D) \) for a trained classifier \( f \) on a dataset \( D \) would be one satisfying the followings properties:

(I) (Monotony) for two classifiers \( f_1, f_2 \) with the same test accuracy on \( D \), if all well-classified points are further, considering the Euclidean distance, from the decision boundary of \( f_1 \) than from the one defined by \( f_2 \), then \( R_s(f_1, D) \geq R_s(f_2, D) \).

(II) (Separability) if all well-classified points from \( D \) are at distance 0 from the decision boundary defined by \( f \), then \( R_s(f, D) = 0 \).

(III) (Subset Independence) \( R_s(f, D) \) does not depend on the samples from \( D \) is it evaluated on.

(III) is fundamental, because the main objective of testing robustness is to ensure that in applications the classifier will be robust, and not only on the considered validation set.

3.2. Mean-case and worst-case robustness

Suppose we have a labelled dataset \( D = \{x_i, y_i\} \), consisting of \( |D| \) elements with \( x_i \in \mathbb{R}^d \) and \( y_i \in \{1, \ldots, K\} \). We denote by \( f \) a trained classifier that associates any valid input \( x \) with a decision \( f(x) \in \{1, \ldots, K\} \). The robustness of \( f \) is usually defined using the concept of the radius of robustness:

**Definition 3.1.** The (maximum) radius of robustness of \( (x, y) \) with respect to the classifier \( f \) is defined to be the largest \( r \geq 0 \) such that any well-classified input is still well classified after any deformation of norm less than \( r \). Some took a probabilistic approach [18] and consider the maximum \( r \geq 0 \) such that there exists, with probability less than \( \alpha \), an adversarial attack of norm less than \( r \). We will show, both theoretically and empirically, that our certification is more robust than the classical deterministic one because way less examples are the mean-case robustness [6], defined as:

\[
R_m(D, f) = \frac{1}{|D|} \sum_{i=1}^{|D|} r(x_i, y_i),
\]

and the worst-case robustness, defined as:

\[
R_w(D, f) = \min_{i=1}^{|D|} r(x_i, y_i).
\]

It is easy to check that both of these definitions satisfy properties (I) and (II), but not (III). In practice, robustness is measured using a validation set \( D \), that is supposed to be a proxy to real-world data. Obviously, the worst-case robustness is very sensitive to the choice of \( D \), as removing the worst case would likely change the measure of robustness. But even mean-case robustness is sensitive to how examples in \( D \) are drawn. Consider the toy illustration of Figure 1a) for example, where we can see two samples that are likely to yield distinct radii of robustness, even for the most robust of classifiers.

One could argue that since classifiers are usually compared over the same validation set, there is no major concern about this observation. Let us point out that considering the validation set is biased towards easy or hard samples, it is very likely that the difference measured between two classifiers in terms of robustness is a very weak statistical test. Indeed, a similar radius of robustness for all samples is not necessarily desirable in practice, as some of these samples might approach the boundary between class regions. Problematically, it is not known in advance precisely which samples are close to the
boundary and which are far. In the next section, we show that the radii of robustness of samples can be directly linked to their corresponding loss for a trained classifier in the case of a logistic regression, so that it is possible to define a notion of robustness that is statistically robust to the sampling of the validation set \( D \) it is tested upon.

### 3.3. Robustness and Loss

We show how radius of robustness and loss can be dependent in the case of a logistic regression. Consider \( f \) to be a multinomial logistic regression with parameters \( (\beta_j, \beta^0_j)_{j=1, \ldots, K} \) where \( \beta_j \in \mathbb{R}^d, \beta^0_j \in \mathbb{R} \) and let \( D = \{ (x_i, y_i)_{1 \leq i \leq |D|} \} \) be a dataset where \( x_i \in \mathbb{R}^d \) and \( y_i \in \{1, \ldots, K\} \). Let \( \ell(x, y) \) be the cross-entropy loss for a sample \((x, y)\), that is

\[
\ell(x, k) = -\log \left( \frac{\exp(\beta_k \cdot x + \beta^0_k)}{\sum_{j=1}^{K} \exp(\beta_j \cdot x + \beta^0_j)} \right) .
\]

We then have the following:

**Proposition 3.2.** For every well-classified input \( x \) belonging to class \( k \), there exists \( m \neq k \) such that

\[
r(x, k) \leq \frac{-1}{||\beta_m - \beta_k||} \log \left( \exp(\ell(x, k)) - 1 \right)
\]

and

\[
r(x, k) \geq \frac{-1}{||\beta_m - \beta_k||} \log \left( \frac{1}{K-1} \left( \exp(\ell(x, k)) - 1 \right) \right),
\]

where \( r \) is defined for \( q = 2 \), i.e. the euclidean norm.

The proof of proposition (3.2) can be found in the long version of this paper [19]. When \( K = 2 \) and with \( \beta := \beta_2 - \beta_1 \) we immediately have:

\[
r(x, k) = \frac{-1}{||\beta||} \log \left( \exp(\ell(x, k)) - 1 \right)
\]

In the remaining of this work, we will denote \( g(t) = -\log(\exp t - 1) \) and we will always consider \( \ell(x, y) \) to be the cross-entropy loss of a sample \((x, y)\). For binomial logistic regression, a direct consequence of (6) is that if we sample elements in \( D \) using a probability measure \( \nu \) such that

\[
\nu(x, y) \propto \frac{1}{g(\ell(x, y))},
\]

the expected radius of robustness becomes \( \frac{\alpha}{||\beta||} \) where \( \alpha \) is the test accuracy of the classifier on \( D \). In the case of SVMs, the quantity \( \frac{1}{||\beta||} \) happens to be the margin between the classes, a notion obviously closely related to robustness. This motivates:

**Definition 3.3.** We call difficulty-aware robustness of a classifier \( f \) over a dataset \( D = \{ (x_i, y_i)_{1 \leq i \leq |D|} \} \) the quantity:

\[
\mathbb{E}_\nu \left[ (r(x_i, y_i))_{1 \leq i \leq |D|} \right],
\]

where \( \nu \) is a probability distribution on \( D \).

It is easy to check that irrespective of the probability distribution, the score satisfies properties (I) and (II) stated in section 3.1.

Our major statement now is the following: For any deep neural network using the cross-entropy loss \( \ell, r(x, y) \) increases approximately linearly with \( g(\ell(x, y)) \), hence the score

\[
R_\nu := \mathbb{E}_\nu \left[ (r(x_i, y_i))_{1 \leq i \leq |D|} \right] = \frac{1}{|D|} \sum_{i=1}^{|D|} \frac{r(x_i, y_i)}{g(\ell(x_i, y_i))}
\]

becomes a statistical quantity aiming at evaluating the slope of the line \( (g(\ell(x, y)), r(x, y)) \). Intuitively, the greater this slope is, the faster the radius of robustness increases as we move away from the class boundary to the center of the class. Implicitly, we set \( \nu \) as in (7). (6) guarantees that this statement is completely true for binomial logistic regression and Proposition 3.2 tells us it is pretty accurate for multinomial logistic regression. In the experiments section, we shall empirically confirm this statement for complex architectures and datasets, and see how this score predicts the quality of generalization in controlled synthetic cases and real-life datasets.

### 4. EXPERIMENTS

In the following experiments, we work with the \( L^2 \) norm, unless mentioned otherwise. In this section we aim at showing that the proposed measure of robustness (Definition 3.3) is indeed more robust to sampling of the dataset than commonly used counterparts, even when considering complex deep learning architectures and real datasets.

#### 4.1. Synthetic data

We first discuss our definition of robustness on completely controlled data and in the simple setup of logistic regression. To do so, we sample points from two two-dimensional isotropic Gaussian distributions of the same standard deviation and different mean and call this dataset \( D \). In this case, the asymptotically best classifier is the median line of the means of the two Gaussian distributions. We call this classifier the baseline and denote it by \( (\beta^*, \beta^{0*}) \). Now, we define a distance on linear classifiers. Let \( (\beta_1, \beta^{01}) \) and \( (\beta_2, \beta^{02}) \) defining two linear classifiers, then the distance between these two is defined as:

\[
d((\beta_1, \beta^{01}), (\beta_2, \beta^{02})) := ||\beta_2 - \beta_1||_2 + |\beta^{02} - \beta^{01}|.
\]

Figure (2) shows the evolution of the score \( R_\nu(f, D) \) as a function of the distance of the classifier \( (\beta, \beta^0) \) to the baseline, i.e. \( d((\beta^*, \beta^{0*}), (\beta, \beta^0)) \). For each distance \( d \), we randomly pick \( n \) classifiers \( (\beta, \beta^0) \) of distance \( d \) to the baseline and we average their scores \( R_\nu \). We observe some fluctuations around 1 at the beginning, since \( (\beta^*, \beta^{0*}) \) is only asymptotically best. Apart from that, the score \( R_\nu \) has the expected behaviour: It decreases smoothly as we go away from the asymptotically optimal classifier.
4.2. Correlation between Loss and Robustness

In the following, we investigate for several architectures and real-life data the link between loss and radius of robustness.

We train a multinomial logistic regression classifier on the MNIST dataset and, for 100 points in the validation set, plot their loss and their radii of robustness. They are depicted as blue dots on Figure (3a). To compute this radius at each step, we try in at most 5000 random directions of fixed norm to find a deformation that changes the class of the data point. If one is found we try with half the radius. We stop when the precision on the radius attains approximately 0.5. We also calculated the radius of robustness via adversarial attacks such as Projected Gradient Descent [20] and obtained similar results.

In the case of the logistic regression, a linear regression gives us that, with a coefficient of determination of 0.93, the radius of robustness is a linear function of $g(\ell)$. The same holds true for LeNet5 and for a ResNet on CIFAR-10: except for outliers, the radius still grows linearly with $g(\ell)$. Thus, we empirically verified that this behaviour holds in the case of a multinomial logistic regression, a CNN like LeNet5 and a Residual Neural Network trained with real data.

4.3. A dataset-independent score

For a logistic regression, the score $R_m$ is designed to be completely independent of the dataset samples it is evaluated upon. Let us stress if this property holds for complex neural networks on a complex dataset like CIFAR10. We first randomly pick 3000 samples from the validation set. Then, for a given trained network, we order them according to their loss: this is a pretty legitimate way of evaluating their difficulty. We then take on one hand the 300 easiest samples and in the other hand the 300 hardest ones. Finally we compute the mean score $R_m$ and the proposed score $R_\nu$ on both subsets and study their difference. In average and over 8 architectures we tested (c.f. [19]), the relative variation is 2.4 times larger with the mean score than with the proposed score. In all cases, our proposed criterion yielded a smaller relative variation. This shows empirically that the score $R_m$ is less dependent of the dataset than the score $R_\nu$ is and therefore that it goes towards verifying property (III).

### 4.4. Influence of adversarial training

Next, we study the effect of increasing the robustness of a classifier. We choose to consider the work in [20], where Madry and al. studied the effect of adversarial training on the MNIST and CIFAR-10 datasets. In Table 1, we present the results computed using 60 samples:

|                          | $R_m$ | $R_\nu$ |
|--------------------------|-------|---------|
| Naturally trained network | 2.4   | 0.25    |
| Adversarially trained network | 7.1   | 1.49    |

We notice for both scores an increase. This supports the fact that these scores do measure well the gain in robustness of adversarially trained networks for complex datasets.

### 5. CONCLUSION

In this paper, we pointed out the limits of using the mean radius to measure the robustness of a classifier. We showed that it is expected that some samples yield larger radii than others, depending on their difficulty and typicality. As such, we designed a simple score of robustness that accounts for this variation. We proved this score is theoretically independent of the choice of the samples for the simple case of a binomial logistic regression. We also derived multiple experiments with various datasets and neural network architectures to demonstrate that it still provides an interesting measure of robustness for more complex settings. Finally, the computational cost of this score is the same as the classical mean-case $R_m$ and worst-case $R_\nu$ as it is only needed to compute the losses of the samples on top of the radii, which consists in a negligible additional cost. In future work, it would be interesting to investigate whether sampling training inputs depending on their distance to the class boundary could result in more robust trained classifiers.
6. REFERENCES

[1] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, “Delving deep into rectifiers: Surpassing human-level performance on imagenet classification,” in Proceedings of the IEEE international conference on computer vision, 2015, pp. 1026–1034.

[2] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller, “Playing atari with deep reinforcement learning,” arXiv preprint arXiv:1312.5602, 2013.

[3] Tom Young, Devamanyu Hazarika, Soujanya Poria, and Erik Cambria, “Recent trends in deep learning based natural language processing,” IEEE Computational Intelligence Magazine, vol. 13, no. 3, pp. 55–75, 2018.

[4] Gaël Varoquaux and Russell A Poldrack, “Predictive models avoid excessive reductionism in cognitive neuroimaging,” Current opinion in neurobiology, vol. 55, pp. 1–6, 2019.

[5] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus, “Intriguing properties of neural networks,” arXiv preprint arXiv:1312.6199, 2013.

[6] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard, “Deepfool: a simple and accurate method to fool deep neural networks,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 2574–2582.

[7] Pin-Yu Chen, Huan Zhang, Yash Sharma, Jinfeng Yi, and Chunjui Hsieh, “Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models,” in Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security, 2017, pp. 15–26.

[8] Dan Hendrycks and Thomas Dietterich, “Benchmarking neural network robustness to common corruptions and perturbations,” arXiv preprint arXiv:1903.12261, 2019.

[9] Wenjie Ruan, Min Wu, Youcheng Sun, Xiaowei Huang, Daniel Kroening, and Marta Kwiatkowska, “Global robustness evaluation of deep neural networks with provable guarantees for the $l_0$ norm,” arXiv preprint arXiv:1804.05805, 2018.

[10] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy, “Explaining and harnessing adversarial examples,” arXiv preprint arXiv:1412.6572, 2014.

[11] Florian Tramèr, Alexey Kurakin, Nicolas Papernot, Ian Goodfellow, Dan Boneh, and Patrick McDaniel, “Ensemble adversarial training: Attacks and defenses,” arXiv preprint arXiv:1705.07204, 2017.

[12] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky, “Domain-adversarial training of neural networks,” The Journal of Machine Learning Research, vol. 17, no. 1, pp. 2096–2030, 2016.

[13] Jeremy M Cohen, Elan Rosenfeld, and J Zico Kolter, “Certified adversarial robustness via randomized smoothing,” arXiv preprint arXiv:1902.02918, 2019.

[14] Hadi Salman, Jerry Li, Ilya Razenshteyn, Pengchuan Zhang, Huan Zhang, Sebastien Bubeck, and Greg Yang, “Provably robust deep learning via adversarially trained smoothed classifiers,” in Advances in Neural Information Processing Systems, 2019, pp. 11289–11300.

[15] Moustapha Cisse, Piotr Bojanowski, Edouard Grave, Yann Dauphin, and Nicolas Usunier, “Parseval networks: Improving robustness to adversarial examples,” in Proceedings of the 34th International Conference on Machine Learning-Volume 70. JMLR. org, 2017, pp. 854–863.

[16] Haifeng Qian and Mark N Wegman, “L2-nonexpansive neural networks,” arXiv preprint arXiv:1802.07896, 2018.

[17] Aditi Raghunathan, Jacob Steinhardt, and Percy Liang, “Certified defenses against adversarial examples,” arXiv preprint arXiv:1810.09344, 2018.

[18] Tsui-Wei Weng, Pin-Yu Chen, Lam M Nguyen, Mark S Squillante, Ivan Oseledets, and Luca Daniel, “Proven: Certifying robustness of neural networks with a probabilistic approach,” arXiv preprint arXiv:1812.08329, 2018.

[19] Théo Giraudon, Vincent Gripon, Matthias Löwe, and Franck Vermet, “Towards an intrinsic definition of robustness for a classifier,” arXiv preprint arXiv:2006.05095, 2020.

[20] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu, “Towards deep learning models resistant to adversarial attacks,” arXiv preprint arXiv:1706.06083, 2017.