Detecting Adversarial Examples through Nonlinear Dimensionality Reduction

Francesco Crecchi [1], Davide Bacciu [1], Battista Biggio[2]

1- Universitá di Pisa - Dipartimento di Informatica
Largo Bruno Pontecorvo, 3, 56127 Pisa - Italy

2-Universitá degli Studi di Cagliari - Dip. di Ingegneria Elettrica ed Elettronica
Via Marengo, 09123 Cagliari - Italy

Abstract. Deep neural networks are vulnerable to adversarial examples, i.e., carefully-perturbed inputs aimed to mislead classification. This work proposes a detection method based on combining non-linear dimensionality reduction and density estimation techniques. Our empirical findings show that the proposed approach is able to effectively detect adversarial examples crafted by non-adaptive attackers, i.e., not specifically tuned to bypass the detection method. Given our promising results, we plan to extend our analysis to adaptive attackers in future work.

1   Introduction

Deep neural networks (DNNs) reach state-of-the-art performances in a wide variety of pattern recognition tasks. However, similarly to other machine-learning algorithms [1], they are vulnerable to *adversarial examples*, i.e., carefully-perturbed input samples that mislead classification [2].

The existence of adversarial examples questions the use of deep learning solutions in safety-critical contexts, such as autonomous driving and medical diagnosis. Despite several approaches have been proposed to protect a classifier from adversarial examples, many of them turned out to be ineffective in front of *adaptive* attacks that exploit knowledge of the defense mechanism [3, 4]. According to [5], we can categorize defense mechanisms that remain effective also against adaptive attackers into two main groups. The first ones include robust optimization techniques that can be exploited to reduce input sensitivity (or to increase the input margin) of the classifier; for example, *adversarial training* obtains a smoother decision function by retraining the model on adversarial samples generated against it. This requires, however, generating the attacks and re-training the classifier on them, which is very computationally demanding for large models such as state-of-the-art convolutional neural networks, and it does not provide any performance guarantee for adversarial samples crafted via a different attack algorithm. The second line of effective defenses includes explicit detection and rejection strategies for adversarial samples.

In recent years, different rejection-based countermeasures have been proposed. In [6] it is proposed to distinguish natural samples from adversarial ones exploiting a Kernel Density Estimator on the embeddings obtained from the last hidden layer of the neural network. The idea of a layer-wise detector has been pushed forward in [7] by providing multiple "detector" subnetworks placed at

different layers of the classifier which is intended to protect. Each subnetwork is trained to perform a binary classification task of distinguishing genuine data from samples containing adversarial perturbations. Such multilayer detectors have been showed to achieve relevant results in protecting against *static* adversaries, i.e. those who only have access to the classification network but not to the detector, and to significantly hardener the task of producing adversarial samples for *dynamic* adversaries, i.e. in which also the detector gradient is used by the attacker to craft adversarial samples.

In this paper, we build on the core concepts of [6, 7] by implementing a multilayer adversarial example detector based on t-SNE [8], a powerful manifold-learning technique mainly used for nonlinear dimensionality reduction. Our preliminary results are encouraging, showing that we can successfully detect adversarial examples crafted by non-adaptive attack algorithms. For this reason, we plan to further investigate and strengthen our method against adaptive attacks in the near future.

2   Detection of Adversarial Examples using t-SNE

High-dimensional data, such as images, are believed to lie in a low dimensional manifold embedded in a high dimensional space. In [2] it is speculated that adversarial examples come from "pockets" in the data manifold caused by the high non-linearity of deep networks. The adversarial inputs contained in such pockets occur with very low-probability which prevents them to be used during training and testing. Yet, these pockets are dense and so an adversarial example can be found near every test case but lies out-of-manifold.

In this paper, we support the so-called, *manifold hypothesis* by empirically demonstrating that is it possible to detect adversarial samples exploiting a non-linear dimensionality reduction method such as t-SNE to separate in- and out-of-manifold samples.

Mostly used for data visualization, t-SNE maps high-dimensional data into low-dimensional spaces while maintaining the relative distances between samples. As such, we used this technique to investigate the separability of natural and adversarial samples in the projected space. As shown in Fig.1, when the t-SNE projection is learned from a population of samples made of neural activations at a certain layer l for both natural and adversarial network inputs then it can very well separate them into two different clusters.

Detection Procedure. Figure 2 provides a representation of the detector architecture. Given an already-trained DNN classifier to be protected, we build on [7] by attaching a *Layer Detector* (LD) to potentially all classifier layers. Each LD takes as input $h_l(x)$, the internal representation of a given input sample x extracted by the classifier at layer l and outputs a vector of probability scores $p_l(x)$ for x to be a natural sample of each of the target classes. The predictions of each LD form a data series that is fed into the $Multilayer Detector$ (MLD) which is responsible to capture trends that identify adversarial examples with respect to natural data. In case of detection of an adversarial attack ($y_{detector} = 1$), the
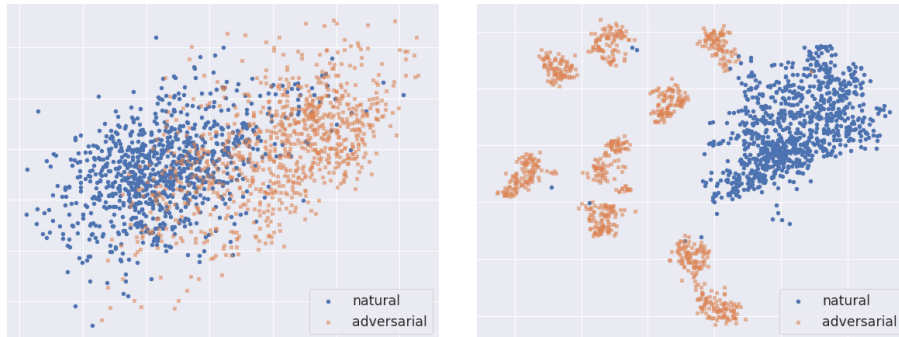
Fig. 1: Separability test for the MNIST dataset: if t-SNE projections are learned from natural samples only, the learned map cannot separate natural from adversarial samples (left). Whereas, if mixing natural and adversarial samples when learning, the resulting map is able to separate them (right).
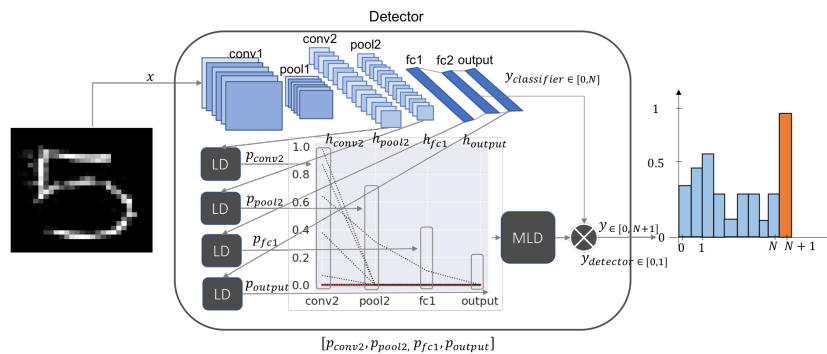

Fig. 2: Detector architecture for a convolutional NN with four monitored layers.

output of the classifier $y_{classifier} \in [0, N]$ (for N target classes) is overridden by class $N + 1$ to indicate that x is an adversarial example.

Attached to a given layer l in the network architecture, LD_l is composed by a *Mapper* and an *Estimator* for each target class. The *Mapper* of a given class, exploits the t-SNE algorithm to map samples from the high-dimensional embedding space in which $h_l(x)$ lies, to a small-dimensional one (typically two or three dimensions) in which natural and adversarial samples for such class are *likely* to be separated [1]. In this low dimensional space, a Kernel Density Estimator (KDE) is fitted on natural samples only, so to learn the distribution of natural data of a given target class. When a new data sample x' is to be classified and $h_l(x)$ is computed, the LD_l (at layer l) use each class *Mapper* to

---

[1]Experiments show that the deeper in the classifier architecture is the layer which the LD is attached to, the better t-SNE is able to discriminate between natural and adversarial samples.

project this new point in a low dimensional space and then uses class *Estimators* to score the probability of x' to be of each target class. Intuitively $p_l(x)$, the probability-score vector of LD_l for x, represents the likelihood of x being from one of the target classes at layer l.

In order to capture slight input perturbations, as in the case of adversarial examples, t-SNE has to be fed both with natural and adversarial examples (see the example in Fig.1). As such, *Mapper* training requires crafting adversarial samples à la *Adversarial Training* [9]. Adversarial Training provides evidence of increasing robustness of the classifier with respect to the attack used for crafting training adversarial samples solely, nevertheless, it is widely accepted as the most reliable countermeasure provided that the attack used for training is general enough to subsume many different attack algorithms [10]. The same assumption holds in our mapping algorithm: by crafting minimal-distance adversarial examples with a powerful attack as [11] during training, we found *Mapper* to be able to separate adversarial samples also from other existing attack algorithms as *Fast Gradient Sign Method* (FGSM) [9], *Basic Iterative Gradient Sign Method* (BIM) [12] and *Projected Gradient Descent* (PGD) [10]. A great drawback of Adversarial Training is that, since adversarial samples are used to regularize the model during training, they have to be generated at each epoch. This is very computationally demanding and needs *fast* algorithms to craft adversarial samples (e.g., FGSM), which produces sub-optimal solutions for the attacker optimization problem. Our detector, instead, requires to generate adversarial samples only *once* before starting to train, as the very same adversarial samples are used to train the LDs altogether.

The rationale for treating the output of each LD as a whole comes from observing the trends of the p scores across layers for natural and adversarial samples. In Fig. 3, it is reported a comparison between the series of p scores produced for a given natural example (on the left) and a corresponding adversarial one (on the right). The two data series present significative differences in trends: as one may expect, in case of natural data the series of p scores across layers achieve high values for the target class (i.e., above 0.5) consistently (see Fig. 3 left). Whereas, for adversarial samples, the p scores series for the target class drops to values slightly above zero, mixing with the data series of the non-target classes, resulting in misclassification of the sample.

3   Experiments

Experiments assess the detection capabilities of the proposed detector in protecting a convolutional neural network (CNN) of *LeNet* [13] type for image recognition on MNIST and CIFAR10 datasets. The accuracy of the classifier on test data drops from 0.98 for MNIST dataset and from 0.77 for CIFAR10 to approaching zero when samples are perturbed with adversarial noise. Wrapping the classifier with our detector allows maintaining high accuracy on slightly perturbed samples (i.e. on which the perturbation does not induce the label change) while detecting adversarially perturbed ones (i.e. in which the label change is
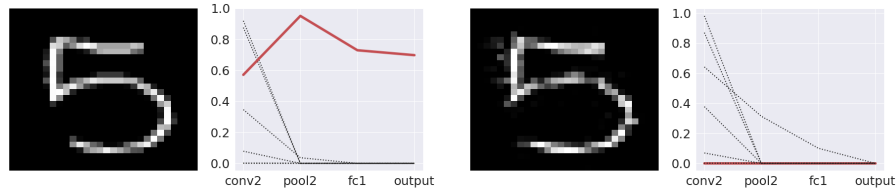
Fig. 3: Comparison of p scores data series for a natural sample from MNIST dataset (left) and a corresponding adversarial example (right). Solid line identifies the p scores data series of the original target label for the input sample, whereas dashed lines are for other classes.
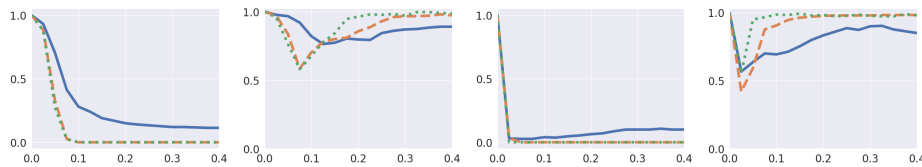

Fig. 4: Security evaluation curves for classifier and detector on MNIST and CIFAR10 datasets, respectively. We used solid lines for FGSM attack, dashed lines for BIM and dotted lines for PGD.

induced). In our experiments, we crafted the adversarial samples needed to fit the *Mappers* using Carlini-Wagner attack [11]. Protecting an N class classifier from adversarial examples is cast as a $N + 1$ classification problem, where the $N + 1$ class represents adversarial samples. We measured the increase in robustness against adversarial examples by means of *security evaluation curves* [14]. Given a classifier to be evaluated, this amounts to testing its accuracy (on y-axis in the figures) with respect to increasing adversarial perturbations (x-axis). We performed a security evaluation of the CNN classifier and of our detector against state-of-the-art adversarial attacks such as FGSM, BIM, and PGD using L_2 norm as distance metrics between samples. In Fig. 4 are reported the security curves constructed for the classifier the detector respectively for MNIST and CIFAR10 dataset. It can be noticed that, by exploiting p values trends, our detector is able to protect the CNN classifier "under attack" by maintaining the accuracy above 0.8 for MNIST and above 0.7 for CIFAR10, on average.

4   Contributions, Limitations and Future Work

We introduced a new detector for adversarial examples which exploits manifold learning techniques to identify adversarial samples from natural ones. Our findings provide experimental support to the *manifold hypothesis* which identifies adversarial examples as out-of-distribution samples in the neural representation manifolds. This renews the hope to be able to protect an existing classifier by wrapping it with a detector that identifies and reject out-of-distribution data.

In future work, we intend to evaluate our detector against adaptive attacks that also try to bypass the detection method and to compare it with other existing detection techniques.

References

[1] Battista Biggio, Igino Corona, Davide Maiorca, Blaine Nelson, Nedim Šrndić, Pavel Laskov, Giorgio Giacinto, and Fabio Roli. Evasion attacks against machine learning at test time. In *Joint European conference on machine learning and knowledge discovery in databases*, pages 387–402. Springer, 2013.

[2] 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.

[3] Nicholas Carlini and David Wagner. Adversarial examples are not easily detected: Bypassing ten detection methods. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, pages 3–14. ACM, 2017.

[4] Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. *arXiv preprint arXiv:1802.00420*, 2018.

[5] Battista Biggio and Fabio Roli. Wild patterns: Ten years after the rise of adversarial machine learning. *Pattern Recognition*, 84:317–331, 2018.

[6] Reuben Feinman, Ryan R Curtin, Saurabh Shintre, and Andrew B Gardner. Detecting adversarial samples from artifacts. *arXiv preprint arXiv:1703.00410*, 2017.

[7] Jan Hendrik Metzen, Tim Genewein, Volker Fischer, and Bastian Bischoff. On detecting adversarial perturbations. *arXiv preprint arXiv:1702.04267*, 2017.

[8] Laurens Maaten. Learning a parametric embedding by preserving local structure. In *Artificial Intelligence and Statistics*, pages 384–391, 2009.

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

[10] 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.

[11] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 39–57. IEEE, 2017.

[12] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial examples in the physical world. *arXiv preprint arXiv:1607.02533*, 2016.

[13] Yann LeCun, Léon Bottou, Yoshua Bengio, Patrick Haffner, et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[14] Battista Biggio, Giorgio Fumera, and Fabio Roli. Security evaluation of pattern classifiers under attack. *IEEE Transactions on Knowledge and Data Engineering*, 26(4):984–996, 2014.