From investigating detection of adversarial examples to achieving better adversarial robustness.

Yoann Lemesle [1] **Claire Vernade** [2]

Abstract

This report investigates the connections between the problem of detecting adversarial examples for neural networks and that of improving their robustness. We first explore state-of-the-art detection techniques and we find out that one of their key component is that they enforce regularized representations to ease outlier detection. We connect this idea with another line of work in this field which is concerned with improving adversarial robustness.

1. Introduction

Adversarial examples have been identified early as an important breach in the robustness of deep neural networks (Szegedy et al., 2013; Goodfellow et al., 2014). These inputs can be crafted by more or less knowledgeable adversaries to fool the classifier while being visually (nearly) identical to images in the training set. There has been huge efforts spent in attempting to improve the robustness of the existing architectures, especially via new training methods called *adversarial training* (Madry et al., 2019) that aim at regularizing the classification borders to prevent networks from the existence of artifacts. Indeed, most of these methods come at either higher computational cost, or a loss in accuracy, or both.

Comparatively, the problem of detecting adversarial inputs is less well understood. Nonetheless, it arises from a simple intuition. The way adversarial examples are crafted is via gradient ascent whereby a natural training image is slowly transformed into a wrongly labeled sample which lies on the other side of, but close to, the classification border. Thus, one can conjecture that the representation of these samples are outliers in the manifold of their predicted class. Most previous methods rely on estimating the uncertainty of the network at that point, either via a Kernel Density Estimate (KDE) or other uncertainty estimates (Feinman et al., 2017b). Nevertheless, Carlini & Wagner (2017) show that for multiple reasons, and using a variety of techniques, these detectors can be byassed.

In this work we investigate the connections between state-of-the-art detection techniques (Pang et al., 2017) and adversarial robustness. We show that regularized representations are at the heart of both problems, but both objectives remain incompatible. In particular, we find out that more adversarially robust networks tend to have worse detection scores.

This report presents the background, the methods and the results of our investigations. We begin by providing the necessary background on adversarial robustness and detection. Our methodology is presented in Section 3 and our experimental results are described in details in Section 4.

2. Background

2.1. Neural Networks

A neural network, or deep learning model, is a mapping function $F(x, \theta) : \mathbb{R}^d \to \mathbb{R}^N$ where $x \in \mathbb{R}^d$ is the input and θ is the model's parameters. Such a function is differentiable with respect to θ, meaning that its parameters can be updated through gradient descent in order to learn a specific mapping. These models have been shown to have great empirical performance in tasks like image classification or natural language processing (Goodfellow et al., 2016).

In the case of supervised learning, neural networks can learn a specific mapping by minimizing a loss function $J(F(X, \theta), Y)$ that outputs how close the model's predictions $F(X, \theta)$ are to the expected outputs Y on an (X, Y) training dataset containing pairs of input-label examples. Learning then corresponds to finding the parameters θ^* that minimize such a loss function. We will now refer to $F(x, \theta)$ as $F(x)$ for simplicity. Whenever we talk about optimizing a loss function, optimizing J with respect to θ is implied.

For classification tasks where inputs must be discriminated between N classes, a model outputs a probability vector $F(x) \in \mathbb{R}^N$ giving the estimated confidence for each class. These probabilities $F(x) = S(Z(x))$ are obtained by passing the outputs $Z(x) \in \mathbb{R}^N$ of the model, denoted as the

[1]École Normale Supérieure de Rennes [2]Deepmind, London, UK. Correspondence to: Yoann Lemesle <yoann.lemesle@ens-rennes.fr>.

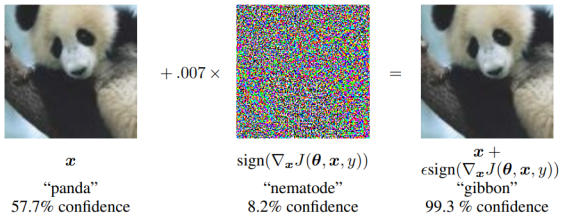

Figure 1: Illustration of the FGSM attack (Goodfellow et al., 2014)

logits, in the softmax function :

$$S(\mathbf{x})_i = \frac{\exp(\mathbf{x}_i)}{\sum_{j=1}^{N} \exp(\mathbf{x}_j)}$$
$$\text{where } \mathbf{x} \in \mathbb{R}^N$$

An input x is then classified as the label that has the largest probability, such a label is : $\hat{y} = \arg\max_i F(x)_i$.

The loss function that is the most commonly used for classification tasks is the Cross-Entropy loss function :

$$CE(x, y) = -\log F(x)_y \tag{1}$$

2.2. Adversarial Examples

It has been shown that neural networks are susceptible to adversarial perturbations, small modifications to an input that are able to fool a model and completely modify its outputs (Szegedy et al., 2013). Namely, for some small $\epsilon > 0$ and some input x, there exists x_{adv} such that $\|x - x_{\text{adv}}\| < \epsilon$[1] but $\hat{y}(x) \neq \hat{y}(x_{\text{adv}})$. Such modified inputs are called adversarial examples and we call adversarial robustness the stability of the outputs of a neural network with respect to these adversarial perturbations. We call adversarial attacks the process of crafting adversarial examples.

Most adversarial attacks rely on the gradients Δ_x of the loss function with respect to the input's values, as illustrated by the Fast Gradient Sign Method (FGSM) (Goodfellow et al., 2014), a simple single-step method where :

$$x_{adv} = x + \epsilon * sign(\nabla_x J(F(x, \theta), y))$$

Reviewing the entire literature on that problem is beyond the scope of this report but we focus on providing the key references that our work builds upon.

Many works have attempted to improve the adversarial robustness of neural networks by regularizing the loss function during training. Adversarial training (Goodfellow et al., 2014; Madry et al., 2019) is a method that involves the use of adversarial attacks on inputs during training. By learning on adversarial examples instead of normal inputs, the model can be made more robust to the attacks that are used during training. However, using adversarial attacks during training comes with a high computational cost. Indeed, performing an adversarial attack at every training iteration implies to differentiate the entire model with respect to the input's values. Single-iteration methods like FGSM are not especially costly, however one needs to use more powerful multi-iterations methods in order to achieve robustness for a bigger range of attacks, thus dramatically increasing the computational cost of training.

In order to mimic the effects of adversarial training with a computationally efficient method, (Shafahi et al., 2019) make use of two previously known method : label smoothing and logit squeezing. Label smoothing (Szegedy et al., 2015) minimizes the variance in the model's probability outputs while logit squeezing (Kannan et al., 2018) minimizes the norm of the logits. Combined with the addition of Gaussian Noise on the training inputs, these regularization methods allows for models that are even more robust than with adversarial training, without its computational cost. We describe these methods in greater details in the next sections.

However, the issue of adverarial robustness in deep learning models is still an open problem. Indeed, as the exact origin of adversarial examples is still not well understood (Serban et al., 2019), researching methods of defenses has been proven to be especially challenging. As of today, full robustness to adversarial attacks has yet to be achieved.

3. Methodology

3.1. Reverse Cross Entropy

Because of how difficult it is to improve the adversarial robustness of neural networks, some have instead investigated adversarial detection as an alternative. Instead of making a model resistant to adversarial examples, the idea behind these works is to detect when inputs have been modified by an adversary. Several methods have been proposed like training networks for detection (Metzen et al., 2017), reducing the dimensionality of the inputs (Bhagoji et al., 2017) or detecting representations that lie far from the natural activations manifold (Feinman et al., 2017a).

[1]The choice of norm differ across papers, most common choices are l_2, l_∞ or l_1 norm.

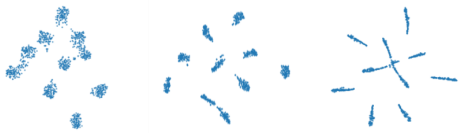

Figure 2: t-SNE representation of the activations in the last hidden layer of models trained using label smoothing regularization with $\lambda = 0$ (left), $\lambda = 1$ (center) and $\lambda = \infty$ (right).

However, all these methods can easily be bypassed with appropriate defenses (Carlini & Wagner, 2017). Indeed, these detection methods $D(x)$ are differentiable with respect to the input's values. It is therefore possible to compute $\Delta_x D(x)$ which means that the value of $D(x)$, like $J(x, y)$, can easily be manipulated with an adversarial attack that is adapted to the specific detection term. This highlights the problem of the *robustness of detection itself*.

In order to achieve more robust detection of adversarial examples, (Pang et al., 2017) propose an alternative to the Cross-Entropy loss function : the Reverse Cross-Entropy (RCE) loss. This method builds on the label smoothing (Szegedy et al., 2015) regularization method :

$$J^{\lambda}_{CE}(x, y) = CE(x, y) - \lambda \cdot R_y^{\top} \log F(x)$$
$$\text{where } R_y^y = 0 \text{ and } R_y^{i \neq y} = \frac{1}{N-1}$$

where CE is the cross-entropy defined in Eq 1 and F is the network function.

Optimizing the regularization term maximizes the probability of the wrong classes while still requiring the confidence of the true class to be maximal, thus reducing the variance in the model's output values. On the other hand, training a model with Reverse Cross-Entropy only maximizes the probability of the wrong classes, thus being equivalent to training with J^{∞}_{CE} :

$$RCE(x, y) = -R_y^{\top} \log F(x)$$

Training with such a loss function yields a model with reverse classification : the true class has the minimal probability while the other classes are equiprobable.

By using t-SNE visualization, a probabilistic method of visualizing high-dimensional data (van der Maaten & Hinton, 2008), on models trained with label smoothing (Figure 2), we can see how an increase in λ is correlated with a constraint on the distribution of the models activations. The authors explain how this constraint, maximized with RCE, increases the robustness of KDE-based adversarial detection on these activations.

Kernel Density Estimation (KDE) (Rosenblatt, 1956; Parzen, 1962) is a method to estimate the probability of a point x to belong in a distribution given a set X of samples from that distribution. Using a parameter σ known as the bandwidth, such a probability is defined as follows :

$$\hat{f}_{\sigma}(x) = \frac{1}{|X|} \sum_{x_i \in X_t} \exp(-||x_i - x||^2 / \sigma^2) \qquad (2)$$

As the distributions of the models activations are less spread out in the representation space, bypassing the detector involves to carefully craft an adversarial perturbation such that the input's representation in the last hidden layer is close enough to the dense distributions. The increase in the required precision of the attacks makes bypassing the detection harder, requiring the use of perturbations with higher norms.

3.2. Output Variance Minimization (OVM)

Building on the correlation between constrained representations and the robustness of detection, our initial motivation was to explicitly constraint the activations of a model in order to achieve robust detection.

To do so, we propose to minimize the variance in the model's outputs for each class of inputs by adding a regularization term to the Cross-Entropy loss function. For an (X,Y) batch of input-label pairs of data with N possible labels, we use the following loss function :

$$OVM(X, Y) = CE(F(X), Y)$$
$$+ \lambda \cdot \frac{1}{|X|} \sum_{i=0}^{N-1} \sum_{x \in X_i} ||\log F(x) - \overline{\log F(X_i)}||_2$$
$$\text{where } X_i = \{x \mid (x, i) \in X\}$$

However, an easy way to minimize the regularization term :

$$\sum_{x \in X_i} ||\log F(x) - \overline{\log F(X_i)}||_2$$

would be to mostly minimize :

$$\sum_{x \in X_i} ||\overline{\log F(X_i)}||_2$$

by preventing values in $F(x)$ to be too close to zero, a value that causes the l_2 norm of a $\log F(x)_i$ output to be infinite. Such a constraint is easier to optimize with respect to minimizing the Cross-Entropy loss function than minimizing the variance in the model's outputs.

Figure 3: t-SNE representation of the activations in the last hidden layer for the CE (left), RCE (center) and OVM (right) models.

Constraining the values of $F(x)$ to be far away from zero encourages a smaller difference between the maximal and non-maximal values of the logits. This has the effect of minimizing the relative differences between the class probabilities, which is related with label smoothing, while also minimizing the l_2 norm of the logits. The latter effect is known as logit squeezing (Kannan et al., 2018), a regularization method that constraints a model to output logits with low l_2 norms.

Because of the known relationship between label smoothing, logit squeezing and adversarial robustness, our detection-oriented method is actually closely related with adversarial robustness. This also sparks the question of the relationship between adversarial detection and robustness. All these questions are further explored in the next section.

4. Experiments

Optimizer	SGD
Momentum	0.9
Weight Decay	5e-4
Initial LR	0.1
LR Scheduler	Cosine Annealing
Epochs	200
Batch Size	256
Data Augmentation	Random Horizontal Flips Random Cropping
Pixel Values Range	[-0.5,0.5]
Relu Leakiness	0.1

Table 1: Training Hyperparameters

Method	Accuracy
CE	93.13
RCE	92.94
OVM	91.5

Table 2: Classification accuracy (%) on the CIFAR-10 test set

4.1. Setup

In order to evaluate the properties of a model trained with our method, we trained three Resnet-32 models on the CIFAR-10 dataset (Krizhevsky, 2009) using the Cross-Entropy (CE), Reverse-Cross-Entropy (RCE) and Output Variance Minimization (OVM) training procedures. We used a balancing parameter $\lambda = 1$ for OVM. The full training setup is described in Table 1 while the resulting models classification accuracies are shown in Table 2. While being lower than the others, the classification accuracy does not seem to have been significantly affected by our method.

4.2. Adversarial Detection

	FGSM	BIM	PGD
CE	88.4	99.9	99.9
RCE	85.7	99.6	98.4
OVM	73.1	83.9	77.2

Table 3: AUC-scores of detection for the different models using different adversarial attacks.

In (Pang et al., 2017) the authors show the t-SNE representations of the activations of the last hidden layer for the CE and RCE models in order to illustrate how the distributions of these outputs differ between the two models. In the case of RCE, more compact distributions were correlated with better and more robust detection of adversarial examples using KDE on the last hidden layer. Because of this correlation, we started by measuring these t-SNE representations for the three models in order to see how our method affects the activations distributions. Results shown in Figure 3 show how OVM seems to effectively constrains the representations in the last hidden layer to be more compact in their distributions, hinting a better robustness in adversarial detection.

To measure the adversarial detection capacities of the models, we measured the AUC-scores of detection using KDE (see Eq 2) on the last hidden layer, with a bandwidth $\sigma = 1/0.26$ for CE and $\sigma = 0.1/0.26$ for RCE/OVM. Adversarial examples are generated by using three different adversarial attacks, detailed in Section 4.3. While the com-

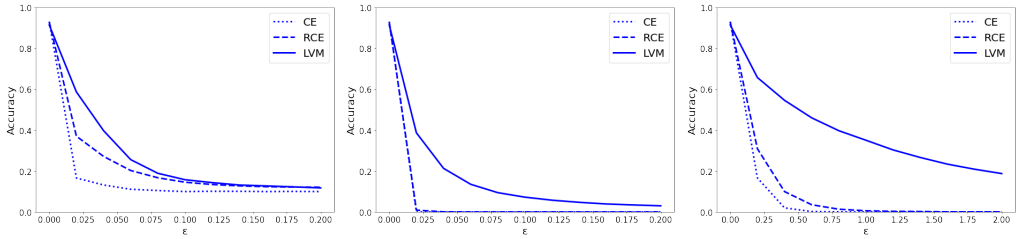

Figure 4: Adversarial robustness of the CE, RCE and OVM models with respect to ϵ using FGSM (left), BIM (center) and PGD (right).

pact distributions of the OVM model suggest that detection could be better than with CE, the results of Table 3 actually show the opposite : the model trained with our regularization method has the worst detection performance of all three models.

This shows how the correlation between the distributions of the activations in the last hidden layer and the detection performance by using KDE on these outputs may not be as valid as suggested in (Pang et al., 2017).

4.3. Adversarial Robustness

We then evaluated the adversarial robustness of the models using the following adversarial attacks with different values of ϵ :

Fast Gradient Sign Method (FGSM) (Goodfellow et al., 2014). $x' = x + \epsilon \cdot sign(\nabla_x J(x, y))$ where x' is the resulting adversarial example, x is the original input, J is the loss function and y is the correct label.

Basic Iterative Method (BIM) (Kurakin et al., 2017). $x'_i = x'_{i-1} - + \frac{\epsilon}{n} \cdot sign(\nabla_x J(x, y))$ where x'_0 is the original input and n is the number of iterations. We used this attack with $n = 10$ iterations.

Projected Gradient Descent (PGD) (Madry et al., 2019). Similar to BIM, except that the adversarial perturbation $x' - x$ is constrained to be under a certain l_p-norm[2] D_{total}, while each gradient step is constrained to have a fixed l_p-norm, D_{step}. We used this attack with n = 10 iterations, $D_{total} = \epsilon$ and $D_{step} = \frac{\epsilon}{10}$.

The adversarial robustness of the three models for each attack are shown in Figure 4. While the OVM model does not seem to be better in terms of detection, it is significantly more robust than the other methods for all three attacks, especially for the most powerful PGD attack.

These results support the idea, described in Section 2, that OVM is actually closely related with the label smoothing and logit squeezing regularization methods.

[2]Again, the choice of norm differs across papers and is often $p = 1, 2, \infty$.

4.4. Additional Experiments

As we've seen, the encouraging results of our methods may only be due to its close relation with logit squeezing and label smoothing, two regularization methods already known to have positive effects on adversarial robustness.

However, our methods seems to yield bad results in term of detection, which sparks the question of the relationship between adversarial robustness, adversarial detection, and regularized models.

In order to explore possible correlations between these different properties, we trained several Resnet-32 models on the MNIST dataset (Deng, 2012) using both logit squeezing and label smoothing regularizations. Each regularization term is multiplied by its respective balancing parameter (see Eq 3). The values used for both parameters are [0,0.1,0.2,0.4,0.8,1].

$$J^{\lambda_1,\lambda_2}(x, y) = \\ CE(x, y) - \lambda_1 \cdot R_y^\top \log F(x) + \lambda_2 \cdot ||Z(x)||_2 \tag{3}$$

We then measured the adversarial robustness and detection results of the 36 models under the PGD attack, using the same parameters as in section 4.3. The results shown in Figure 5 seem to show an inverse correlation between adversarial robustness and adversarial detection.

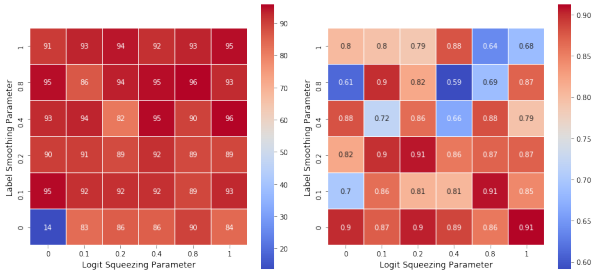

Figure 5: Adversarial robustness (left) and AUC-scores of detection (right) of the regularized MNIST models with respect to $\lambda 1$ (label smoothing) and $\lambda 2$ (logit squeezing).

5. Conclusion

In this report, we investigated the connections between the detection of adversarial examples and adversarial robustness. Inspired by the Reverse Cross-Entropy training procedure, we proposed a regularization method aimed at constraining the activations distributions of a model in the hope of improving adversarial detection. In doing so, we actually achieved a regularization method that is closely related with label smoothing and logits squeezing, two regularization methods that improve adversarial robustness. These results motivated an investigation of the relationship between adversarial robustness and adversarial detection, whose results hint that there may exist an inverse correlation between the two. However, these results are still too preliminary and more research needs to be done on this topic.

References

A. Bhagoji, Daniel Cullina, and Prateek Mittal. Dimensionality reduction as a defense against evasion attacks on machine learning classifiers. *ArXiv*, abs/1704.02654, 2017.

Nicholas Carlini and David Wagner. Adversarial examples are not easily detected: Bypassing ten detection methods, 2017.

L. Deng. The mnist database of handwritten digit images for machine learning research [best of the web]. *IEEE Signal Processing Magazine*, 29:141–142, 2012.

Reuben Feinman, Ryan R. Curtin, Saurabh Shintre, and Andrew B. Gardner. Detecting adversarial samples from artifacts, 2017a.

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

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples, 2014.

Harini Kannan, Alexey Kurakin, and Ian Goodfellow. Adversarial logit pairing, 2018.

A. Krizhevsky. Learning multiple layers of features from tiny images. 2009.

Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial examples in the physical world, 2017.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks, 2019.

Jan Hendrik Metzen, Tim Genewein, Volker Fischer, and Bastian Bischoff. On detecting adversarial perturbations, 2017.

Tianyu Pang, Chao Du, Yinpeng Dong, and Jun Zhu. Towards robust detection of adversarial examples, 2017.

Emanuel Parzen. On Estimation of a Probability Density Function and Mode. *The Annals of Mathematical Statistics*, 33(3):1065 – 1076, 1962. doi: 10.1214/aoms/1177704472. URL https://doi.org/10.1214/aoms/1177704472.

Murray Rosenblatt. Remarks on Some Nonparametric Estimates of a Density Function. *The Annals of Mathematical Statistics*, 27(3):832 – 837, 1956. doi: 10.1214/aoms/1177728190. URL https://doi.org/10.1214/aoms/1177728190.

Alexandru Constantin Serban, Erik Poll, and Joost Visser. Adversarial examples - a complete characterisation of the phenomenon, 2019.

A. Shafahi, Amin Ghiasi, F. Huang, and T. Goldstein. Label smoothing and logit squeezing: A replacement for adversarial training? *ArXiv*, abs/1910.11585, 2019.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks, 2013.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision, 2015.

Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(86):2579–2605, 2008. URL http://jmlr.org/papers/v9/vandermaaten08a.html.