ESANN 2020 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 2-4 October 2020, i6doc.com publ., ISBN 978-2-87587-074-2.
Available from http://www.i6doc.com/en/.

Towards Adversarial Attack Resistant Deep Neural Networks

Tiago A.O. Alves[1] and Sandip Kundu[2]

1- State Unversity of Rio de Janeiro - UERJ
Rio de Janeiro, Brazil

2- University of Massachusetts - UMass
Amherst, MA, United States

Abstract. Recent publications have shown that neural network based classifiers are vulnerable to adversarial inputs that are virtually indistinguishable from normal data, constructed explicitly for the purpose of forcing misclassification. In this paper, we present several defenses to counter these threats. First, we observe that most adversarial attacks succeed by mounting gradient ascent on the confidence returned by the model, which allows adversary to gain understanding of the classification boundary. Our defenses are based on denying access to the precise classification boundary. Our first defense adds a controlled random noise to the output confidence levels, which prevents an adversary from converging in their numerical approximation attack. Our next defense is based on the observation that by varying the order of the training, often we arrive at models which offer the same classification accuracy, yet they are different numerically. An ensemble of such models allows us to randomly switch between these equivalent models during query which further blurs the classification boundary. We demonstrate our defense via an adversarial input generator which defeats previously published defenses but cannot breach the proposed defenses do to their *non-static* nature.

1 Introduction

In adversarial attacks against black-box ML systems, an attacker does not have any information about the model or the training dataset. The adversary has to explore different input feature vectors to maximize some function of the output of the ML model to which they have access. Typically, the goal of the adversary is to either cause a misclassification, by adding minimal disturbance to the input, or to obtain sensitive information, by maximizing the confidence levels of the output of the model. Fredrikson *et al.* [1] first introduced this method known as *Model Inversion Attack* (MIA).

In this work we present two defenses against Model Inversion Attack on black-box model by adding entropy to the confidence levels returned by the Machine Learning System, to prevent convergence of gradient-ascent algorithms which constitute the core of adversarial input generation. The first defense simply injects random noise with long-tailed distribution to the confidence information returned by the model. The reasoning behind the choice of this kind of distribution is to statistically bound the noise level to prevent/reduce misclassification on normal data. The second defense adopts replicas of the same ML model trained

ESANN 2020 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 2-4 October 2020, i6doc.com publ., ISBN 978-2-87587-074-2. Available from http://www.i6doc.com/en/.

with different datasets or with the same dataset with shuffled order and randomized transforms. Then, each call to the model picks one of the instances of the model to conduct the query and the result will not be deterministic because of the small variation between the weights and biases of each trained model. This approach also, as we will show, has no effect on the misclassification rate, since all models used are expected to have similar levels of accuracy. Neither of these defenses are without their caveats which are addressed in this paper.

To demonstrate the effectiveness of the solution, first we build an adversarial input generator as described in Algorithm 1. A key difference between this generator and previous ones is that we do not place artificial bounds on the amount of noise an adversarial input is allowed to have. Our examples show that using this approach against unprotected or statically protected classifiers still results in adversarial images that to the human eye look very similar to the original ones. This relaxation on the requirements of the attack makes it highly intrusive and, thus, more suited to prove the effectiveness of our defenses.

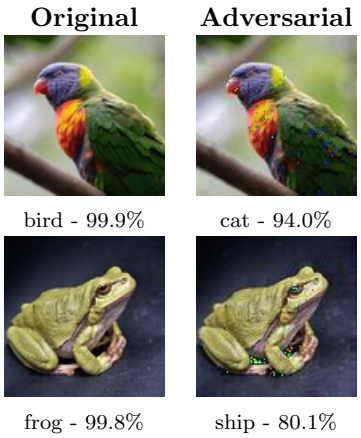

Fig. 1: Adversarial examples. Original images (left) and the **targeted** adversarial examples (right) obtained using our numerical implementation of gradient descent based attack. Below each image is the classification and confidence returned by the ResNet CIFAR-10 Image Classifier.

We describe the proposed attack in Algorithm 1, where $f_k(\mathbf{x})$ represents the confidence level of the k-th output of the model. In the case of a classifier, this algorithm returns an input vector that gets classified as the class k, since its output is the one being maximized.

Algorithm 1 General Adversarial Attack against Image Classifiers. Where a threshold for the minimum confidence level $f_k(\mathbf{x})$ is arbitrarily defined and α is some empirically chosen scalar that represents the *step size* of the algorithm.

Require: Initial vector x representing the pixels of the input image and the target class k

Ensure: Output vector x with the adversarial image

while $f_k(\mathbf{x}) <$ THRESHOLD **do**

$\mathbf{x} \leftarrow \mathbf{x} + \alpha \cdot \nabla f_k(\mathbf{x})$

For the creation of RGB image adversarial examples such as the ones in Figure 1, the attack performs better if the gradient is replaced by the *sign gradient*, $sign(\nabla f_k(\mathbf{x}))$, in Algorithm 1.

ESANN 2020 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 2-4 October 2020, i6doc.com publ., ISBN 978-2-87587-074-2.
Available from http://www.i6doc.com/en/.

1.1 Defenses

Existing defenses such as *robust neural networks* [2] or applying transforms, such as *crop* or *jpeg* compression, to the input image [3], can be undermined by targeted adversarial attacks, as evidenced by our experiments. In all of the attacks performed in this work, jpeg compression as well as resize and crop were used as transforms on the input and still the proposed attacks prevailed, unless our *defenses* are enabled. As our goal in this work is to present defenses that can endure virtually all black-box confidence based attacks, we will not present further discussion or results on which variation of these attacks yields the smallest perturbation that causes targeted or non-targeted misclassification. Instead we will just focus on presenting countermeasures to defeat such attacks, since in this black-box scenario they all have to be based on numerical approximations of $\nabla f_k(\mathbf{x})$. So, the basis for our attacks remains as the approximation of each partial derivative of $f_k(\mathbf{x})$: $\frac{\partial f_k}{\partial x_i} \approx \frac{f_k(x_0,x_1,..,x_i+h,..,x_n)-f_k(\mathbf{x})}{h}$

2 Proposed Defenses

2.1 Noise-Injection to Confidence Levels

Our first countermeasure to prevent adversarial attacks does not require retraining or any alterations to the ML system's inner workings. This defense relies on the injection of long-tailed distributed errors to the output $f(\mathbf{x})$ of the model, so that a model inversion attack can not converge, whilst maintaining the functionality of the ML system reliable for legitimate users.

The motivation for this defense is that when entropy is added to the confidence information returned by ML model, attacks based on gradient ascent/ descent are unable to converge as easily in *static* black-box systems. Since a black-box attack has to rely on numerical approximations of the gradient $\nabla f(\mathbf{x})$, which are obtained by numerical differentiation of f on each of the features, the attack is unable to produce the same results if the value of f is nondeterministic. Meanwhile it is important to preserve correct classifications for legitimate users. As we will show in Section 3, this is ensured by the chosen distributions for the random noise. Since the injected noise has probability density functions (pdf) which are long-tailed, noises big enough to render missclassification are rare.

Furthermore, 50% of the times the error is multiplied by -1, which yields a distribution of errors with a mean equal to 0. This type of pdf, potentially yields a wavelike behaviour when adversarial attacks are attempted against ML System where this defense was implemented. This behaviour, shown in Section 3, hinders the ability of adversarial attacks based on gradient descent to perform their approximations.

The way it was presented so far, this defense suffers from a flaw which is also remedied in this paper. It would be possible for the attacker to just repeat the same query multiple times and average out the injected noise injected to zero. To counter this possibility, here we propose to use the input vector $\mathbf{x}$ as a seed of the pseudo-random number generator, so that the attacker will get always the same *skewed* output (same noise) for a given input, regardless of how many times the

ESANN 2020 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 2-4 October 2020, i6doc.com publ., ISBN 978-2-87587-074-2.
Available from http://www.i6doc.com/en/.

model is queried with the same input. This could simply be done by hashing the input vector, but since in the case of numerical approximations of the gradient, the input vectors used by the attacker to perform the approximations have very close values, in our observations, the hashes collide frequently, rendering our first attempt to add entropy ineffective.

To address hash collision issue, we first encrypt the input vector (using AES-128 encryption in our experiments with a fixed key) and then hash the encrypted information to fit the data size of the seed for the pseudo-number generator. The reason for this is that even a small difference between two input vectors generates a large difference in the output of the encrypted information, so that the hash values we end up using for consecutive seeds in the pseudo-random number generation are far from each other, effectively eliminating the hash collisions. This modification to the defense, to the best of our knowledge, was the key factor in preventing all current possible forms of evasion against it.

Another positive aspect this defense mechanism is that there is no need for re-training of any kind of modification in the ML model itself. Since we just inject noise in the confidence information returned by the model, this approach can potentially work on any ML model without any modifications to the model itself. As we will show in Section 3, the provider of the model only has to chose the distribution and its parameters parameters. This can be done empirically via experimentation for each model and dataset, as demonstrated in this work[1].

2.2 Pooling Multiple Models

This defense relies on the nondeterminism present in the training process of the ML models to add entropy to the outputs. The entropy here comes from the fact that, during the training stage, the dataset presented to the model is shuffled and each instance of the training set goes through transforms that are also random (such as random crop and rotation). Therefore, if you train the same DNN multiple times, the final trained models will have some entropy between that, i.e., the outputs for the same input will not be exactly the same.

We take this fact and train the same model multiple times (five models for the purpose of results shown in this paper), ending up with different weights and biases for the model after each training. Then, we store these instances and randomly (with uniform distribution) select which trained model to use to classify the input.

The downside of this approach, when compared to Noise-Injection, is the extra memory required to store the extra instances of weights and biases of the DNN.

[1]Due to space constraints we only present our experiments with the ResNet-18 architecture and CIFAR-10 Benchmarks. But our defenses successfully prevent attacks against all state of the art image classifier architectures.

ESANN 2020 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 2-4 October 2020, i6doc.com publ., ISBN 978-2-87587-074-2.
Available from http://www.i6doc.com/en/.

3 Experiments

3.1 Adversarial Attack Used for Testing Defenses

It is important to notice that all adversarial attacks conducted in this paper are based on black-box scenario, where the adversary only has access to input and output of the model, the latter carrying confidence information about the classification. This way, the attacker is oblivious to what kind of ML model is applied and can not possibly use regular analytical mathematics to obtain gradients. In such a scenario, the adversary is forced to rely on numerical methods which, as we show in our experiments, fail to converge once the proposed defenses are applied.

This approach sets our work apart from most of previous work. Since the attacks rely on approximations, it is not possible to obtain precise minimal disturbances to target a class. In addition to that, it is necessary to fine tune the attacks in order to obtain adversarial examples that still can clearly be classified by a human observer. We observed in our experiments that the higher the resolution of the original image, the less perceptual impact the adversarial disturbances have on the product of the attack. However, due to computing-time and resources constraints we limit most of our experiments to 100x100 pixel images.

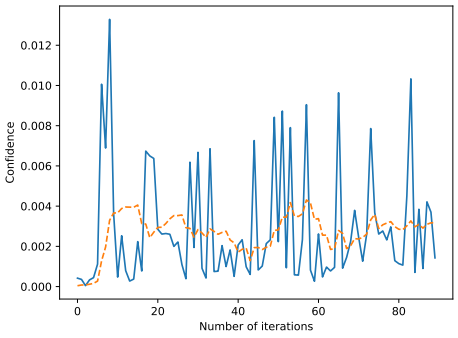

Fig. 2: Adversarial Attack on CIFAR-10 protected by Multi-Models defense. The dashed line represents the moving average of the confidence along the iterations, its trend, close to a constant line, suggests the attack is unable to converge. For this same example, without any defense, the attack does not converge and the image degrades beyond perceptual recognition.

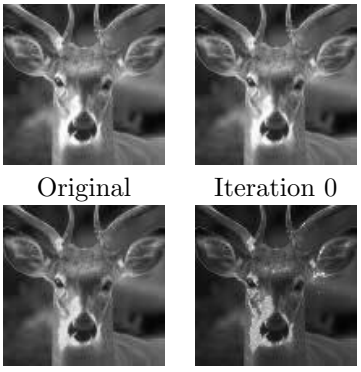

Fig. 3: Adversarial attack performed on an image originally classified as *deer*, i.e. $k_0 = deer$, with confidence $f_{k_0}(\mathbf{x}) = 0.99$. The target class $k_1 = truck$. After 3 iterations, the model classifies the image as a truck with $f_{k_1}(\mathbf{x}) = 0.99$.

3.2 Pooling Multiple Models Defense

Figure 2 shows the confidence levels obtained by the adversarial attack along its iterations, when the Multi-Models defense is applied. The overall result is that the numerical method does not converge and the resulting image ends up completely unrecognizable to a human observer. In the graph of Figure 2 we

ESANN 2020 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 2-4 October 2020, i6doc.com publ., ISBN 978-2-87587-074-2. Available from http://www.i6doc.com/en/.

also show the moving average of the confidence level, which has slope close to zero (notice the scale in the y-axis), which corroborates our assumptions.

3.3 Noise-Injection Defense

Figures 3 and 4 show how the attack fails to perform when the Noise-Injection defense is applied to the ML system. In Figure 3, the original image, which gets classified as *deer* with 99% confidence is used as input for the attack and, after three iterations, it gets classified as the target class, *truck*, with 99% confidence as well. On the other hand, in the example of Figure 4 where Noise-Injection is adopted, the attack does not converge and through the iterations, the generated image becomes perceptually recognizable.

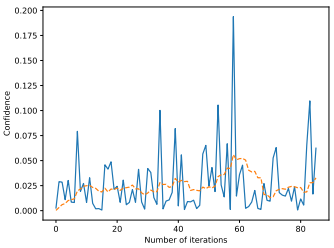

Fig. 4: Adversarial Attack on CIFAR-10 ResNet classifier with Noise-Injection Defense enabled.

4 Conclusions and Future Work

In this paper, we presented two defenses against Model-Inversion attacks on DNN. In contrast to prior literature, where defenses are static, our defenses are dynamic. Static defenses [4], require the models to be retrained and do not provide sufficient defenses when adversarial perturbation is not bounded.

Our proposed defenses overcome these limitations without compromising model accuracy. Since the noise injection of the first defense follows a long-tailed distribution, as shown experimentally, the accuracy remains unaffected. In the case of model-pooling, the accuracy remains unaffected by model construction methods. As future work, we intend to prove that these defenses also work to prevent Model Stealing Attack [5], where the goal is to introduce sufficient nondeterminism to hinder the adversary's ability to approximate a $\tilde{f} \approx f$.

References

[1] Matthew Fredrikson, Eric Lantz, Somesh Jha, Simon Lin, David Page, and Thomas Ristenpart. Privacy in pharmacogenetics: An end-to-end case study of personalized warfarin dosing. In *Proceedings of the 23rd USENIX Conference on Security Symposium*, SEC'14, pages 17–32, Berkeley, CA, USA, 2014. USENIX Association.

[2] Eric Wong and J. Zico Kolter. Provable defenses against adversarial examples via the convex outer adversarial polytope. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, pages 5283–5292, 2018.

[3] Chuan Guo, Mayank Rana, Moustapha Cisse, and Laurens van der Maaten. Countering adversarial images using input transformations. In *International Conference on Learning Representations*, 2018.

[4] Eric Wong and J. Zico Kolter. Provable defenses against adversarial examples via the convex outer adversarial polytope. 2017.

[5] Florian Tramèr, Fan Zhang, Ari Juels, Michael K. Reiter, and Thomas Ristenpart. Stealing machine learning models via prediction apis. In *Proceedings of the 25th USENIX Conference on Security Symposium*, SEC'16, pages 601–618, Berkeley, CA, USA, 2016. USENIX Association.