A full version of this paper is available at https://papernot.fr/files/extending-distillation.pdf

Poster: Extending Defensive Distillation

Nicolas Papernot and Patrick McDaniel
Pennsylvania State University
{*ngp5056,mcdaniel*}*@cse.psu.edu*

***Abstract*—Machine learning is vulnerable to adversarial examples: inputs carefully modified to force misclassification. Defending against such inputs remains an open problem. In this work, we revisit defensive distillation to address its limitations. We view our results not only as an effective way of addressing some of the recently discovered attacks but also as reinforcing the importance of improved training techniques.**

1. Introduction

Deployed machine learning (ML) models are vulnerable to inputs maliciously perturbed to force them to mispredict. Such inputs, i.e., *adversarial examples*, are systematically constructed through slight perturbations of otherwise correctly classified inputs [1], [2]. These perturbations are chosen to maximize the model's prediction error while leaving the semantics of the input unchanged. They are typically found by evaluating gradients of the model's output with respect to its inputs [1], [3].

To defend against adversarial examples, two classes of approaches exist. The first algorithmically improves upon the learning to make the model inherently more robust: e.g., adversarial training [1], [3] or defensive distillation [4]. The second is a set of detection mechanisms used to reject inputs suspected to be malicious: e.g., with an outlier class [5], [6].

However, most—if not all—of these defenses fail to adapt to novel attack strategies. Distillation is no exception. It is successful against attacks known at the time of writing, such as the Fast Gradient Sign Method (FGSM) [3] and the Jacobian-based Saliency Map Approach (JSMA) [7].

As advancements found new ways to mount attacks against ML [8], [9], defensive distillation can now be evaded. Black-box attacks from [8] enable adversaries capable of training a surrogate model to generate adversarial examples that transfer back to the distilled model (i.e., they are misclassified). In addition, distillation can be evaded with optimization attacks à la Szegedy et al. [1], which have been revisited in [9]. This is because of a phenomenon called *gradient masking*: the defense mechanism destroys gradients used by attacks, including the FGSM and JSMA, instead of reducing the model's error.

Given the two failure modes identified above, we propose in this work a variant of defensive distillation to address them. We demonstrate that our approach is much less susceptible to gradient masking by attacking it using a surrogate model. Like the original defensive distillation, the technique does *not* require that the defender generate adversarial examples. The applicability of our approach is thus less likely to be limited to specific adversarial example strategies. Unfortunately, it is currently impossible to formally prove robustness guarantees for models like deep neural networks, so we resort to experimental validation of our approach and leave a formal analysis to future work.

2. Defensive distillation

Defensive distillation successively trains two instances of a ML model. The first is trained with the original dataset $\{(x, y)\}$ where y indicates the correct class for x. Learning is performed conventionally to the exception of the softmax temperature, which is usually set to 1 and is here increased. In the interest of space, we refer readers to the detailed presentation in [4]. Briefly, the model outputs probabilities that are closer to a uniform distribution at high temperatures. This first model f is then used to label the data with its probabilities, thus defining a new training set $\{(x, f(x))\}$. The second model f^d is trained on this newly labeled data. When deployed with a temperature of 1, f^d is found to be robust to the FGSM and JSMA attacks.

3. Extending defensive distillation

We revisit the defensive distillation approach by modifying the labeling information used to train the distilled model f^d. Our changes are motivated by previous work introducing an outlier class to mitigate adversarial examples [5], [6] and providing uncertainty estimates in neural nets [10].

After training the first model at temperature $T = 1$, we take N passes with dropout to compute the uncertainty of its logits: $\sigma(x) = \frac{1}{N} \sum_{m \in 0..N-1} \left(\sum_{j \in 0..n-1} \left(z_j^m(x) - \bar{z}_j \right)^2 \right)$. Using this vector, we construct a labeling vector $k(x)$ whose components $j \in 0..n$ are defined by:

$$k_j(x) = \begin{cases} 1 - \alpha \cdot \frac{\sigma(x)}{\max_{x \in \mathcal{X}} \sigma(x)} & \text{if } j = l \text{ (correct class)} \\ \alpha \cdot \frac{\sigma(x)}{\max_{x \in \mathcal{X}} \sigma(x)} & \text{if } j = n \text{ (outlier class)} \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

The uncertainty defines the probability of the outlier class, and the remaining probability is assigned to the correct class. Parameter α weights the importance given to the uncertainty measure. When training the distilled model, the temperature of its softmax may be raised in accordance to the value of α. A penalty is then added to the training loss to prevent the logits' from exploding. At test time, the model always infers with a normal temperature of $T = 1$.

4. Evaluation

We evaluate the approach against attacks mounted both in *white-box* and *black-box* threat models. We experiment with MNIST [11], using the model and attacks provided in the cleverhans v.1 library. The strength and weakness of the approach are characterized by three rates:

- *Recovered*: percentage of adversarial inputs whose original class (i.e., correct label) was recovered.
- *Detected*: percentage of adversarial examples classified in the model's outlier class.
- *Misclassified*: all other adversarial examples.

The distilled model was trained with $N = 20$ dropout passes for 50 epochs at temperature $T = 4$ and a variance coefficient $\alpha = 0.9$. Its accuracy on legitimate test inputs is 98.63%, compared to a 99.19% accuracy for the same architecture trained without defensive distillation.

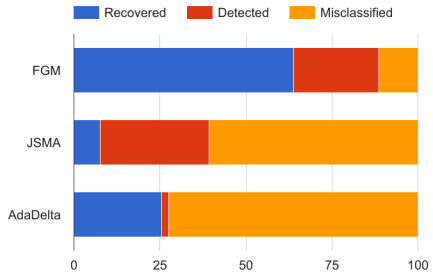
In Figure 1a, we attack the distilled model directly with the FGSM ($\varepsilon = 0.20$) [3], JSMA ($\Gamma = 7\%$) [7] and AdaDelta ($\kappa = 7$) [9] strategies. A large fraction of adversarial examples is either recovered (i.e., correctly classified) or detected (i.e., classified as outliers). In comparison, the undefended model misclassifies at 87.1%, 82.8% and 97.6% on the FGM, JSMA and AdaDelta attacks. The JSMA and AdaDelta algorithms are more successful at evading the defended model than the FGM. Although defensive distillation does not completely prevent white-box attacks, it increases the adversary's effort: for $\varepsilon, \Gamma, \kappa$ parameters values smaller to the ones used in Figure 1a, the defense detects or recovers 60% to 90% of the adversarial examples produced by the FGM, JSMA and AdaDelta.

We mount worst-case black-box attacks through transferability from a surrogate model trained with the same architecture and data. We consider two types of surrogate models: undefended (U) and defended (D). This allows us to test for gradient masking (which would be indicated by strong transferability of adversarial examples from the surrogates to the distilled model). Our results, reported in Figure 1b, are comparable with those obtained in the white-box setting (see Figure 1a). One notable exception is the large difference for the FGM attack from the undefended (U) and defended (D) surrogates.

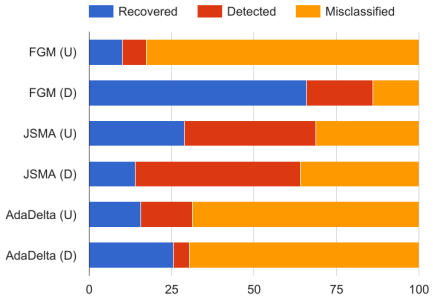
5. Conclusions

The defensive distillation variant proposed defends models in a comparable capacity in the face of white-box and black-box attacks. This indicates that the defense does not suffer from gradient masking. While the effectiveness of the defense recedes as the adversary increases the perturbation, it is able to improve the robustness in a vicinity of the training data while paying a modest price in accuracy.

The approach's most appealing aspect remains that it does not require that the defender generate adversarial examples. This leaves room for another line of work combining defensive distillation with other defenses. Although the method is generic and applicable to any neural network and input domain, the evaluation performed remains preliminary given that our experimental setup is restricted to MNIST.



(a) White-box FGSM, JSMA and AdaDelta



(b) Black-box FGSM, JSMA and AdaDelta

Figure 1: Recovered, detected and misclassified rates for white-box and black-box attacks against the distilled model. Attacks marked with U are computed on an undefended surrogate model and D the distilled model from Section 3.

Nicolas Papernot is supported by a Google PhD Fellowship in Security. Research was supported in part by the Army Research Laboratory, under Cooperative Agreement Number W911NF-13-2-0045 (ARL Cyber Security CRA), and the Army Research Office under grant W911NF-13-1-0421.

References

[1] C. Szegedy *et al.*, "Intriguing properties of neural networks," *arXiv preprint arXiv:1312.6199*, 2013.

[2] B. Biggio *et al.*, "Evasion attacks against machine learning at test time," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2013, pp. 387–402.

[3] I. J. Goodfellow *et al.*, "Explaining and harnessing adversarial examples," *arXiv preprint arXiv:1412.6572*, 2014.

[4] N. Papernot *et al.*, "Distillation as a defense to adversarial perturbations against deep neural networks," in *Security and Privacy (SP), 2016 IEEE Symposium on*. IEEE, 2016, pp. 582–597.

[5] K. Grosse *et al.*, "On the (statistical) detection of adversarial examples," *arXiv preprint arXiv:1702.06280*, 2017.

[6] H. Hosseini *et al.*, "Blocking transferability of adversarial examples in black-box learning systems," *arXiv preprint arXiv:1703.04318*, 2017.

[7] N. Papernot *et al.*, "The limitations of deep learning in adversarial settings," in *Security and Privacy (EuroS&P), 2016 IEEE European Symposium on*. IEEE, 2016, pp. 372–387.

[8] N. Papernot et al., "Practical black-box attacks against machine learning," in *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security*. ACM, 2017, pp. 506–519.

[9] N. Carlini *et al.*, "Towards evaluating the robustness of neural networks," *arXiv preprint arXiv:1608.04644*, 2016.

[10] Y. Gal *et al.*, "Dropout as a bayesian approximation: Representing model uncertainty in deep learning," *arXiv preprint arXiv:1506.02142*, vol. 2, 2015.

[11] Y. LeCun *et al.*, "The mnist database of handwritten digits," 1998.