Bridging machine learning and cryptography in defence against adversarial attacks

Olga Taran, Shideh Rezaeifar, and Slava Voloshynovskiy [*]

Computer Science Department,
University of Geneva, Geneva, Switzerland
{olga.taran,shideh.rezaeifar,svolos}@unige.ch

Abstract. In the last decade, deep learning algorithms have become very popular thanks to the achieved performance in many machine learning and computer vision tasks. However, most of the deep learning architectures are vulnerable to so called *adversarial examples*. This questions the security of deep neural networks (DNN) for many security- and trust-sensitive domains. The majority of the proposed existing adversarial attacks are based on the differentiability of the DNN cost function. Defence strategies are mostly based on machine learning and signal processing principles that either try to *detect-reject* or *filter* out the adversarial perturbations and completely neglect the classical cryptographic component in the defence.

In this work, we propose a new defence mechanism based on the second Kerckhoffs's cryptographic principle which states that the defence and classification algorithm are supposed to be known, but not the key.

To be compliant with the assumption that the attacker does not have access to the secret key, we will primarily focus on a *gray-box* scenario and do not address a *white-box* one. More particularly, we assume that the attacker does not have direct access to the secret block, but (a) he completely knows the system architecture, (b) he has access to the data used for training and testing and (c) he can observe the output of the classifier for each given input. We show empirically that our system is efficient against most famous state-of-the-art attacks in *black-box* and *gray-box* scenarios.

Keywords: Adversarial attacks, defence, data-independent transform, secret key, cryptography principle.

1 Introduction

In the last decade, scientists achieved a big breakthrough in enhancement of functionality and extension of scope of applications of the DNN. Nowadays, neural networks have become very efficient in many machine-learning tasks. However, despite the remarkable progress the DNN stay vulnerable to adversarial attacks that aim at designing such a perturbation to original samples that, in general,

---

is imperceptible for humans, but it is able to trick the DNN. This vulnerability seriously restricts the usage of the DNN in many security- and trust-sensitive domains.

In recent years, researchers proposed a large number of different defence mechanisms. However, the growing number of defences stimulates the invention of even more universal attacks. This is due to the fact that the overwhelming majority of existing attacking algorithms are based on the principle of end-to-end differentiability of the DNN and possibility to add the modification back to the original spatial domain images.

In this paper, we propose a new defence mechanism for the DNN classifiers based on the second Kerckhoffs's cryptographic principle, that states that the fewer secrets the system contains, the higher its safety [1]. In this regard, the structure of the proposed system is supposed to be known, except the key that is kept secret. This key is used in a security imposing pre-processing block that might be implemented in various ways.

Based on the best cryptographic practice, one can state that there does not exist any secure algorithm that does not contain any secret unknown to the attacker. From this point, the defences against so called *white-box* attacks will unlikely find practical applications besides some rare exceptions. In this respect, we will primarily focus our attention to a *gray-box* scenario that assumes that the attacker has general knowledge about the structure of the system of interest. However, due to the existence of secret parameters/key, in reasonable time, he is capable neither to discover or to estimate these secret elements, even with the help of modern computational means, nor to train the "bypass" system that would give a sufficiently accurate estimation of the secret parameters/key.

We evaluate our defence mechanism on two standard datasets, namely, MNIST [2] and Fashion-MNIST [3] for the *Fast Gradient Sign Method* (FGSM) [4] as the simplest attack case, and attack proposed by N. Carlini and D. Wagner in [5] as the most efficient one for most of the existing adversarial defences mechanisms.

The main contributions of this paper are:

- We analyse the existing state-of-the-art adversarial attacks and most well-known defence algorithms.
- We present a new defence mechanism for DNN classifiers based on cryptographic principles.
- We investigate the efficiency of the proposed approach on two standard datasets for several well-known adversarial attacks.
- We empirically show that a sufficiently simple data-independent transformation, based on a secret key, can serve as a reasonable defence against gradient based adversarial attacks.

Notations. We use small bold letters $\mathbf{x}$ to denote a signal that can be represented either in 1D or 2D format, $\mathbf{x}_i$ corresponds to the ith entry of vector $\mathbf{x}$. $E(.)$ and $D(.)$ denotes the *encoder* and *decoder* parts of the DNN classifier, respectively. $P(.)$ indicates the data-independent transformation operator.

The remainder of this paper is organised as follows: Section 2 briefly summarizes the basic principle of the DNN classifier and gives the general classification of the existing state-of-the-art attacks against the DNN classifiers as well as a brief classification of the existing defence mechanisms. Section 3 introduces the main idea and principles of the proposed defence approach. Section 4 presents the empirical results obtained for the proposed algorithm and their analysis. Finally, Section 5 concludes the paper.

2   Background

2.1   Neural Networks

In general case, it is possible to represent a DNN classifier as a model that consists of two parts: (a) *encoder* or training part and (b) *decoder* or classifier. As an input, the *encoder* takes a multidimensional vector $\mathbf{x} \in \mathbb{R}^{N \times C}$, where C is the number of channels, and outputs a vector $\mathbf{y} \in \mathbb{R}^{M}$, where in most of the cases, M is equal to the number of classes, and each $\mathbf{y}_j$ is treated as a probability that a given input $\mathbf{x}$ belongs to class j. Typically, the *encoder* consists of several nested layers:

$$\mathbf{y} = E(\mathbf{x}) = \sigma_n\bigg(\mathbf{W}_n\sigma_{n-1}\big(...\sigma_1(\mathbf{W}_1\mathbf{x})\big)\bigg),\qquad(1)$$

where, at each layer i, $\mathbf{W}_i$ corresponds to the model parameters and σ_i is an activation function, usually, non-linear, with $1 \leq i \leq n$.

The *decoder* assigns most likely class label based on the result of the *encoder*:

$$D(\mathbf{x}) = \underset{1 \leq j \leq M}{\arg\max}\, E(\mathbf{x})_j \qquad(2)$$

$$= \underset{1 \leq j \leq M}{\arg\max}\, \mathbf{y}_j = \hat{j}.$$

2.2   Attacks against DNN classifiers

Based on the knowledge available to the attacker, the adversarial attacks against the DNN classifiers can be combined into three main groups [6]:

1. *White-box* attacks require the full knowledge of treated model and/or training and test data.
2. *Black-box* attacks, where the attacker (a) has no information about the structure and parameters of the used model and training data, (b) has possibility to observe the class labels assigned to chosen input like as in case of cryptographic oracle.
3. *Gray-box* attacks, where the attacker (a) knows the system architecture, (b) has access to the data used for training and testing, (c) for each given input can observe the assigned class label, (d) does not have access to or knowledge of the defence mechanism parameters.

In turn, based on the goal of the attacker, the attacks can be [6]:

–   *Targeted* that aim at modifying the input in a way that the classifier classifies it as a specified target class. Namely, for a given target class t such that $D(\mathbf{x}) \neq t$, the goal is to find such a perturbation $\mathbf{z}$ that $D(\mathbf{x} + \mathbf{z}) = t$ [5].
–   *Non-targeted* that aim at modifying the input in a way that the classifier classifies it incorrectly. Namely, the goal is to find such a perturbation $\mathbf{z}$ that $D(\mathbf{x} + \mathbf{z}) \neq D(\mathbf{x})$, i.e., the classifier output any wrong class label [5].

It should be pointed out that many adversarial attacks have a transferability property that consists in the fact that adversarial examples trained on one model can be successfully applied to another model with a different architecture of the *encoder* part [6]. In general, targeted adversarial examples are much harder to transfer than non-targeted ones.

Due to the limited interest of *white-box* attacks for real-life applications, in our paper we focus preliminary our attention on the *gray-box* targeted and non-targeted attacks that can be also extended to the *black-box* scenario.

State-of-the-art attacks against DNN classifiers. Without loss of generality, it is possible to group the state-of-the-art adversarial attacks against the DNN classifiers into two main groups:

1. *Gradient* based attacks. The core principle of which consists in the end-to-end differentiability of many neural network classification systems.

    This group comprises L-BFGS attack proposed by Szegedy et al. in [7]. This attack is time-consuming due to the used expensive linear search and, as a consequence, is impractical for real-time applications. However, this attack served as a basis for several more successful attacks such as *Fast Gradient Sign Method* (FGSM) [4]. In contrast to L-BFGS, FGSM is fast, but not all the time gives the minimal adversarial perturbation between original and targeted samples. FGSM method has several successful extensions, like FGSM with momentum [8], *One-step Target Class Method* (OTCM) [9], RAND-FGSM algorithm [10], proposed in [11] *Basic Iterative Method* (BIM) and *Iterative Least-Likely Class Method* (ILLC), etc. In addition, it should also be mentioned the *Jacobian-based Saliency Map Attack* (JSMA) [12] and the *DeepFool* approach [13] with its extension *Universal perturbation* [14]. Moreover, one should note the attack proposed by N. Carlini and D. Wagner in [5]. As it has been shown in many works, like for example in [15] and [16], this attack is among the most efficient ones against many existing defence mechanisms. Finally, A. Athalye et al. in [17] propose Backward Pass Differentiable Approximation technique that aims at avoiding the gradient masking in *white-box* scenario.

2. *Non-gradient* based attacks

   The attacks of this group do not require any knowledge of the DNN gradients. The most well-known members of this group are the *Zeroth Order Optimisation* (ZOO) [18] and the *One Pixel Attack* [19].

2.3   Defence strategies

In general, the existing state-of-the-art defence strategies can be classified into four main groups:

1. *Defence via retraining*
   The most well-known work in this group is *network distillation* proposed by Papernot et al. in [20]. Moreover, a significant number of papers were focused on the investigation of the potential of *adversarial retraining*. The main idea behind this retraining is to use adversarial examples in varying degrees during the network training. Representatives of this approach are the works of Goodfellow et al. [4], Huang et al. [21], Kurakin et al. [9], Wu et al [22], etc.

2. *Defence via detection and rejection*
   Formally, one can distinguish several subclasses, the main ideas behind which can be mixed:
   (a) Based on integration of *additional DNN*, like for example in the work of Metzen et al. [23] the original network is augmented by adding an auxiliary "detector" sub-network that aims at distinguishing the original data from data containing adversarial perturbations.
   (b) *Statistical* based, for example in [24] the authors claim that the adversarial perturbations affect in a special way the lower-ranked principal components from PCA. In [25], the authors perform analysis of the statistics of convolutional layers outputs to detect the adversarial inputs. Feinman et al. in [26] analyse Bayesian uncertainty available in dropout neural networks, etc.
   (c) Based on analysis of *DNN properties*, such as [27] and [28] where the authors make their decision by extending the simplex algorithm to support the non-convex ReLUs activation function and consider the neural network as a whole, without any simplifying assumptions.

3. *Defence via input pre-processing*
   In general case, defence is achievable through filtering and removal of modifications introduced to original images, like for example the denoising used in [29] or recently proposed in [30].

4. *Defence via regeneration*
   The key point behind this approach consists in the assumption that the adversarial examples can be mapped back to the manifold of the original

clean data via regeneration. For example, Gu et al. in [31] propose a *deep contractive autoencoder* that is a variant of the classical autoencoder with an additional penalty increasing the robustness to adversarial examples. Meng et al. in [29] introduce *MagNet*, which combines the detector and regeneration networks.

Despite the big variety of the existing defence mechanisms, there are still several important open issues:

– In the vast majority of cases, the attacker knows exactly the same amount of information or can easily learn any complementary information about the defence strategy. Therefore, the attacker can relatively easy bypass the defence mechanism.
– Usually, there is no information advantage for the defender over the attacker.
– In general, no cryptographic principles are used by the defender.

3   Proposed approach

In this paper, we propose for the first time up to our best knowledge, a defence strategy using a cryptographic formulation. This strategy is based on the next principles:

– *Information advantage of the defender over attacker*: we consider a party consisting of: encoder $E(.)$ and classifier (decoder) $D(.)$ that share a common secret, i.e., secret key that is used in one of the blocks. This party is a defender that plays against the attacker, who does not know the secret key.
– *Data-independence of security imposing transformation*. The transformation used in the security imposing block should be based on a secret key and should be data-independent to avoid possibility to be learned from the training dataset. The entropy of the secret key should be at least as high as the entropy of the signal. In addition, it is desirable, but not absolutely necessary, that the security imposing transformation is non-differentiable. Non-differentiability makes the proposed approach end-to-end non-differentiable and provides the additional difficulties for the attacker.
– *Protection of "internal variables" via the assumed protocol*. To be compliant with most of practical requirements about the deployment of AI we will assume that the attacker has access to the output of the classifier, but can not observe the internal variables of the network to avoid the access to the secret key or its easy learning protocols. This assumption is reasonable assuming that the recognition is done on protected servers or special devices or chips, which is typical in domains as for example biometric or digital watermarking applications facing similar concerns.

One can imagine the extension of similar principles to other applications such as digital forensics, device identification, etc., facing similar deficiencies.
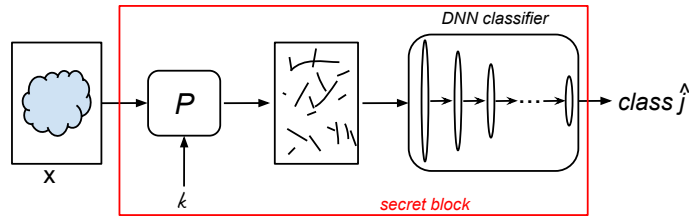
Fig. 1. Principal scheme of the proposed algorithm: input and output are observable variables for the attacker, the secret block architecture is known, but the inner parameters and variables are not observable.

The generalized scheme of the proposed approach is illustrated in Fig. 1 and is quite simple. At first, an input signal $\mathbf{x} \in \mathbb{R}^{N \times C}$ is fed through a specially designed transform block P, where the secret key $k \in \mathcal{K}$ is used. After that, the result comes to the input of the DNN classifier. In general, the architecture of block P and the DNN classifier is supposed to be public. Following the classification given in Section 2.3, the proposed defence can be formally associated to the class of *defences via input pre-processing*. However, in the proposed approach, the pre-processing block P does not assume any filtering or artefact removing. The main requirement to this block consists in the fact that it should be data-independent transformation $P(.)$ based on a secret key that in addition, preferably, but not necessary, can be non-invertible and non-differentiable. Mathematically, for the given input signal $\mathbf{x} \in \mathbb{R}^{N \times C}$ the classification can be defined as:

$$D(\mathbf{x}) = \underset{1 \leq j \leq M}{\arg\max}\, E(P(\mathbf{x}, k))_j, \tag{3}$$

where k is a known secret key.

Requirements to secret elements. Following the second Kerckhoffs' cryptographic principle [1] we assume that all details of our algorithms are publicly known and available to the attacker besides the key. At the same time, taking into account the capabilities of modern computing systems, keeping a key secret can be not so trivial. We assume that if the attacker would have access to the output of transform block P, he can easily discover the key k or make the defence system end-to-end differentiable in his attack by simply replacing the block P by a trained differentiable mapper or applying the Backward Pass Differentiable Approximation technique recently proposed by A. Athalye et al. in [17]. In this regard, we assume the use of some standard measures that restricts the access to the internal results and parameters of the proposed system.

Moreover, in order to protect the system against brute force attacks, theoretically the entropy of the key is supposed to be at least as high as those of the input signal. In practice, it is supposed that the key is used correctly meaning that it is random and is unique (never reused) for each application instance.

Fig. 2. Examples of original images from each class from MNIST (top line) and Fashion-MNIST (bottom line) datasets.

In our case, we will assume that each classifier has its own key as for example each self-driving car equipped by a DNN based recognition system would have an individual classifier parametrized by k. Obviously, in this case, the process of training is more complex in comparison with one common non-secret classifier.

4   Evaluation

4.1   Datasets

Our evaluation strategy starts with more simple examples and evolves to more complex ones. As a simple dataset we used the MNIST set of hand-written digits [2] that contains 10 classes, 60 000 training and 10 000 test grayscale images of the size 28×28. Fashion-MNIST set [3] serves as a dataset with more complex and diverse structure of objects. Similarly to MNIST, Fashion-MNIST consists of 10 classes, 60 000 training and 10 000 test grayscale images of the size 28×28. Examples of images from each dataset are illustrated in Fig. 2.

In case of both datasets, we used 55 000 examples for training and 5 000 for validation from the train set, and the first 1 000 samples from the test set for testing on the original and attacked data.

4.2   Base-line attacks

Guided by the same principle as when selecting databases, we selected two well-known algorithms of adversarial attacks, namely, the *Fast Gradient Sign Method* (FGSM) [4] as a simplest case, and the attack proposed by N. Carlini and D. Wagner in [5] (that will be referred to as CW) as the most efficient for most of existing adversarial defences mechanisms.

Fast Gradient Sign Method. The FGSM method determines for each pixel of input signal $\mathbf{x}$ the direction in which the intensity of pixel should be changed by using the gradient of the loss function [4]:

$$\mathbf{x}^t = \mathbf{x} + \epsilon \cdot sign(\nabla_{\mathbf{x}} J(\mathbf{W}, \mathbf{x}, t)), \qquad (4)$$

where t is a target class, $\mathbf{x}^t$ is the adversarial input, $J(.)$ is the loss function of the DNN classifier, $\mathbf{W}$ denotes the parameters of the model and ϵ is a small constant that controls the level of distortions.

In general, the FGSM attack was developed to be fast, but it does not always produce the minimal required distortions.

CW attack. In contrast to the FGSM, the CW attack proposed in [5] aims at finding for a given input $\mathbf{x} \in \mathbb{R}^{N \times C}$ the adversarial example $\mathbf{x}^t = \mathbf{x} + \boldsymbol{\delta}$ with a minimum possible distortion $\boldsymbol{\delta}$:

$$
\begin{aligned}
&\min_{\delta} \|\boldsymbol{\delta}\|_p + c \cdot f(\mathbf{x} + \boldsymbol{\delta}) \\
&s.t \quad 0 \leq \mathbf{x}_i + \delta_i \leq 1 \qquad \forall i = 1, ..., (N \times C),
\end{aligned}
\tag{5}
$$

where $c > 0$ is a suitable chosen constant, $\boldsymbol{\delta}$ is the desired distortion, $f(.)$ is a new objective function, such that $D(\mathbf{x} + \boldsymbol{\delta}) = t$, if and only if $f(\mathbf{x} + \boldsymbol{\delta}) \leq 0$, $\|.\|_p$ is a ℓ_p norm defined as:

$$
\|\boldsymbol{\delta}\|_p = \left(\sum_{i=1}^{N \times C} |\delta_i|^p \right)^{\frac{1}{p}}.
$$

The authors in [5] investigate several objective functions f and as the most effective they propose:

$$
f(\mathbf{x}^t) = \max \left(\max\{Z(\mathbf{x}^t)_j : j \neq t\} - Z(\mathbf{x}^t)_t, -\kappa \right),
\tag{6}
$$

where Z is the result of the network before the last activation function that, in case of classification, usually is a *softmax*, according to the equation (1), $E(\mathbf{x}) = \sigma_n(Z(\mathbf{x}))$ and κ is a constant that controls the confidence.

4.3   Empirical results and analysis

As it has been described in Section 3, the proposed approach consists of two parts: (1) a security imposing block P and (2) a standard DNN classifier.

In order to follow the best practices of reproducible research, in our experiments we used the FGSM attack implementation from the CleverHans python library[1]. The structure of the DNN classifier used for the generation of this kind of the adversarial examples is given in Table 1. For the generation of the CW adversarial examples we used the code provided by the authors of this attack[2]. The structure of the corresponding DNN classifier is shown in Table 2. In this way we investigate the applicability of proposed defence approach to the different architectures of DNN classifiers.

As it has been indicated in Section 3, the fundamental requirement to the block P is its data-independence and the presence of the secret key k. In general case, one can choose $P(.)$ from a broad family of transformations. In our experiments, in order to validate our theory, we considered the simplest case of $P(.)$ that is a standard random permutation based on the secret key k . For the simplicity of our analysis, we have assumed that the length of the key is equal to the dimensionality of the input data. It should be noted that in the general case it does not ensure that $H(k) \geq H(\mathbf{x})$, where $H(.)$ denotes entropy. However, since all images are normalized between 0 and 1 and posses correlation while the

---

[1] https://github.com/tensorflow/cleverhans
[2] https://github.com/carlini/nn_robust_attacks

Table 1. Architecture of the DNN classifiers used for the FGSM attack.

Layer	# filters	kernel size
Conv2D	64	8×8
ReLu		
Conv2D	128	6×6
ReLu		
Conv2D	128	5×5
ReLu		
Flatten		
Dense		10

Table 2. Architecture of the DNN classifiers used for the CW attack.

Layer	# filters	kernel size
Conv2D	32/64	3×3
ReLu		
Conv2D	32/64	3×3
ReLu		
MaxPool		2×2
Conv2D	64/128	3×3
ReLu		
Conv2D	64/128	3×3
ReLu		
MaxPool		2×2
Flatten		
Dense		200/256
ReLu		
Dropout		
Dense		200/256
ReLu		
Dense		10

key is initialized from i.i.d Gaussian noise $\mathcal{N}(0, 1)$, our assumption is reasonably satisfied. This allows us to reduce to a minimum the number of possible parameters that could affect the results and comprehension of whether the proposed approach can serve as a natural defence against the adversarial attacks.

Before starting the analysis of the obtained results, it should be pointed out that our goal was not to obtain the state-of-the-art classification accuracy on the chosen datasets, but we aim at investigating the possibilities of the proposed defence idea from the point of view of existence of simple yet efficient mechanisms based on proven by practice cryptographic principles for the defence of the DNN classifiers against the adversarial attacks.

The obtained experimental results for the discussed attacks on the chosen datasets are given in Table 3, where "classical classifier" corresponds to the standard DNN classifiers without any defence and with the parameters indicated in Tables 1 - 2. The term "classifier on permuted data" corresponds to the proposed approach with the same standard DNN classifiers.

According to Table 3, in the case of the simple MNIST dataset and original non-attacked input samples, the proposed defence leads to an insignificant decrease of classification accuracy for both tested architectures of the DNN classifiers with respect to the one without defence. As for adversarial attacks, then it should be noted that such a simple defence strategy decreases the classification error from 92 - 100% to 9 - 18% for all types of considered attacks. In case of the

Table 3. Classification error (%) on the first 1 000 test samples

Attack	Classical classifier		Classifier on permuted data	
	original	attacked	original	attacked
MNIST				
CW ℓ_2	1.00	100.00	3.00	8.64
CW ℓ_0	1.00	100.00	3.00	14.53
CW ℓ_∞	1.00	99.99	3.00	12.24
FGSM	1.00	92.10	1.40	18.00
Fashion MNIST				
CW ℓ_2	7.50	100.00	11.50	12.12
CW ℓ_0	7.50	100.00	11.50	13.48
CW ℓ_∞	7.50	99.90	11.50	12.55
FGSM	8.60	60.60	11.20	27.50

Fashion-MNIST dataset, the observed tendency is the same. Thus, the obtained results demonstrate a great potential of the proposed concept and show that even a simple random permutation in the defence block P can serve as a quite promising defence.

From the point of view of security, the attacker can try to guess the secret key via brute force. However, one should take into account that in the proposed approach the length of the used key is at least as the length of the input signal. In case of the (Fashion-) MNIST dataset the size of the input signal is 784, which is sufficiently large, to make the brute force attack practically infeasible. Moreover, taking into account the fact that attacker does not have access to the internal results of the system, he will not be able to train a "bypass" mapper between the input of the system and output of the transform block P in order to avoid the necessity to know the secret key or to apply the Backward Pass Differentiable Approximation technique.

5   Conclusions

In this work our main focus was the defence of the DNN classifiers in *gray-box* scenario as the most suitable for the real-life applications based on the best cryptographic practices. We briefly discussed the state-of-the-art adversarial attacks and most well-known defence algorithms and proposed our vision on a possible classification for each direction. We presented a new defence mechanism based on cryptographic principles that can be applied to many existing DNN classifiers. It should be noted that our goal was not to reach a state-of-the-art classification accuracy on the considered datasets. Our empirical findings showed a quite promising potential of the proposed idea from the point of view of existence of simple yet efficient mechanisms based on proven by practice cryptographic prin-

ciples for the defence of the DNN classifier against the gradient based adversarial attacks.

As a future work we aim at investigating other data-independent transformations that can be used in the proposed algorithm and examining the behaviour of our defence on the more complex datasets, like for example CIFAR-10 [32] and on the non-gradient based adversarial attacks.

References

1. Massey, J.L.: Cryptography: Fundamentals and applications. In: Copies of transparencies, Advanced Technology Seminars. Volume 109. (1993) 119
2. Lecun, Y., Cortes, C., Burges, C.J.: The mnist database of handwritten digits, 2009. URL http://yann. lecun. com/exdb/mnist (2009)
3. Xiao, H., Rasul, K., Vollgraf, R.: Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. arXiv preprint arXiv:1708.07747 (2017)
4. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. arXiv preprint arXiv:1412.6572 (2014)
5. Carlini, N., Wagner, D.: Towards evaluating the robustness of neural networks. In: Security and Privacy (SP), 2017 IEEE Symposium on, IEEE (2017) 39–57
6. Yuan, X., He, P., Zhu, Q., Bhat, R.R., Li, X.: Adversarial examples: Attacks and defenses for deep learning. arXiv preprint arXiv:1712.07107 (2017)
7. Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., Fergus, R.: Intriguing properties of neural networks. arXiv preprint arXiv:1312.6199 (2013)
8. Dong, Y., Liao, F., Pang, T., Su, H., Hu, X., Li, J., Zhu, J.: Boosting adversarial attacks with momentum
9. Kurakin, A., Goodfellow, I., Bengio, S.: Adversarial machine learning at scale. arXiv preprint arXiv:1611.01236 (2016)
10. Tramèr, F., Kurakin, A., Papernot, N., Boneh, D., McDaniel, P.: Ensemble adversarial training: Attacks and defenses. arXiv preprint arXiv:1705.07204 (2017)
11. Kurakin, A., Goodfellow, I., Bengio, S.: Adversarial examples in the physical world. arXiv preprint arXiv:1607.02533 (2016)
12. Papernot, N., McDaniel, P., Jha, S., Fredrikson, M., Celik, Z.B., Swami, A.: The limitations of deep learning in adversarial settings. In: Security and Privacy (EuroS&P), 2016 IEEE European Symposium on, IEEE (2016) 372–387
13. Moosavi Dezfooli, S.M., Fawzi, A., Frossard, P.: Deepfool: a simple and accurate method to fool deep neural networks. In: Proceedings of 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Number EPFL-CONF-218057 (2016)
14. Moosavi-Dezfooli, S.M., Fawzi, A., Fawzi, O., Frossard, P.: Universal adversarial perturbations. arXiv preprint (2017)
15. Carlini, N., Wagner, D.: Adversarial examples are not easily detected: Bypassing ten detection methods. In: Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security, ACM (2017) 3–14
16. He, W., Wei, J., Chen, X., Carlini, N., Song, D.: Adversarial example defenses: Ensembles of weak defenses are not strong. arXiv preprint arXiv:1706.04701 (2017)
17. Athalye, A., Carlini, N., Wagner, D.: Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. arXiv preprint arXiv:1802.00420 (2018)

18. Chen, P.Y., Zhang, H., Sharma, Y., Yi, J., Hsieh, C.J.: 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, ACM (2017) 15–26
19. Su, J., Vargas, D.V., Kouichi, S.: One pixel attack for fooling deep neural networks. arXiv preprint arXiv:1710.08864 (2017)
20. Papernot, N., McDaniel, P., Wu, X., Jha, S., Swami, A.: Distillation as a defense to adversarial perturbations against deep neural networks. In: Security and Privacy (SP), 2016 IEEE Symposium on, IEEE (2016) 582–597
21. Huang, R., Xu, B., Schuurmans, D., Szepesvári, C.: Learning with a strong adversary. arXiv preprint arXiv:1511.03034 (2015)
22. Wu, Y., Bamman, D., Russell, S.: Adversarial training for relation extraction. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing. (2017) 1778–1783
23. Metzen, J.H., Genewein, T., Fischer, V., Bischoff, B.: On detecting adversarial perturbations. arXiv preprint arXiv:1702.04267 (2017)
24. Hendrycks, D., Gimpel, K.: Early methods for detecting adversarial images. (2017)
25. Li, X., Li, F.: Adversarial examples detection in deep networks with convolutional filter statistics. CoRR, abs/1612.07767 **7** (2016)
26. Feinman, R., Curtin, R.R., Shintre, S., Gardner, A.B.: Detecting adversarial samples from artifacts. arXiv preprint arXiv:1703.00410 (2017)
27. Katz, G., Barrett, C., Dill, D.L., Julian, K., Kochenderfer, M.J.: Reluplex: An efficient smt solver for verifying deep neural networks. In: International Conference on Computer Aided Verification, Springer (2017) 97–117
28. Katz, G., Barrett, C., Dill, D.L., Julian, K., Kochenderfer, M.J.: Towards proving the adversarial robustness of deep neural networks. arXiv preprint arXiv:1709.02802 (2017)
29. Meng, D., Chen, H.: Magnet: a two-pronged defense against adversarial examples. In: Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, ACM (2017) 135–147
30. Lee, S., Lee, J.: Defensive denoising methods against adversarial attack. (2018)
31. Gu, S., Rigazio, L.: Towards deep neural network architectures robust to adversarial examples. arXiv preprint arXiv:1412.5068 (2014)
32. Krizhevsky, A., Nair, V., Hinton, G.: The cifar-10 dataset. online: http://www. cs. toronto. edu/ kriz/cifar. html (2014)