Sun et al. *Cybersecurity* (2018) 1:9
https://doi.org/10.1186/s42400-018-0012-9

Cybersecurity

SURVEY Open Access

A survey of practical adversarial example attacks

Lu Sun, Mingtian Tan and Zhe Zhou*

Abstract

Adversarial examples revealed the weakness of machine learning techniques in terms of robustness, which moreover inspired adversaries to make use of the weakness to attack systems employing machine learning. Existing researches covered the methodologies of adversarial example generation, the root reason of the existence of adversarial examples, and some defense schemes. However practical attack against real world systems did not appear until recent, mainly because of the difficulty in injecting a artificially generated example into the model behind the hosting system without breaking the integrity. Recent case study works against face recognition systems and road sign recognition systems finally abridged the gap between theoretical adversarial example generation methodologies and practical attack schemes against real systems. To guide future research in defending adversarial examples in the real world, we formalize the threat model for practical attacks with adversarial examples, and also analyze the restrictions and key procedures for launching real world adversarial example attacks.

Keywords: AI systems security, Adversarial examples, Attacks

Introduction

Artificial intelligence (AI) is quickly permeating into our daily life, snatching working opportunities from, but performing better and more efficiently than human being. Specifically, AI is blossoming in many and increasingly more fields, from robotic trading to intelligent diagnosis, from advertisement recommendation to autonomous driving. Unsurprisingly, it also has been applied to system security related areas, like spam filtering, face authentication, etc. AI intrudes those fields and emancipates those professionals from brain works, but also outperforms the professionals in many ways, making it preferred and increasingly more widely accepted.

Machine learning (ML), the *de facto* approach to achieve artificial intelligence, provides a convenient way for AI practitioners to rapidly implant intelligence to machines, with the help of labeled data, and without needing to make clear the logics and theory behind data. All in a sudden, the convenient approach was acquired by professionals in nearly every fields. People continually collect data from their users, train machine learning models using the collected data and pack the trained models to their products to provide better service to their users. The intelligent service, in turn, attracts more users and usages, and simultaneously provides more data to refine the machine learning models, resulting in a virtuous circle that absorbing users and practitioners.

Deep neural networks (DNN), an algorithm class for machine learning with breakthrough in accuracy, gained a great success in fields like image processing, natural language processing etc, however has security risks. For a lot of problems, solutions employing DNN outperform human beings Sun et al. (2014), making DNNs so popular. A key factor of such an achievement is that DNN features a cascade of many layers, which makes the neurons inside the network very hard to be understood even by their designers. Though the indigestibility does not affect its wide application, it indeed increased the difficulty for researchers to analyze the vulnerabilities of DNN models and fortify the security. What's worse, adversaries may make use of those vulnerabilities to attack DNN models.

A huge risk results from crafted malicious inputs. An assumption of DNN is that the test data fall to the same distribution of the training data. Therefore, it is not surprising that the output of a model for data from a

*Correspondence: zhouzhe@fudan.edu.cn
Fudan University, Shanghai, People's Republic of China

deviated distribution is prone to be unpredictable, especially when the model is not specially treated for security protections. This observation was be exploited by adversaries to craft artificially generated inputs that mislead DNN models to targeted outputs, in which case the inputs were called *adversarial examples*. Adversarial examples were firstly noticed and formally defined in Szegedy et al. (2013), when the authors found that the mappings of DNNs are so discontinuous. They designed an optimization approach to search perturbations such that a natural input adding a small perturbation together can lead the model to output differently from when the input is the natural input only. The perturbation can be small enough, so human beings can barely notice the existence of the perturbation.

Research around adversarial examples developed from different directions, including defenses against adversarial examples or attacks with the examples. Some works focus on the defense mechanism to avoid the generation of adversarial examples, while some others aim at designing algorithms to generate examples satisfying all kinds of requirements. Researchers defend adversarial examples mainly by masking the gradient, through which adversaries' optimizers are expected to fail to move toward malicious. However, this mechanism was proven to be null and void, as a bunch of works got around this kind of protection and successfully generated effective examples, mainly by training substitutional models to remove the mask. This is even true for recent works (Buckman et al. 2018; Ma et al. 2018; Guo et al. 2017; Dhillon et al. 2018; Xie et al. 2017; Song et al. 2017; Samangouei et al. 2018). For the second direction, researchers proposed different norms to measure the conspicuousness of perturbation (Papernot et al. 2015; Nguyen et al. 2015; Goodfellow et al. 2014).

There are still technical barriers between a generated adversarial example and a successful exploit to a system. Even for commercially deployed models, it is not difficult for an attacker to generate effective adversarial examples with the help of gradient descending optimizers. However, there are only several works where real world systems were cracked because of adversarial examples, mainly because that only little research concerned how an example can be input to the target model, which usually resides inside the target system without direct interface to attackers. To have a successful and practical attack, attackers must mount the worked out perturbation to construct an example, which is usually difficult for different scenarios.

The largest challenge for practical adversarial example attackers lie in that the input interface of the target model does not expose to adversaries. The adversarial examples calculated by adversaries require pixel modifications to input images. However, in a lot of practical cases, it is impossible for the attacker to find an interface to inject the perturbed image to the model inside the target system, so the adversarial examples though can be generated while cannot be directly used for attacks. For example, one can calculate perturbation of only some pixels for a face image, but an attacker does not know how to make up herself to feed such an pixel level modified image to the target face authentication system. It is obviously less practical to find an interface to directly inject the image to the model behind the face authentication system.

To shed light on future working direction, we completed a comprehensive survey on existing physical and practical adversarial example attacks, including two attacks against face recognition models, one for road sign models. The only existing works against real systems concerning adversarial example share a lot in their assumptions, methods and restrictions, and nonetheless conquered different challenges when facing different systems. In this survey, we summarize their works and abstract the threat model their attacks shared in common, and formalize the key techniques a success adversarial example attack should have. We also propose possible future work direction in our opinion.

In this survey, we firstly introduce some background knowledge about adversarial examples. Then we compare the theoretical adversarial example attack model and that of the practical one. Following we introduce the routines existing practical adversarial example attacks mainly follow. At last, we show some existing practical adversarial example works.

Adversarial example

The concept of adversarial example was first mentioned and defined in Szegedy et al. (2013). In that paper, the authors found two counter-intuitive properties of neural networks. One of the properties is that the input-output mappings of DNN is fairly discontinuous to a significant extent. Therefore, the author made the network misclassify the images by applying certain hardly perceptible perturbations. And the mixture of original inputs with a little perturbation in order to maximize the prediction error is so-called adversarial example.

Figure 1 illustrate an adversarial example that mislead a machine learning model to recognize a photo of a dog with mixture of perturbations as a cat.

Generating methods of adversarial examples

Finding such adversarial examples for a given model is a key step. In early times, researchers proposed to use only one iteration to generate adversarial examples Goodfellow et al. (2014). Nowadays, the adversarial example searching problem can be reduced to the following optimization

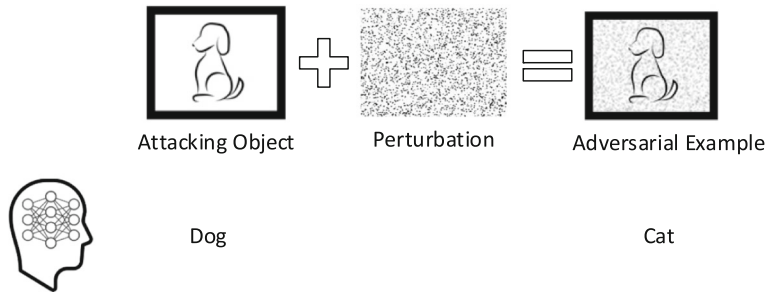

Fig. 1 Adversarial example illustration

problem, with the objective function defined for malicious purpose.

$$\arg\min_{r} J(f(x+r), y) + QualityPenalty(r) \quad (1)$$

Where J is the loss function measuring the distance between outputs of the model. f represents the model and y is the target output. r is the perturbation and $x + r$ is the adversarial example. Quality Penalty is usually a kind of norm of r, which will be introduced later.

The optimization problem can be solved with a gradient optimizer. After an optimizer having minimized the objective, the example $x + r$ could be regarded as adversarial if only the loss fall below a given threshold, indicating that the output of the model is close enough to the attacker's target when she inputs the adversarial example to the model. The state of the art adversarial example generating method is C&W's Carlini and Wagner (2017b). There are also adversarial example generating methods without the help of gradient optimizers (Su et al. 2017). Even when there is no details about the target model, adversaries can still generate adversarial examples Bhagoji et al. (2017).

The author of Yuan et al. (2017) summarized main stream adversarial example generating method.

Distance metrics of adversarial perturbations

The quality of the generated perturbations can be weighted by the norms of the perturbation. When searching adversarial perturbations, researchers mainly use three distance metrics L_0, L_2, L_∞ to weight the quality.

Minimizing different distances results in different perturbations. For example, minimizing L_0 can get perturbations with minimum number of pixels differing from those on the original input. And Jacobian-based Saliency Map (JSMA) Papernot et al. (2015) is an instance for it. Minimizing L_2 helps adversaries obtain perturbations that have the minimum norm, in terms of Euclidean distance, across all pixels. Using this metric, Nguyen et al. (2015) proposed an interesting attack that adds perturbations on a blank image to fool recognition systems. Besides, L_∞ helps finding perturbations with the smallest maximum-change to pixels. Under this metric, the adversary is allowed to freely make changes to pixels if only no change exceeds the L_∞ distance. An example of this kind of attack is Fast Gradient Sign Method (FGSM) Goodfellow et al. (2014), which iteratively updates perturbations by stepping away a small stride along with the direction of the gradient.

Detecting adversarial examples

Realized the consequences of adversarial examples, researchers proposed to detect adversarial examples.

People proposed to detect adversarial examples by introducing an extra classifier to pick out adversarial examples, which is called adversarial training, recent related works include Gong et al. (2017). In this kind of defense, people should generate a set of adversarial example for the model to be protected, and then train an adversarial example detector with the generated adversarial examples.

People also proposed to remove the gradient of the model to be protected Papernot et al. (2016), which is called defensive distillation. This method prevents adversaries from generating adversarial examples with gradient optimizers.

Recent researches also uses the noise levels of the given input to pick out adversarial examples Meng and Chen (2017). The authors found that adversarial examples carry a higher recovery error when passing through an autoencoder trained with all normal examples. Or, for some other cases, the noise removed example results in a totally different model output.

These methods, however, all turned to not effective enough by Carlini and Wagner (2017a), mainly because adversaries can adapt their adversarial example generating methods accordingly.

Practical adversarial example attacks

Although it's hard to defend adversarial examples, limited effort has been made on practical adversarial learning. The reason is that the attacker usually can only change the input on a limited degree to the system, which accounts for attacker's little access to the system device.

However, it is easy to imagine that it would be very dangerous if real world systems can be compromised by attackers with adversarial examples, if only the systems employs ML models,especially when the attacker didn't break into the system. For instance, attackers may freely pass face authentication based entrance access doors if the face authentication models were compromised. Autonomous vehicles may overspeed if the road sign recognition models inside were compromised.

Threat models

An adversarial example generated by the aforementioned methods cannot be directly used to attack a real world system, because of their utterly different threat models. In this section, we compare the their models in detail and hight light the model used for practical adversarial example attacks.

Threat model for theoretical attacks

White box attacks.

As shown by Fig. 2, for theoretical attacks, it is assumed that the model to be attacked is trained, fixed and directly exposed to attackers. Therefore, an attacker can generate perturbations by minimizing the loss over the model between the target (the cat) and the sum of the attacking object (the dog) and the perturbation.

The attack is thought to be successful when the output of the model is indeed the target in stead of the attacking object, regardless of how the perturbation is added into the image of attacking object.

Black box attacks.

In some cases, the model is not transparently exposed to attackers, in which case the attacker can only query the model with images input and get the result returned. In this setting, the attacker still only needs to work out a perturbation for the target, such that the model outputs the target Bhagoji et al. (2017).

Untergeted attacks

For some scenarios, attackers do not have a specific target that must be output by the victim model. Instead, they only want the output is not correct, i.e., output whatever rather than dogs. This kind of attack was called untargeted attacks.

In this case, attackers need only maximize the distance between the perturbed output and the authentic label.

Practical adversarial example attack threat model

Different from the threat model of theoretical adversarial example attacks, in practical attack scenarios, a model should be assumed with more restrictions.

As shown by Fig. 3, the biggest difference lies in that the input of model is not directly exposed to attackers. Instead, the system using the model exposes its capture interfaces to attackers. In this case, attackers cannot break the integrity of the system to directly inject input data to the model inside. Attackers can only manipulate the object in front of the camera (the dog). The goal is still the same: wanting the output of the system to be the target chosen by the attacker (cat).

For this scenario, attackers even face more challenges: the system may impose detection modules between the model and the front end input camera to detect potential attacks. For instance, in a face authentication system, there exists liveness detection modules to examine if the object in front of the camera is a live human being or a printed photo. Considering those mechanisms, attackers usually firstly place a object that can pass the detection and then apply small perturbations that won't fail the front end examinations.

Similar to theoretical attacks, practical attacks may also differ in black box and white box settings. usually, researchers assume white box settings first where the model structure and weights are known to attackers and come up with supporting black box extension to ease the assumption. For instance, attackers may firstly train a substitutional model by querying the black box. Then they can generate adversarial examples for the substitutional mode, which by expectation will also be valid for the model inside the black box Papernot et al. (2016).

Attack routines

The known practical adversarial example attacks also employ optimizers to work out perturbations, but with more restrictions related to implementation considerations. A practical adversarial example attack consists of several steps: 1) Constructing loss function for optimizers. 2) Adversarial example Searching and implementation.

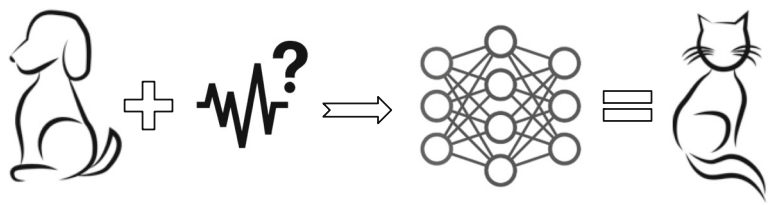

Fig. 2 Threat model for theoretical adversarial example attack

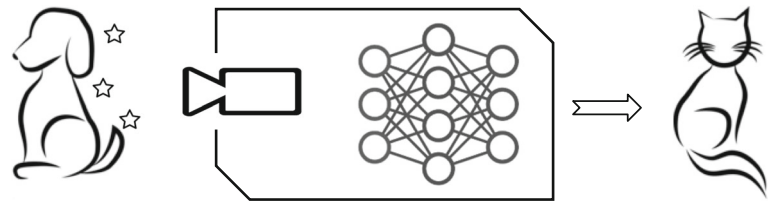

Fig. 3 Threat model for practical adversarial example attack

We first introduce the restrictions should be taken into consideration in a practical attacking setting and then illustrate how attackers can construct schemes to work out and implement perturbations that under those restrictions.

Attack restrictions

We firstly introduce some restrictions that may met in practical scenarios but was not considered in theoretical adversarial example cases.

Printability.

The first issue the attacker should pay attention to is how perturbations can be presented on the attacking object. In theoretical settings, perturbation pixels are assumed to be directly laid over the image of the attacking object. However, attackers cannot overlay skins with arbitrarily color on the attacking object, when launching attacks in the real world. As a result, adversarial example generating method not limiting the color of perturbation cannot be directly borrowed to launch practical attacks.

To counter the color issue, attackers must restrict the perturbations generated to be printable on the attacking object. The restriction is also tightly related to the method the attacker is going to deploy perturbation. For instance, the colors a perturbation has must be inside the color triangle of the printer, if the attacker is planing to print the perturbation into a sticker and paste the sticker on the attacking object. The perturbation should be of the specific color if the attacker is going to modulate perturbations by using mono color light source.

To restrict the perturbation to be printable. The attacker could add to the loss an item representing the difficulty of printing the perturbation, which is firstly defined and called in Sharif et al. (2016) as NPS (non-printability score), shown in Eq. 2. According to its definition , the value will be low when the pixel $\hat{p}$ is close to a printable color p chosen from the printable color set P. Evtimov et al. (2017) later upgraded it to generate smooth road sign adversarial example.

$$NPS(\hat{p}) = \prod_{p \in P} |\hat{p} - p| \tag{2}$$

With the item in loss, the optimizer will try its best to generate perturbations that is prone to printable.

Perturbation precision

In theoretical adversarial example generating schemes, example pixels can be independently modified. However, in real world, when launching attacks, attackers can hardly precisely control the perturbed image captured by cameras in a pixel level accuracy without breaking the integrity of the victim system. Usually, the attackers must restrict the smoothness (precision) of the perturbation generated by the optimizer.

To get around the pixel level precision issue, attacks may introduce *Smoothing* restrictions to the examples to be generated. This can be achieved by also adding a factor measuring the variance of the perturbations, which is called and defined as TV in Mahendran and Vedaldi (2015). Sharif et al. (2016) employed this value to restrict the smoothness of their perturbations.

$$TV(r) = \sum_{i,j} \left((r_{i,j} - r_{i+1,j})^2 + (r_{i,j} - r_{i,j+1})^2 \right)^{\frac{1}{2}} \tag{3}$$

The adversary may employ a model to produce fully smooth perturbation without pixel precision problem. For example, Zhou et al. (2018) designed an infrared dot model to make sure the perturbation is Gaussian smooth. This may be extended to help attackers to bypass more restrictions. Specifically, attackers may also construct a model describing the printable perturbations with parameters outlet to optimizers. In this way, the optimizers will no longer optimize the loss over the pixels of an attack image. Instead, it will optimize over the parameters of the model. The model should guarantee that the generated perturbations are always implementable if only the parameters are in the correct range. The idea is pretty similar with C&W's clipped and smooth function i.e., tanh Carlini and Wagner (2017b).

$$r = model(position, shape, ...) \tag{4}$$

Attack procedures

1. To design a practical adversarial example attack against a real world system, the attacker should firstly design a perturbation mounting scheme. For

instance, he can print perturbation on stickers and paste the sticker on attacking object. He can also use a projector to project perturbations on attacking object.

2. Then he need construct a loss function that fools the target model while at the same time measures the implementability.

$$\arg\min_{r} J(x+r, y) + \sum_{i} Penalty_i(r) \tag{5}$$

Where the function J represent the loss (or distance) between the adversarial example and the target from the perspective of the target model. Each penalty function weights the difficulty of implementing the perturbation under each restriction.

3. The attack chooses an attacking object x and his target y, after which he can run a gradient optimizer to work out a perturbation r.
4. The attacker can print the solved perturbation into using his mounting method and start to attack the system.

Existing practical attacks

Adversarial examples worked out by optimizers were firstly found to be also recognizable by models in physical world in 2016 Kurakin et al. (2016), where a photo of washer was printed on a paper and the paper was recognized by a classification model as safe or loudspeaker. Figure 4 illustrates this kind of attack.

Though it is not an attack to a real world system, it did inspire researchers to explore ways to attack real systems using adversarial examples. We introduce some attacks using physical adversarial examples according to their perturbation mounting methods.

Eyeglasses frame

Sharif et al. proposed a scheme to attack face classification system Sharif et al. (2016) in 2016 for both targeted and untargeted, white box and black box. They have two goals:

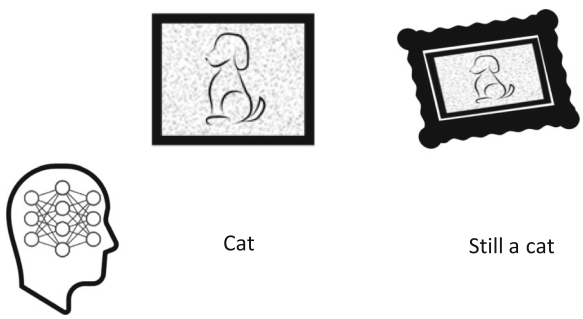

Fig. 4 Physical adversarial illustration

dodging and impersonation. For the dodging, the classification model fails to class the attacker as the attacker while for the impersonation, the model classes the attacker as another someone who is specified by the attacker.

For the dodging attacks, they used a gradient descend algorithm to maximize the softmaxloss between the perturbed attacker and himself, while for the impersonation, they minimize the loss between the attacker and the victim target. The loss was also added penalties for printability, smoothness, robustness.

With the help of the gradient descend algorithm, An attacker can easily get the values of the pixels on the frame, which can then be printed on paper frames by a commercial printer. In theory, the attacker gets a successful attack when the cross entropy part of the loss is optimized to below (for impersonation) or over (for dodging) a threshold. The attacker needs only wear the frame like wearing glasses and sitting in front of the camera of the target system. The system will take photos for the attacker and pass the photo to back end photo for prediction. The photo with no doubt contains not only the attacking object but also perturbations, which results to the model's misleading. By expectation, the model will classify the photo with the attacker's willing, as the expected cross entropy is already satisfied. Figure 5 illustrate this kind of attack.

As a extension of their main work, they proposed a query based method and employed a Particle Swarm optimizer to attack classification model in a black box model.

Road sign

Evtimov et al. proposed a white box adversarial example attack against their own trained road sign recognition models Evtimov et al. (2017). They trained several CNN models, including LISA-CNN and GTSRB-CNN models, to recognize road signs, which then were used as target models. They proposed two kinds of perturbation mounting methods for the road sign scenario, i.e., poster and sticker, which are both proved as effective, according to their experiments. They followed Sharif et al. (2016)'s method to construct loss functions which also considered printability and restricted the mounting positions.

Their scenario slightly differs from previous works because they need consider the varying viewing angles for their drive-by requirements. Surprisingly, their evaluation results showed that they got a 100% attack success rate for the drive-by experiments with the poster mounting method, as Fig. 6 shows.

Infrared

Another interesting practical adversarial example attack was researched by Zhou et al. (2018). They discovered that infrared, which is totally invisible to human beings, can also be used to mount perturbations. As shown by Fig. 7, attackers can mount some infrared LEDs on the cap

Fig. 5 Impersonation attacks with eyeglasses

peak, which can light the attacker's face with some dots from the perspective of cameras, but cannot be noticed by nearby people. With this technique, they successfully attacked face authentication system in a white box setting.

Different from previous two works, they didn't directly optimize perturbation. they built a model describing the infrared dot produced by LED lights with positions, radius and brightnesses as parameters. Therefore, optimizers can optimize over the parameters to precisely search perturbations close to real infrared dots.

They also developed a real time feedback system for attackers to adjust the positions of the LEDs to help them better implement perturbations. This work also differs from previous two in that they don't need to print perturbation for each target. Instead, attackers need only adjust the device to attack different targets.

Future work directions

We believe that practical adversarial example attacks will outbreak in the near future, with the permeation of machine learning into our life. We think future research could be conducted through the following two directions.

Mounting methods

In different scenarios, attackers need different methods to deploy perturbations, which is the key step for practical adversarial example attacks. It's mainly due to the lack of a kind of generic perturbation mounting method.

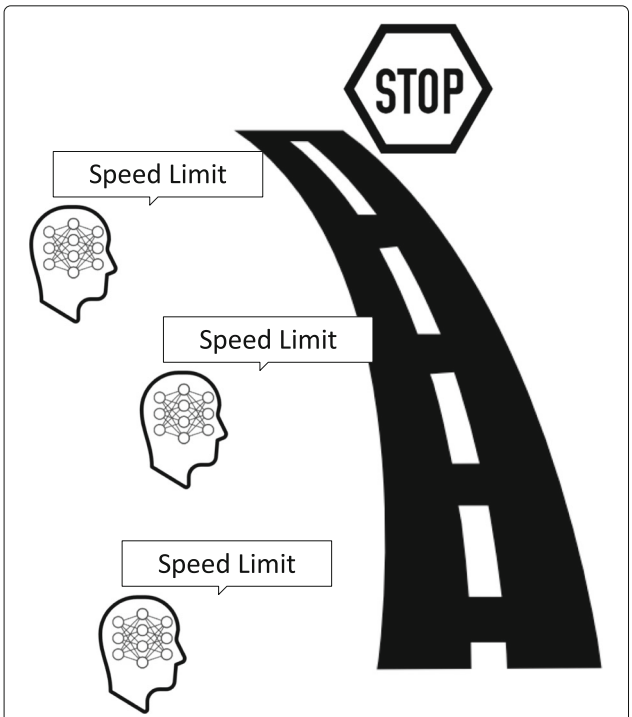

Fig. 6 Adversarial road signs valid in different view of angle

Fig. 7 The cap mounting LEDs to launch attacks

To avoid future outbreak of physical adversarial example attacks, researchers could make clear the possible perturbation mounting methods, and thereby can devise countermeasures accordingly.

Detection methods

Defending adversarial examples from the model side is still an unsolved problem. Researcher tried all kinds of method but they rarely take effect Athalye et al. (2018), because of the nature of machine learning.

We believe methods less relying on machine learning could be useful in detecting adversarial examples. Therefore, people can put the detection in between the image capturing and the model to tick out adversarial example attacks. For instance, liveness detection method may help systems employing face recognition model to find unusual stuffs on face like the frames or other printed objects.

Conclusion

Protections on systems with AI are still far from enough, as recent three pieces of attack we surveyed showed. Attackers can easily generate perturbations by upgrading theoretical adversarial example generating methods. More importantly, they can devise perturbation mounting schemes for specific scenarios, so that attackers can mount the generated perturbations to attack real world systems. We surveyed their works, abstracted their models and proposed direction for future works.

We believe the consequences of practical adversarial example attacks would be severe if the principles behind the attacks are not made clear. To avoid so, researchers must untangle all possible perturbation mounting vectors and system designers must attach enough attention to adversarial examples when integrating AI models.

Acknowledgments

We thanks any the reviewers for your precious time and comments. To clearly illustrate the works we surveyed, we directly used the original equations in the relevant referenced works.

Funding

This work is partially sponsored by Shanghai Sailing Program No.18YF1402200.

Authors' contributions

All authors read and approved the final manuscript.

Competing interests

The authors declare that they have no competing interests.

Publisher's Note

Received: 14 August 2018 Accepted: 15 August 2018
Published online: 06 September 2018

References

Athalye A, Carlini N, Wagner DA (2018) Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. CoRR abs/1802.00420:1–12. http://arxiv.org/abs/1802.00420

Bhagoji AN, He W, Li B, Song D (2017) Exploring the space of black-box attacks on deep neural networks. arXiv preprint arXiv:1712.09491

Buckman J, Roy A, Raffel C, Goodfellow I (2018) Thermometer encoding: One hot way to resist adversarial examples. In: International Conference on Learning Representations

Carlini N, Wagner D (2017a) Adversarial examples are not easily detected: Bypassing ten detection methods. In: Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security. ACM, Dallas. pp 3–14

Carlini N, Wagner D (2017b) Towards evaluating the robustness of neural networks. In: 2017 38th IEEE Symposium on Security and Privacy (SP). IEEE, San Jose. pp 39–57

Dhillon GS, Azizzadenesheli K, Lipton ZC, Bernstein J, Kossaifi J, Khanna A, Anandkumar A (2018) Stochastic activation pruning for robust adversarial defense. arXiv preprint arXiv:1803.01442

Evtimov I, Eykholt K, Fernandes E, Kohno T, Li B, Prakash A, Rahmati A, Song D (2017) Robust physical-world attacks on deep learning models. CoRR abs/1707.08945:1–11. http://arxiv.org/abs/1707.08945

Gong Z, Wang W, Ku WS (2017) Adversarial and clean data are not twins. arXiv preprint arXiv:1704.04960

Guo C, Rana M, Cissé M, van der Maaten L (2017) Countering adversarial images using input transformations. arXiv preprint arXiv:1711.00117

Goodfellow IJ, Shlens J, Szegedy C (2014) Explaining and Harnessing Adversarial Examples. ArXiv e-prints:1–11

Kurakin A, Goodfellow I, Bengio S (2016) Adversarial examples in the physical world. arXiv preprint arXiv:1607.02533

Ma X, Li B, Wang Y, Erfani SM, Wijewickrema S, Houle ME, Schoenebeck G, Song D, Bailey J (2018) Characterizing adversarial subspaces using local intrinsic dimensionality. arXiv preprint arXiv:1801.02613

Mahendran A, Vedaldi A (2015) Understanding deep image representations by inverting them. In: Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on. IEEE, Boston. pp 5188–5196

Meng D, Chen H (2017) Magnet: a two-pronged defense against adversarial examples. In: Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security. ACM, Dallas. pp 135–147

Nguyen A, Yosinski J, Clune J (2015) Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. IEEE, Boston. pp 427–436

Papernot N, McDaniel PD, Jha S, Fredrikson M, Celik ZB, Swami A (2015) The limitations of deep learning in adversarial settings. CoRR abs/1511.07528:1–16

Papernot N, McDaniel P, Wu X, Jha S, Swami A (2016) Distillation as a defense to adversarial perturbations against deep neural networks. In: Security and Privacy (SP), 2016 IEEE Symposium on. IEEE, San Jose. pp 582–597

Papernot N, McDaniel PD, Goodfellow IJ, Jha S, Celik ZB, Swami A (2016) Practical black-box attacks against deep learning systems using adversarial examples. CoRR abs/1602.02697

Samangouei P, Kabkab M, Chellappa R (2018) Defense-gan: Protecting classifiers against adversarial attacks using generative models. CoRR abs/1805.06605:1–17. http://arxiv.org/abs/1805.06605

Sharif M, Bhagavatula S, Bauer L, Reiter MK (2016) Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition. In: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. ACM, Vienna. pp 1528–1540

Song Y, Kim T, Nowozin S, Ermon S, Kushman N (2017) Pixeldefend: Leveraging generative models to understand and defend against adversarial examples. arXiv preprint arXiv:1710.10766

Su J, Vargas DV, Kouichi S (2017) One pixel attack for fooling deep neural networks. arXiv preprint arXiv:1710.08864

Sun Y, Wang X, Tang X (2014) Deep learning face representation from predicting 10,000 classes. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp 1891–1898

Szegedy C, Zaremba W, Sutskever I, Bruna J, Erhan D, Goodfellow I, Fergus R (2013) Intriguing properties of neural networks. arXiv preprint arXiv:1312.6199

Xie C, Wang J, Zhang Z, Ren Z, Yuille A (2017) Mitigating adversarial effects through randomization. arXiv preprint arXiv:1711.01991
Yuan X, He P, Zhu Q, Bhat RR, Li X (2017) Adversarial examples: Attacks and defenses for deep learning. arXiv preprint arXiv:1712.07107
Zhou Z, Tang D, Wang X, Han W, Liu X, Zhang K (2018) Invisible mask: Practical attacks on face recognition with infrared. arXiv preprint arXiv:1803.04683