Reading Report of Adversarial Examples: Attacks and Defenses for Deep Learning

Huang yucheng[*]

Zhejiang University

April 10, 2022

Abstract

Nowadays deep learning has had a wide application around the world, but there are vulnerabilities found in the major technology of deep learning– deep neural networks(DNN). This report I will represent the defination of adversarial example abd its attack model, meanwhile I will demonstrate the reason why the adversarial examples exist. Also, some typical technique will be introduced with their relative defense technique. In addition, I will explore some possible research direction and application.

I. Defination of the adversarial example

Although deep learning has brought life so many convience, it has raised great concerns in the field of safety and security. Despite great successes in numerous applications, studies find that due to the well-designed input sample, the DL may be vulnerable. So what is the well-designed input sample? Szegedy et al. first generated small pertuebations on the images for the image classification problem and fooled state-of-the-art deep neural network with high probability. These misclassified samples were named adversarial examples. This may affect the real world's secure problem such as confusing autonomous vehicles by manipulating the stop sign. Here a banch of threat model based on different aspects.

i. threat model

Definition 1 *Adversarial Falsification*

this kind of adversarial example aims to mislead the binary classification act wrong, and this kind of adversarial example can be divided into false

positive attack and false negtive attack. Adversarial Falsification can be seen in the image recognization problem.

Definition 2 *Adversary's Knowledge*

adversary knowledge attack the model by use specific raw data as input to modify the model. This kind of threat model can also be divided into two, one is White-box attack and the other is Black-box attack. White-box attack assumes that the attacker know all information about model and is esaier to operate, it can be seen in some origin model trainning while Black-box attack assume that attacker knows nothing about the model and is hard to perform. It can be seen as attacking online ML services linke Google Cloud AI.

Definition 3 *Adversarial Specificity*

This kind of attack often apply in the classification neural network, especially in the multi-

[*] 3190105546

class classification problem. Target attack require the classifier to predict the adversarial example as a certain class while nontarget attack has no such requirement. This aim makes target attack more difficult to apply. These attack mainly appear in real life as a break of facial recognition system, for example I happen to find that the artificial college of our university's facial recognition system can regard any face as one of the student in the college, I personally think that it is because that it didn't add any Negative feedback sample, so that it only gives a positive feedback for any input–this is also a kind of nontarget attack which cause secure problem.

Definition 4 *Attack Frequency*

This threat model aims to divide the adversarial example by how many time it has been optimized in the model. It can be clearly separated into One-time attack and Iterative attack, and it is easy to understand the iterative attack performs better.

ii. how does adversarial example generate

Small *perturbation* is a fundamental premise for adversarial example. Adversarial examples are designed to be close to the original samples and imperceptible to a human, which causes the performance degradation of DL models compared with that of a human. Here I list different concept of perturbation.

Definition 5 *Perturbation Scope*

Perturbation scope determine how many different kinds of perturbation add into the raw input data, which Individual attack generate different perturbations to different input meanwhile Universal attack attack only create a single perturbation to a dataset.

Definition 6 *Perturbation Limitation*

Perturbation Limitation limit how much perturbation add into the origin input. *Optimized perturbation* is stronger than *Constraint perturbation*

Definition 7 *Perturbation Measurement*

Perturbation Measurement measure the magnitude of perturbation by p-norm distance

$$||x||_p = (\sum_{i=1}^{n} ||x_i||^p)^{\frac{1}{p}}$$

Among which l_0 l_2 are commonly used, for l_0 counts the number of pixels changed in the adversarial examples and l_2 measures the Euclidean distance between the adversarial example and the original sample.

II. How to generate adversarial example

i. L-BFGS Attack

this attack is introduced by Szegedy, and it uses the L-BFGS method.

$$\min_{x'} c||\eta|| + J_\theta(x', l')$$
$$s.t.\ x' \in [0,1]$$

this can be seen in the formula that L-BFGS attack search a approximate value by line-search.

ii. Fast Gradient Sign Method

Fast Gradient Sign Method is introduced by Goodfellow et al. and is faster than L-BFGS. It porfamed one-step gradient update along the direction of the sign of gradient at each pixel.

$$\eta = \epsilon sign(\nabla_x, J_\theta(x, l))$$

ϵ is the magnitude of the perturbation, the generated adversarial example x′ is calcalated as $x' = x + \eta$, after this thought has been discovered, many new optimized method is introduced by other, such as Dong et al. generate adversarial example more iteratively

$$g_{t+1} = \mu_t + \frac{\nabla_x J_\theta(x'_t, l)}{||\nabla_x J_\theta(x'_t, l)||}$$

Tramer et al. proposed a new attack by updating the adversarial examples adding random.

$$x_{tmp} = x + \alpha * sign(\mathcal{N}(0^d, I^d))$$

$$x' = x_{tmp} + (\epsilon - \alpha) sign(\nabla_x tmp J(x_{tmp}, l))$$

where α and ϵ are the parameters, $\alpha < \epsilon$

iii. Basic Iterative Method and Iterative Least-Likely Class Method

Many applications today people can only pass the data through devices, so Kurakin et al. applys adversarial examples to the physical world. Then use multiple iteration way by clipping pixel value to avoid a large change on each pixel.

$$Clip_{x,\zeta} = \min 255, x + \zeta, max 0, x - \epsilon, x'$$

where Clip limit the change of the generated in multiple iterations.

$$x_0 = x$$

$$x_{n+1} - Clip_{x,\zeta} x_n + \epsilon sign(\nabla_x J(x_n, y))$$

the author referred to this method as BIM

iv. acobian-Based Saliency Map Attack

This attack is designed by Papernot et al. for its efficiency. They compute the Jacobian matrix of given sample x

$$J_F(x) = \frac{\partial F(x)}{\partial x} = [\frac{\partial F_j(X)}{\partial x_i}]_{i*j}$$

A small perturbation awas designde to successfully induce large output variations so that a change in a small portion of features could foll the neural network.

v. DeepFool

Moosavi-Defooli et al. proposed DeepFool to find the closest distance from the original input to the decision boundary of adversarial examples. To overcome the nonlinearity in high dimension, they performed an iterative attack with a linear approximation.

$$\arg_{\eta_i} \min ||\eta_i||_2$$

$$s.t. f(x_i) + \nabla f(x_i)^T \eta_i = 0$$

this method can also be extended to the multiclass classifier.

vi. CPPN EA Fool & C and W Attack

Nguyen et al. discovered a new type of attack, they use EA algorithm to produce the adversarial examples. For many images from the same evolutionary are found similar in closely related categories. Carlini and Wagner launched a targeted attack to defeat defensive distillation. They first define a new function g, so that

$$\min_{\eta} ||\eta||_p + c * g(x + \eta)$$

$$s.t. x + \eta \in [0, 1]^n$$

where g(x′) $\geq$ if and only if f(x′) = l′. Author lists seven objective function candidate g, here list an effective functions.

$$g(x') = max(\max_{l \neq l'}(X(x')_i) - Z(x')_t, -\kappa)$$

where Z denotes the softmax function and κ is a constant to control the confidence. This author is introduced a new method of finding minimal perturbation.

vii. Zeroth-Order Optimization

Chen et al. proposed a ZOO-based attack. Since this attack does not require gradients, it can be directly deploy in a black-box attack without model transferring. ZOO does not need the access to the victim DL models, it only requires expensive computation to query and estimate the gradients.

viii. Universal Perturbation

Leveraging their previous method on DeepFool, MoosaviDezfooli et al. develop a universal adversarial attack. For each iteration, they use DeepFool method to get a minimal sample perturbation to the total perturbation η, this loop will stop until most data samples are fooled.

ix. One-pixel Atttack

Su et al. generate adversarial examples to avoid the problem of measurement of perceptiveness, the optimization problem becomes

$$\min_{x'} J(f(x'), l')$$

$$s.t.||\eta||_0 \leq \epsilon_0$$

where ϵ_0 = 1 for modifying only one pixel.

x. Feature Adversary

This kind of attack is a kind of targeted attack performed by Sabour et al. The problem can be described by

$$\min_{x'} ||\Phi_k(x) - \phi_k(x')||$$

$$s.t.||x - x'||_\infty < \delta$$

where ϕ_k denotes a mapping from image input to the output of the kth layer.

xi. Hot/Cold

Rozsa et al. proposed a method to find multiple adversarial examples for every single image input by defining a new metric, PASS, to measure the noticeable similarity to humans. PASS os defomed by the combination of the alignment and the similarity measurememt

$$SSIM(x', x) = SSIM(\phi^*(x', x), x)$$

where

$$SSIM(X_{i,j}, x'_{i,j}) = \frac{1}{n * m} \sum_{i,j} RSSIM(x_{i,j}, x'_{i,j})$$

$$RSSIM(x_{i,j}, x'_{i,j}) = L(x_{i,j}, x'_{i,j})^\alpha C(x_{i,j}, x'_{i,j})^\beta S(x_{i,j}, x'_{i,j})^\gamma$$

In each iteration, they moved toward a target (hot) class while moving away from the original(cold) class.

xii. Natural GAN

Zhao et al. utilized generative adversarial networks(GANs) as part of their approach to generate adversarial examples of images and texts, which made adversarial examples more natural to human. The adversarial noise was generated by minimizing the distance of the inner representations.

xiii. Model-Bsed Ensembling Attack

Liu et al. conducted a stduy of transferability over DNN on ImageNet and proposed a *model-based* ensembling attack for rageted adversarial examples. It has to be noticed that the adversarial examples are generated with full knewledge of DNNs, it can be discribe as follow:

$$arg \min_{x'} -log((\sum_{i=1}^{k} \alpha_i J_i(x', l'))) + \lambda||x' - x||$$

where k is the number of DNNs in the generation, f_i is the function of each network, and α_i is the ensemble weight.

xiv. Ground-Truth Attack

This method aims to provide adversarial examples with minimal perturbation. It conducted a binary search and found that example with the smallest perturbation by invoking Reluplex iteratively.

III. Example how the adversarial examples work

Many types of adversarial examples have been introduced, so where can them be apply beside the image classification task.

i. Reinforcement Learning

Huang et al. test attacking deep reinforcement learning networks by FGSM, and has a good performance with l_1 norm on both *white-box attack and black-box* attack. Kos and Song used FGSM to attack A3C algorithm and Atari Pong task using injecting perturbation in a fraction of frames.

ii. Generative Modeling

Kos et al. find that adversaries can leverage AE to reconstruct an adversarial image by adding perturbation to the input of the encoder. The adversaries example change the output by modifying how the model generate.

Tabacof et al. test the attack on MNIST and SVHN data set and find that it is much harder for AE than for classifiers. So they extend the work by designing another two kinds of distances.

$$\min_{\eta} c||\eta|| + J(x', l')$$

In their experiment, "Latent Attack " has the best result.

iii. Face Recognition

This is the most commonly seen application use for adversarial examples. There are examples like sysglass frames which only inject the perturbations into the area of eyeglass frames. Also there are perturbations like adding a penalty of nonprintability score. All these method aim to mislead the classifier and limit the influence made towards human's sight.

iv. Object Detection

The obhect detection task is to find the proposal of an object, which can be viewed as an image classification task for every possible proposal. For image classification, the classifier only needs one target – entire image. The main idea is that iteration only update the loss for the targets correctlu predicted in the prevoius iteration.

v. Semantic Segmentation

Image segmentation task can be viewed as an image classification task for every pixel. Since each pertuabation is responsible for at least one-pixel segmentation, this makes the space of perturbations for segmentation much smaller than that for image classification.

vi. Natural Language Processing

NLP is a hotspot of the AI topic, the adversarial example is added to the NLP by adding or deleting words in the sentences. For example, for a task of reading paragraphs and answering questions about it, to generate adversarial examples that consist correct answer and do not confuse human is in two way, one is adding grammatical sentences similar to the question but not contradictory to the correct answer, the other is adding a sentence with arbitrary English words. There are other search that aims to fool a DL-based sentiment classifier by removing the minimum subset of words in the given text. These changes are easily be recognized by humans, but is confusing for Natural GAN.

vii. Malware Detection

DL has been used in static- and behavioral-based malware detection due to its capability of detecting zero-day malware. Many studies has found that there are adversarial malware samples against many kinds of malware on different operation system such as Android.

IV. The Denfense Against the Adversarial Examples

Since the attack technique has been discovered for years, the countermeasures for the adversarial example have also been developing recently. There are two main strategies: one is reactive for detecting the adversarial examples after DNNs are built, the other is proactive which is making the DNNs more robust before adversarial examples are put in the model.

i. Network Distillation

Network distillation was originally designed to reduce the size of DNNs by transferring knowledge from a large network to a small one. From the DNN we know that the input of next DNN is produced by the previous DNN, so the probability of classes extracts the knowledge learned from the first DNN. Actually network distillation extracted knowledge from DNNs to improve robustness by using high-temperature softmax which reduce the model sensitivity to small perturbations.

ii. Adversarial (Re)training

If adversarial example itself is another kind of input, then there exist a thought that we can put the adversarial examples as training set. This method can increase the robustness of the model for one-step attack but would not help under iterative attacks. Also, research about adding the adversarial examples to the model to avoid overfitting is a kind of usage.

iii. Adversarial Detecting

This kind of attack is what mentioned above reactive strategy. There are many different method to detect input such as Lu et al. mention that trained another DNN to classify the legitimate input from adversarial one, or a detector of small and straightforward neural network predicting on binary classification–SafetyNet extract the binary threshold of each rectified linear unit layer's output as the features of the adversarial dector and detects adversarial images by an radial basis function classifier. Further research build the neural networks with "reverse cross entropy" to better distinguish adversarial examples. Pang et al. use a method called "kernal density in the testing stage", which is more convenient for further detection.

iv. Input Reconstruction

This kind of method treats adversarial example as another perspectivity, for regard them as clean data via transforming them by reconstruction. There are two main method, one is adding Gaussian noise and the other is encoding them with AE as Plan B in MagNet. PixelDefend is a example that can be described as

$$\max_{x'} P_t(x')$$

$$s.t. ||x' - x||_\infty \leq \epsilon_{defend}$$

where P_t denotes the training dustrubution. It can change all the pixel along each channel to maximize the probability distribution.

v. Classifier Robustifying

This is a much easier thought that by increasing the robustness of the neural network. Bradshaw et al. perform Gaussian processes with RBF kernels which provide uncertianty estimation. Also, Abbasi et al. find that adversarial examples usually divide into a small subset of incorrect classes and separated the classes into subclasses, which result in preventing adversarial examples from being misclassified.

vi. Network Verification

Verification can check the properties of a neural network: whether an input violates or satisfies the property. This is a efficient way to prevent adversarial examples from input. For years method has been introduced by either verify by nodes or create a safe resions of DNN.

vii. Ensembling Defense

Due to the multifacet of adversarial examples, multiple defense strategies can be performed together to defend adversarial examples, but ensemble of those defensive approaches does not make the neural networks strong. So we can see that all the defenses are effective only for part fo attacks.

V. Challenges and Future Outlook

It can be seen from the previous section that the defense method is not so complete, so where the challenges are, where the future of the defense techniques are.

i. Transferability

There is an important problem for both adversaries and resarchers, why do adversarial examples transfer, and how to stop the transferability? Transferability is a common property for adversarial examples. Adversarial examples generated aganinst a neural network can fool other neural networks with different architectures, even other classifier trained by different ML algorithms. It seems serious for many training model is regard as block-box even for nowadays, and it is hard to find why this phenomenon even exits. From this perspectivity, it can also tell that why the emsembling defense is not so well-behaviour.

ii. Existence of Adversarial Examples

The reason for the existence of adversarial examples is still an open question, here list some possible explanation:

Data Incompletion: One assumption is that adversarial examples are of low probability and low test coverage of corner cases in the testing data set.

Model Capability: Adversarial examples are a phenomenon not only for DNNs but also for all classifiers, it may be too linear in high-dimensional manifolds and exits when the decision boundary is close to the manifold of the training data. others may believe taht adversarial examples are due to the "low flexibility" of the classifier for certian tasks.

No Robust Model: Dong et al. suggested that the decision boundaries of DNNs are inherently oncorrect, which do not detect semantic objects. Similarly, other also find th there is no robust classifier to adversarial examples. Current studies on adversarial examples mainly focus on the image classification task. No existing paper explains the relationship among different applications and existence of a universal attacking/defending method to be applied to all the applications.

iii. Robustness Evaluation

Some defensive method that aim to solve one kind of attack may later be vulnerbale to new type of attack, this is a commmon seen in the comprtition between attack and defense for adversarial examples. Hence, the evaluation on the rubustness of a DNN is necessary.

Methodology for Evaluation on the Robustness of Deep Neural Networks:Many DNNs are planned to be deployed in safety-critical settings. But defending known attack is not so efficient, so a methodology for evaluating the robustness of DNN is required.

Benchmark Platform for Attakcs and Defenses: Most attacks and defenses won't publicly show thier method and code, or parameters they use, so it is difficult for other to reproduce their result. If a open-source platefrom is performed to worldwide, this may construct a benchmark for all the existing attack and can easier to build a defensive way.

Various Applicatios for Robustness Evaluation: A wide range of applications make it hard to evaluate the rubustness of a DNN architecture. So how to compare methods generating adversarial example under different threat models, do we have a universal methodology to evaluate the robustness under all scenarios?