Adversarial Attacks

Sergey Kim
Higher School of Economics
Applied Mathematics and Informatics
Moscow, Russia
kims230599@gmail.com

***Abstract*—Deep Learning is very fast-growing area of artificial intelligence. In the field of Computer Vision, it has become the workhorse for applications ranging from self-driving cars to surveillance and security. But with great force comes great responsibility. And algorithms in this sphere should overcome many different obstacles. One of them is called Adversarial attacks. They pose a serious threat to the success Deep Learning in practice. We review specific features of Avdersarial attacks and defences against them.**

***Keywords*—Deep Learning, adversarial perturbation, black-box attack, white-box attack, adversarial learning**

Introduction

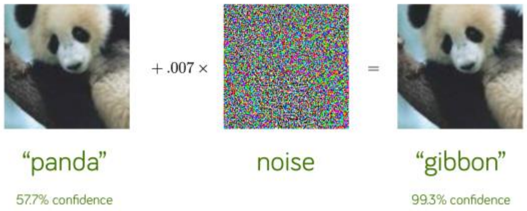
Deep Learning have become the preffered choice to solve many challenging tasks: analysis of mutations in DNA, natural language understanding and speech recognition, playing difficult games, objects detection and classification, etc. We will focus on the last moment.

Mistakes and ways to avoid them are very important in object detection, because these algorithms are integrated into the autopilot, Face ID security, medical diasgnoses and others. And one of the key problem which has no solution is an Adversarial attacks problem.

Main part

To begin with, we will define terms which will be used in this article.

- *Adversarial example/image* is a modified version of an original image.
- *Adversarial perturbation* is the noise that is added to the original image to make it an adversarial image.
- *Adversarial training* uses adversarial images besides the clean images to train machine learning models.
- *Adversary* more commonly refers to the agent who creates an adversarial example. However, in some cases the example itself is also called adversary.
- *Black-box attacks* feed a targeted model with the adversarial examples (during testing) that are generated without the knowledge of that model. In some instances, it is assumed that the adversary has a limited knowledge of the model (e.g. its training procedure and/or its architecture) but definitely does not know about the model parameters. In other instances, using any information about the target model is referred to as 'semi-black-box' attack. We use the former convention in this article.
- *Detector* is a mechanism to detect if an image is an adversarial example.
- *Fooling ratio/rate* indicates the percentage of images on which a trained model changes its prediction label after the images are perturbed.
- *One-shot/one-step methods* generate an adversarial perturbation by performing a single step computation
- *Rectifier* modifies an adversarial example to make the prediction as it was before damage to an original image.
- *Targeted attacks* fool a model into falsely predicting a specific label for the adversarial image. They are opposite to the non-targeted attacks in which the predicted label of the adversarial image is irrelevant, as long as it is not the correct label.
- *Threat model* refers to the types of potential attacks considered by an approach, e.g. black-box attack.
- *Transferability* refers to the ability of an adversarial example to remain effective even for the models other than the one used to generate it.
- *Universal perturbation* is able to fool a given model on 'any' image with high probability. Note that, universality refers to the property of a perturbation being 'image-agnostic' as opposed to having good transferability.
- *White-box attacks* assume the complete knowledge of the targeted model, including its parameter values, architecture, training method, and in some cases its training data as well.



Attacks

Attacks for classification include:

1. Box-constrained L-BFGS. L-BFGS Attack calculated approximate values of adversarial examples by line-searching $c > 0$.

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

2. Fast Gradient Sign Method (FGSM).
3. Basic & Least-Likely-Class Iterative Methods.
4. Jacobian-based Saliency Map Attack (JSMA).
5. One-pixel Attack. It is an extreme case for adversarial attack: we should break the image with one pixel changing only.
6. Carlini and Wagner Attacks (C&W).
7. DeepFool. At each iteration, the algorithm perturbs the image by a small vector. DeepFool algorithm can compute perturbations, and this perturbation will be smaller than a result of FGSM algorithm (in terms of their norm).
8. Universal Adversarial Perturbations. If the methods like FGSM, ILCM, DeepFool etc. compute perturbations to cheat a model on a single image, this approach has to be able to fool a network with high probability on any image.
9. Universal Perturbations for Steering to Extract Targets (UPSET) and Antagonistic Network for Generating Rogue Images (ANGRI).
10. Houdini. It os an approach for fooling gradient-based learning machines by generating adversarial examples that can be tailored to task losses.
11. Adversarial Transformation Networks (ATNs). The adversarial examples generated by these networks are computed by minimizing a joint loss function comprising of two parts. The first part restricts the adversarial example to have perceptual similarity with the original image, whereas the second part aims at altering the prediction of the targeted network on the resulting image.
12. Miscellanious Attacks.

The main idea of L-BFGS and FGSM are one-shot methods, and all others are iterative.

Only One-pixel Attack, UPSET, ANGRI and Houdini methods are Black-box attacks, and all other methods belong to White-box attacks family.

In short, almost everything – transformation of l_0-, l_1-, l_2- or l_∞-norm with FSGM or L-BFGS which have one-step determined algorithm or basic Itearive Methods, and some similar descriptions are omitted.

Defense

Countermeasures for adversarial attacks have two types of defense strategies: proactive (to make deep neural networks more robust before generating adversarial examples) and and reactive (to detect adversarial images).

The first idea is try to modify input during learning and training. But it has some problems and it cannot be used everywhere. Brute-force approach is resource intensive, Data compression can oppose nothing to the strongest attacks, Foveation is yet to demonstrate its effectiveness against more powerful attacks than it was tested. Other types to change input can't avoid all the problems which have these approaches.

We can modify the network by adding gradient regularization, masking, or by adding more specific layers and subnetworks, but there is still no common solution. However, we can defend against certain type of attacks effectively enough.

Proactive strategy as a general rule contains the second auxiliary network, which rejects all adversarial examples.

Using external models as network add-on when classifying unseen examples is nice solution, but it is not enough sometimes.

Combining all these methods may give us som kind of defense, but it is still no one hundred percent guarantee that it is rectifier. Sometimes it will falter.

Conclusion

Adversarial attacks should be explored to get an opportunity to fight against them. It is very important in our everyday life. Drivers permanently calculate the distance to the signs, and we have no margin for error. We can't let everyone unlock smartphones, he don't own. And Deep Learning algorithms are not only about cyberspace, it is also about our physical world.

Studying this area is very important for the models, and every model should be checked with adversarial attacks methods. And this article demonstrates us what is happening. Research defenses will help developers to overcome adversarial attacks methods in the future.

References

[1] N. Akhtar and A. Mian. Threat of adversarial attacks on deep learning in computer vision: A survey. arXiv preprint arXiv:1801.00553, 2018.
[2] X. Yuan, P. He, Q. Zhu, R. R. Bhat Adversarial Examples: Attacks and Defenses for Deep Learning. arXiv preprint arXiv:1712.07107, 2017.

Word count 1196