Deep Learning Adversarial Examples

Christin Wilson
Department of Computer Science, Clemson University
cwils28@clemson.edu

***Abstract*—With rapid progress and advancements, deep learning has been very effectively used for visual recognition. Many safety critical environments make use of deep learning to handle sensitive data. But recently, the deep neural networks (DNN) have been vulnerable to an adversarial attack. Adversarial examples are well-designed inputs that have been intentionally designed so as to look like a normal input to humans but fools the DNN to make an incorrect classification. These adversaries attack only at the deploying / testing phase. This poses a potential security threat. This purpose of this paper is to survey the different adversarial examples, the extent to which they pose a security threat and the countermeasures that can be taken against the attack.**

***Index Terms*—deep learning, Deep neural network, adversarial examples**

I. Introduction

Deep learning (DL) has made significant progress in a wide domain of machine learning (ML). Deep Learning is the type of machine learning in which the computers analyse and extract useful patterns from the raw data and gain knowledge and experience without having to explicitely program it. It has achieved impressive success on a wide range of domains like computer vision [1] and natural language processing [2], outperforming other machine learning approaches.

A significant progress has been made in classification problems by using deep neural networks. Supervised learning is carried out in Deep convolutional neural network (DCNN). A large training set which contains various inputs and their known outputs are fed to the DCNN. This process is called training. After this, when the trained DCNN is presented with unknown inputs, the DCNN can predict the output with accuracy.

Due to the availability of large amount of data and more capable hardware, it has become very simple for a classifier to extract very abstract level data from raw inputs using deep learning. Deep learning also requires less knowledge and human engineering for training unlike other machine learning methods[3]. The ability to recognize visual objects have been greatly improved. Some of the high-end neural networks perform even better than humans on certain difficult, large-scale image classification tasks.

The security and privacy concerns have been raised as most of the real world applications that are powered by deep learning like face recognition, self-driving cars, malware detection are accompanied with sensitive data which require safety-critical environments. For example, Automobile companies are testing out self-driving cars. These use plenty of deep learning techniques, object recognition and reinforcement learning some of those. Companies like Apple and Samsung have introduced Facial recognition system for bio-metric authentication. Although these have been successful, many of these applications are life-crucial and thus require a safety critical environment for their security.

Recent studies have shown that DL models are vulnerable to attacks during the prediction phase from an adversarial input. Adversarial examples are well-designed inputs that have been intentionally designed to look like a normal input to humans but fools the DNN to make an incorrect classification. They are inputs with tiny perturbations added to them that are imperceptible to humans but is capable of easily fooling a deep learning model to misclassify the input. These are like optical illusions to machines. Although the DCNNs have a high accuracy rate, these adversarial examples trick the DCNN to give an incorrect classification.

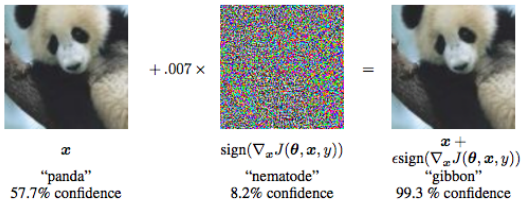

Fig. 1. From Explaining and Harnessing Adversarial Examples by Goodfellow et al[5].

Adversarial examples can be used to confuse various biometric authentication techniques like facial or voice recognition systems to breach into systems and cause harm. Self driving cars could be at a risk of accidents if the different road signs, crafted by adversarial perturbations, appear as different signs and thus lead to an incorrect classification. For example, if the vision camera in a self-driving vehicle recognizes a stop sign as a speed limit sign and doesnt respond with the right action, the consequences can be deadly. If perturbations are added to digits in a cheque, it could lead to the wrong classification of the number and thus an incorrect amount will be transferred between accounts.

In this paper, we summarize the different adversarial examples and their countermeasures. Also we look at the extent to which they pose a security threat and how to avoid them using various techniques.

II. Main techniques

A. Taxonomy of adversarial examples

The different approaches for generating adversarial examples can be categorised into different dimensions: threat model and perturbation.

1) Threat model: Based on different conditions and requirements, the attributes needed in the adversarial examples are detected and specific attack approaches are deployed. Based on this we can further divide the threat model into four aspects: adversarial falsification, adversarys knowledge, adversarial specificity and attack frequency. [3]

Adversarial falsification:

- False positive attacks: This is also called as a Type I error. These are an adversarial image unrecognisable to a human, while the DNN will classify it as an output with a high confidence score. These attacks generate a negative sample which misleads the DNN and is classified as a positive one.
- False negative attacks: This is also called as a Type II error. These are an adversarial image that is recognisable by a human but the DNN cannot identify it. These attacks generate a positive sample which the DNN classifies as a negative one.

Adversary's knowledge:

- Insider Threat: These refer to white box attacks. These adversaries may know everything related to the DNN such as the architecture, the training data, the intermediate results during the computation etc. These help the adversaries to manipulate the training process of the DNN and thus manipulate the outcomes. These are the more common adversarial example attacks.
- Outsider Threat: These refer to black box attacks. These adversaries have no access to any information related to the DNN training or the trained model. They only have access to the predictions of the DNN. These attackers may know the general and common background knowledge of the DNN that is usually available. The white box attacks can be transferred to attack black box services.

Adversarial specificity:

- Targeted attacks: It is a source-target misclassification. The objective of the adversary is to misclassify a benign input by misguiding the DNN to a specific class by perfectly crafting the adversarial example. this results in the predicted class of an input to be changed from a source class to a specific target class that the attacker wants. An example of these kind of attacks is the attack on a biometric authentication system. The adversary must try to disguise a face as an authorized user. Another example is when an adversary wants an image that is a 6 to be classified as a 4.
- Non-targeted attacks: These attacks do not have a specific target class. The objective of the adversary is to misclassify a benign input by misguiding the DNN to any other class. The output of the DNN is not targeted to a certain class by the attacker. It is a source class misclassification attack. These non-targeted attacks are easier to implement compared to targeted attack since it is more flexible and has more options of where to direct the output. these are generated in two ways: 1) choosing the one with the least amount of perturbations. 2) minimizing the probability of the correct class. An example of this kind of attacks is a facial detection system in which the attacker wants to add a few perturbations in order to make his face undetectable. Another example is when an adversary wants an image that is 6 to be classified as any number other than 6.

Attack frequency:

- One-time attacks optimize the adversarial examples in one time. These are preferred if the adversarial example is required to be generated in real time.
- Iterative attacks optimize the adversarial examples in multiple times. These have better performance but requires more computational time to generate.

2) Perturbation: The objective of an adversarial example is to be designed such that it is very close to the input sample and is imperceptible to the human while at the same time the classifier should be misguided to classify it in a wrong class with high confidence. Thus the perturbations should be as minimum as possible while degrading the performance of the deep learning model.

Individual attacks are those that generate different perturbations for different inputs. These are more commonly used in the current adversarial attacks.

Universal attacks are those in which the perturbation is universal and the same ones are applied tp the entire dataset. They are applied to all the inputs. These are more simple to deploy in the real world since new perturbations wont have to be created for each input.

Based on the perturbation limitation there are two types: optimized perturbation where perturbation is set as the goal of the optimization problem and the constraint perturbation where the perturbation is set as the constraint of the optimization problem.

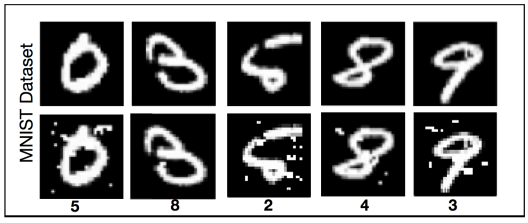

Fig. 2. AUTHENTIC VERSUS ADVERSARIAL EXAMPLES. The images on the top row of the figure are legitimate images. The images on the bottom row of the figure are adversarial examples, and the numbers below each of those images are the numbers that the DCNN mistakenly classifies the adversarial examples. Adapted from [4].

B. DNNs and the MNIST dataset

DNNs are highly efficient in learning highly accurate models in many domains, especially to classify visual images.

MNIST dataset is a popular handwritten digits dataset. It is used for visualizing evasion attacks. This dataset contains 70,000 images of handwritten digits from 0 to 9. Each sample is a 28*28 pixel, 8-bit grayscale image. Perturbations are added to the images such that humans cannot find the difference but the DNN classifies it as an entirely different number with high confidence.

III. Issues and problems

A. Attacking methods

To test the robustness of the systems, we use the following standard and strong attacks. The hyper-parameters and the conditions will be changed as required.

A. Fast gradient sign method (FGSM): A linear search to find the adversarial examples was first proposed but it was not found affordable due to the large amount of computation required. Goodfellow et al [5], described this simple and fast method for generating adversarial examples called fast gradient sign method. . The perturbation is calculated through back propagation after the process is complete. The gradient of the loss function is used by the FGSM to find the direction in which the input data has to be changed in order to minimise the loss function. Even though this method is fast, the adversarial examples found through this method are not the most optimal.

B. Iterative Methods: Kurakin et al [6], extended the FGSM by introducing an iterated version. The fast gradient sign method is made to run for multiple iterations to get a finer optimization. This method produced better results than FSGM. another method called the iterative least likely class method was proposed to attack a specific class with enhanced capability.

C. Deepfool Method : Moosavi-Dezfooli et al [7], introduced the DeepFool adversary, which can choose the class to which an example is switched to. Using minimal perturbations generated through iterative linerzation of the classifier, DeepFool adversary switches the class to a specic yet incorrect label thereby leading to a less accurate or even an incorrect model[7].

IV. Issues and Problems

A. Countermeasures for adversarial examples

There are two type of defence strategies: 1) reactive: detect adversarial examples after deep neural networks are built 2)proactive: train the DNN to be more robust.

The methods to prevent adversarial examples are discussed below.

1) Network Distillation: Papernot et al, used network distillation to defend DNNs. In this method, the probability of the classes produced by a DNN is used as input to train the second DNN. thus the knowledge is transferred from a large network to a small one [8],[9]. For this reason, Network distillation was originally designed to reduce the size of DNNs. By extracting the knowledge from a DNN, network distillation improved the robustness. Network Distillation defence was tested on the MNIST datasets and it was found out that it reduced the success rate of the adversarial examples. The generalization of the Neural network was also improved by the "Network Distillation.

2) Adversarial (Re)training: This method includes training the deep neural network with the adversarial examples. This made the DMM more robust. Goodfellow et al [10] and Huang et al [11] included adversarial examples in the training stage for the MNIST dataset. Adversarial examples were generated in every step of training and these were later on fed to the training set. These studies showed that adversarial (re)training improved the robustness of the DNN. Regularization and precision were improved by adversarial training. Further studies suggested that adversarial training was useful only for one step attacks and it didnt increase the robustness of neural networks when under iterative attacks. [12] also showed that adversarial trained models on MNIST dataset were more robust to white box attacks than to the black box attacks. [12] proposed ensembling adversarial training method to deal with black-box models. In this method, the adversarial examples generated from multiple sources were used to train the model.

3) Input Reconstruction: Adversarial examples can be transferred to clean data via reconstruction. The prediction of deep learning models won't be affected by adversarial examples after this transformation. Gu and Rigazio [13] proposed a denoising auto-encoder network which is trained to encode adversarial examples to original ones to remove the perturbations. [14] reconstructed the adversarial examples by 1)adding Gaussian noise or 2) encoding them with autoencoder.

4) Network Verification: This method makes use of verifying the properties of a deep neural network to defend adversarial examples by detecting the new unseen attacks. This method checks if an input violates or satisfies a property, thus checking the properties of the neural network. Katz et al proposed a verification method called reluplex [15]. The authors showed that for a small perturbation, there was no adversarial example to misclassify the neural networks. Due to a large amount of computation required, reluplex was found to be considerably slow. To make it faster the nodes with higher priority were checked first.

5) Ensembling Defenses: Since adversarial examples are multi-faceted, a combination of multiple defense mechanisms can be made use of to defend the attack. They can be performed either sequentially or parallelly. An example of this method is the aforementioned PixelDefend [16] which comprises of an adversarial detector and an input reconstructor.

6) Fine-tuning: In this method, a combination of both clean images and transformed images of the MNIST training set were used to train the classifier. Now even if the inputs were not adversarial examples, the transformed images led to an increase in the robustness of the deep learning model.

The defence listed are shown to be effective only for certain part of attacks. They fail to defend against some strong and unseen attacks. Most defences target the adversarial examples in the computer vision area. new defences are being developed especially for the ones which involve safety-critical environments.

V. Future Trends

Currently, most of the studies are concentrated on finding defences for the adversarial examples in the computer vision area. Studies should be carried out for other problems as well, especially those that require a safety critical environment. Research must be carried out to develop models which have lower confidence while making mistakes. Also, research must be carried out to make models which are harder to be reverse engineered and that is less prone to transfer from related models since these both result in the attack from the adversarial examples.

References

[1] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classifi- cation with deep convolutional neural networks. In Advances in neural informa- tion processing systems. NIPS Foundation, 1097-1105.

[2] Ronan Collobert and Jason Weston. 2008. A unified architecture for natural lan- guage processing: Deep neural networks with multitask learning. In Proceedings of the 25th international conference on Machine learning. ACM, 160-167.

[3] Yuan, X., He, P., Zhu, Q., Bhat, R.R. and Li, X., 2017. Adversarial examples: Attacks and defenses for deep learning.arXiv preprint arXiv:1712.07107.

[4] N. Papernot, P. McDaniel, X. Wu, S. Jha, and A. Swami, Distillation as a defense to adversarial perturbations against deep neural networks, in 37th IEEE Symposium on Security and Privacy, 2016.

[5] I. J. Goodfellow, J. Shlens, and C. Szegedy, Explaining and harnessing adversarial examples, in International Conference on Learning Representation (ICLR), 2015.

[6] A. Kurakin, I. Goodfellow, and S. Bengio, Adversarial examples in the physical world, arXiv preprint arXiv:1607.02533, 2016.

[7] Moosavi-Dezfooli, Seyed-Mohsen, Alhussein Fawzi, and Pascal Frossard. "Deepfool: a simple and accurate method to fool deep neural networks. "Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016.

[8] J. Ba and R. Caruana, Do deep nets really need to be deep? in Advances in neural information processing systems, 2014, pp. 2654 2662.

[9] G. Hinton, O. Vinyals, and J. Dean, Distilling the knowledge in a neural network, arXiv preprint arXiv:1503.02531, 2015.

[10] . I.J.Goodfellow,J.Shlens,andC.Szegedy,Explainingandharnessing adversarial examples, arXiv preprint arXiv:1412.6572, 2014.

[11] R. Huang, B. Xu, D. Schuurmans, and C. Szepesvri, Learning with a strong adversary, arXiv preprint arXiv:1511.03034, 2015.

[12] F. Tramr, A. Kurakin, N. Papernot, D. Boneh, and P. McDaniel, Ensemble adversarial training: Attacks and defenses, arXiv preprint arXiv:1705.07204, 2017.

[13] S. Gu and L. Rigazio, Towards deep neural network architectures robust to adversarial examples, Proceedings of the International Conference on Learning Representations (ICLR), 2015.

[14] D. Meng and H. Chen, Magnet: a two-pronged defense against adversarial examples, CCS, 2017.

[15] G.Katz,C.Barrett,D.Dill,K.Julian,and M.Kochenderfer,Reluplex: An efficient smt solver for verifying deep neural networks, arXiv preprint arXiv:1702.01135, 2017.