МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
имени М.В. ЛОМОНОСОВА
Факультет Вычслительной Математики и Кибернетика
Кафедра Информационной Безопасности

Ли Хуаюй

«Собрание аннотаций по научным статьям»

Научный руководитель: канд. физ.-мат. наук
с.н.с лаборатории ОИТ Намиот Д.Е

Москва, 2021

Contents

1 Intriguing properties of neural networks

1.1 Link

This article [1] was found and cited at the link: `http://arxiv.org/abs/1312.6199`.

1.2 Introduction

This paper introduces the concept of adversarial samples for the first time and presents a gradient method for generating adversarial samples based on the Box-constrained L-BFGS White-Box attack. The so-called adversarial sample is the introduction of a small perturbation on the original sample, which can make the model misclassify the situation, which is also the idea inspired by the adversarial generative network. Also, this paper argues that the semantic information in deep neural networks is based on the whole network and not on the neurons of a particular layer.

Note 1.1. *In the 3-rd chapter, the property is proved (experiments are performed on the MINST dataset): in the higher levels of the neural network, it is the whole space that contains semantic information, not a particular unit.*

In the 4-th section, the property is described: blind spots in neural networks - finding Adversarial Examples. and the idea of finding adversarial samples is provided: formulaic description, experimental results, conclusions.

Note 1.2. *(Why do adversarial samples arise in neural networks?) The authors give the following hypothesis(In the 4-th section):*

1. *It is assumed that is possible for the output unit to assign nonsignificant (and, presumably, non-epsilon) probabilities to regions of the input space that contain no training examples in their vicinity.*

2. *The adversarial examples represent low-probability (high-dimensional) "pockets" in the manifold, which are hard to efficiently find by simply randomly sampling the input around a given example.*

1.3 Key idea

According to the authors, finding the adversarial samples is a stepwise optimization process divided into two aspects:

1. We need to make sure that the added perturbations are as small as possible so that they are imperceptible to the naked eye.

2. On the other hand, we need to make sure that the model misclassifies the adversarial samples.

Based on these two points, the authors give the following objective function:

$$\begin{aligned}&\text{Minimize } ||r||_2 \text{ subject to :}\\&1.\ f(x+r) = l\\&2.\ x + r \in [0,1]^m\end{aligned}$$

Where,

1. x is the original image.
2. r is the added perturbation
3. f is the classifier.
4. l is the target category (which is different from the correct category for x).

We want r to be as small as possible, while making the adversarial sample $x+r$ be misclassified under a specified category l. We also need the generated $x + r$ to have a value between $[0, 1]$ (to ensure it is a legitimate image). But finding a suitable solution to this problem is not easy, and in the paper the authors give a modified solution: finding the optimal perturbation r from the perspective of the loss function:

$$\text{Minimized } c|r| + \text{loss}_f(x + r, l) \text{ which subjects to } x + r \in [0, 1]^m$$

This problem can be solved by the Box-Constrained L-BFGS algorithm.

1.4 Datasets

The Datasets used in this article are as follows:

1. MNIST dataset
   —A simple fully connected network consisting of one or more hidden layers and softmax classifiers — called "FC".
   —A classifier trained on an autoencoder — called "AE".
2. ImageNet dataset
   —AlexNet Network.
3. 10M image samples from Youtube
   —Unsupervised training network with 1 billion learnable parameters — called "QuocNet".

1.5 Result

The experimental results are detailed in the original article.(In the subsection 4.2)

1.6 Conclusion

Based on the experimental results in the paper, the following conclusions can be roughly drawn:

1. For all the Network Frameworks mentioned in the paper (AlexNet, QuocN...), adversarial samples can be generated by using methods(In the subsection 4.1).

2. Adversarial samples have the ability to generalize across models: a large portion of the adversarial samples generated on model A are also valid on model B (which has the same structure as model A with different hyperparameters).

3. D_1, D_2 are different subsets of dataset D, each trained with a different model, and the adversarial samples generated on D_1 are also valid on D_1.

2 Explaining and harnessing adversarial examples

2.1 Link

This article [2] was found and cited at the link: `http://arxiv.org/abs/1412.6572`.

2.2 Introduction

Currently, machine learning has been widely used in various fields of daily life, but Szegedy et al. first discovered in 2014 that current machine learning models, including neural networks, are vulnerable to adversarial examples. The so-called adversarial examples, i.e., attackers generate adversarial samples by slightly perturbing normal samples, and achieve the purpose of misleading the classifier while ensuring that the attack does not affect the recognition of human eyes.

This article mainly proposes a linear hypothesis that is different from the previous papers to explain the existence of adversarial examples. At the same time, the paper proposes a simple method for generating adversarial examples called FGSM, and then uses the adversarial examples generated by

the attack method for adversarial training. In general, this article mainly describes three aspects of adversarial samples:

1. Existence
2. Attack methods
3. Defense methods

2.3 Key idea

1. The reason why neural networks are so vulnerable to adversarial sample attacks is because of the linear nature of the network.
2. The article also presents the earliest FGSM adversarial sample generation method.
3. The authors also propose an alternative perturbation: a very small rotation of the image that increases the value of the loss function.
4. By adding a certain number of adversarial samples (randomly generated) to the training samples, a certain regularization effect on the model can be achieved.
5. Adversarial training also improves the robustness of the model to adversarially perturbed samples, and the authors also make the point that perturbing the input samples in adversarial training is more effective than perturbing the hidden layer features, and the experiments show a weak regularization effect of the training model obtained from hidden layer perturbation. (Some ideas in the section 3,4,6)

2.4 Datasets

The Datasets used in this article are as follows:

1. MNIST dataset

2.5 Result

For the adversarial sample experiments: The authors experimented with Softmax and Maxout neural networks on the MNIST dataset and the parameters $\epsilon = 0.25$ and $\epsilon = 0.1$, respectively, and the error rates for the adversarial samples were as follows:

- $\epsilon = 0.25 : 99.9\%$ and 89.4%.

- $\epsilon = 0.25$: 99.9% and 89.4%.
- $\epsilon = 0.1$: Maxout 87.15%.(In the section 4)

and the confidence level of the model for both error samples is extremely high.
For the adversarial training experiments:

- The error rate of Maxout network for the adversarial sample before adversarial training was 89.4%.
- After adversarial training, the error rate decreases to 17.9%.
- Also, the error rate of the adversarial samples generated by the model after adversarial training is effectively reduced in the original model.(In the section 5)

2.6 Conclusion

1. The adversarial samples can be interpreted as a high-dimensional product, and they are more due to the high-dimensional linear features of DNNs.
2. The generalizability of the adversarial samples across multiple models of the same task can be interpreted as the high similarity of different models in terms of weights.
3. The perturbation direction is more important compared to the particular point in the sample space.
4. Confrontation training helps regularization.
5. The authors propose that the fully nonlinear model RBF network can resist the adversarial samples to some extent (making their misclassification confidence relatively low). (More informations are in the section 10)

3 Adversarial examples in the physical world

3.1 Link

This article [3] was found and cited at the link: `http://arxiv.org/abs/1607.02533`.

3.2 Introduction

This article was published by Goodfellow and others. It is a classic paper in the field of adversarial examples. This paper differs from others in that it focuses on the input of adversarial samples to the convolutional neural network Inceptionv-Net3 through sensors such as cameras, which is equivalent to an actual attack in the physical world. The paper also proposes BIM(BASIC ITERATIVE METHOD):

$$X_0^{adv} = X, X_{N+1}^{adv} = \text{Clip}_{X,\epsilon}\left\{X_N^{adv} + \alpha \text{sign}(\nabla_X J(X_N^{adv}, y_{true}))\right\},$$

and ILCM(ITERATIVE LEAST- LIKELY CLASS METHOD):

$$X_0^{adv} = X, X_{N+1}^{adv} = \text{Clip}_{X,\epsilon}\left\{X_N^{adv} - \alpha \text{sign}(\nabla_X J(X_N^{adv}, y_{LL}))\right\},$$

adversarial sample generation methods and compares the results with the previously proposed FGSM algorithm [2] on the ImageNet dataset. This paper proposes new attack methods that introduce metrics for the effectiveness of adversarial attacks, with data collected through actual physical cameras. Then a white-box attack was performed in the real world, and finally a black-box attack was performed.

3.3 Key idea

1. BIM method
2. ILCM method
3. Attacks in the physical world are achieved by collecting printed clean samples and counter samples through the phone camera. When the denominator is 1, clean samples are correctly classified and adversarial samples are incorrectly classified, and when the numerator is 1, clean samples are correctly classified and adversarial samples are incorrectly classified, and after transformation (taken after printing) the adversarial samples are correctly classified. (detailed in the subsection 3.1)

3.4 Datasets

The Datasets used in this article are as follows:

1. ImageNet dataset

3.5 Result

1. The adversarial images generated by the "Fast" method are more robust than those generated by the iterative method because the iterative method generates more minute perturbations that are ignored by the photographic transformation.

2. In some cases, the adversarial corruption rate is higher in the "Preiltered case" than in the "Average case".(In the subsection 3.2 and 3.3)

3. Since black-box scenarios are a more realistic model for many security threats. The authors demonstrate that physical adversarial samples deceive a model of a different architecture through a black box attack. The experiments are demonstrated through an open source image classification mobile app.(In the subsection 3.4)

4. The "fast method" is the most robust and the "iterativa least-likely" is the least robust under the photo transformation.

5. Contrast and luminance changes do not have a significant effect on the counter samples.

6. Blur, noise and JPEG encoding have higher corruption rates than contrast and luminance changes.(In the subsection 3.5)

3.6 Conclusion

The authors' team used images taken from a cell phone camera as input to the Inception v3 image classification neural network. The authors show that in such a setup, most of the adversarial images produced using the original network are misclassified even when fed into the classifier via the camera, a finding that demonstrates the possibility of adversarial samples for machine learning systems in the physical world.

4 On Detecting Adversarial Perturbations

4.1 Link

This article [4] was found and cited at the link: `https://arxiv.org/abs/1702.04267`.

4.2 Introduction

The detection network approach will directly predict whether a given sample is an adversarial sample by means of a neural network, i.e., the identification of an adversarial sample is directly transformed into a binary classification problem trained in an end-to-end manner. The detector network approach extends the original neural network (classifier) with a detector sub-network whose task is to discriminate whether the sample is from real data or not. The output of the detector is a scalar of range that represents the probability that the data belongs to the adversarial sample. The design of the detector is related to the specific dataset and the architecture used is generally a convolutional neural network.

4.3 Key idea

To train this detector, the classifier is first trained on the original training Dataset, and when the classifier is trained, a corresponding adversarial sample is generated for each sample in the training Dataset using methods such as FGSM or DeepFool.

This results in one original Dataset and one adversarial sample Dataset of the same size, and then the detector is trained. The training data consists of half of the real sample data and half of the generated adversarial samples, with the labels of the samples replaced with the corresponding sources, i.e., the real samples or the adversarial samples. For detector training, the weights of the original classification network are fixed and the detector is trained using cross-entropy as the loss function.

4.4 Datasets

The Datasets used in this article are as follows:

1. ImageNet dataset
2. CIFAR10 dataset

4.5 Result

Experiments show that when the FGSM method is used to generate adversarial samples, setting , the detection network can detect 90% of the adversarial samples, and setting , the detection network can detect 97% of the adversarial samples; when the DeepFool method is used to generate adversarial samples, the detection network can detect 82% of the adversarial samples. When the attacker knows the gradients of both the classification network and

the detection network, and the FGSM method is used for dynamic adversarial training, setting , the detection network can detect 89% of the adversarial samples.(In the section 4)

4.6 Conclusion

1. Using the detector sub-network attached to the main classification network, adversarial examples can be detected.

2. Although this does not directly allow the correct classification of adversarial examples, it can mitigate adversarial attacks on machine learning systems by resorting to backup solutions.

3. The gradient propagated back through the detector may be used as a source of regularization of the classifier against adversarial examples.

4. Developing methods for training detectors explicitly such that they can detect many different kinds of attacks reliably at the same time would be essential for safety- and security-related applications.

5 Synthesizing Robust Adversarial Examples

5.1 Link

This article [5] was found and cited at the link: `https://arxiv.org/abs/1707.07397`.

5.2 Introduction

After analyzing the previous related work on adversarial sample generation, the authors Athalye et al. made a more in-depth study in the direction of physical environment transformation. Questions are raised on the generation and effectiveness of adversarial samples in 2D, 3D and real-world environments, respectively. This work proposes the framework of Expectation OverTransformation (EOT) algorithm. Simulating specific environments (including 2D and 3D), the authors also propose an adversarial sample generation algorithm based on this framework, which generates samples with good robustness in various perspectives and given distributions of environmental conditions. Real-world adversarial objects are also generated based on this framework with the support of 3D printing technology to demonstrate the existence of 3D adversarial objects.

5.3 Key idea

The adversarial samples obtained by the above method are not immune to various visual transformations. Therefore, based on the above method, EOT introduces the transformation distribution T. For any transformation function t, the input of the classifier is changed from the original adversarial sample x' to $t(x')$. In practice, various transformations such as rotation, translation, noise addition, etc. can be represented. Once the EOT is parameterized, the distribution T is determined and the EOT framework can optimize the samples under the distribution T to obtain the adversarial samples.(In the subsection 2.1)

5.4 Datasets

The Datasets used in this article are as follows:

1. ImageNet dataset

5.5 Result

In the experimental session, the authors used different transform distributions for different experimental environments. In the 2D environment, the authors used transform distributions including scaling, rotation, illumination, Gaussian noise, and other transformations, and the expectation of the transformations was calculated by randomly sampling 1000 transformations from these transform distributions.

In the 3D environment, the authors modeled the transformations by 3D rendering, considering camera distance, lateral translation, object rotation, and solid color background, and randomly sampled 100 transformations to represent the distribution, Figure 16.16 shows the 3D adversarial sample example. In the 3D physical environment, the authors used 3D model printing to fabricate an adversarial sample model, considering various angles and backgrounds for the classification of the adversarial samples.(Detailed in the subsection 3.2 and 3.3)

5.6 Conclusion

The authors prove the existence of robust 3D adversarial objects and present the first algorithm for synthesizing adversarial samples over a selected transform distribution.

The authors synthesize 2D adversarial images that are robust to noise, distortions and affine transformations.

Finally, the authors apply the algorithm to complex 3D objects, using 3d printing to fabricate the first physical adversarial object.

6 A List Of Articles

References

[1] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," *arXiv preprint arXiv:1312.6199*, 2013.

[2] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *arXiv preprint arXiv:1412.6572*, 2014.

[3] A. Kurakin, I. Goodfellow, S. Bengio, *et al.*, "Adversarial examples in the physical world," 2016.

[4] J. H. Metzen, T. Genewein, V. Fischer, and B. Bischoff, "On detecting adversarial perturbations," *arXiv preprint arXiv:1702.04267*, 2017.

[5] A. Athalye, L. Engstrom, A. Ilyas, and K. Kwok, "Synthesizing robust adversarial examples," in *International conference on machine learning*, pp. 284–293, PMLR, 2018.