Adversarial Examples Generation and Defense Based on Generative Adversarial Network

Fei Xia, Ruishan Liu
Department of Electrical Engineering, Stanford University

Abstract

We propose a novel method to generate and defend adversarial examples for deep neural networks (DNN). The adversarial stability of network D is improved by training alternatively with an additional network G. We show that complicated adversarial patterns are generated and the target network D classifies perturbed correctly after the training.

Introduction

Currently, some machine learning models including deep neural networks (DNN) are known to be susceptible *to adversarial examples*, i.e., small input perturbations which lead to wrong predictions[1].

Taking DNN as an example, imperceptible distortion of the input data can lead to 100% misclassification for every example.

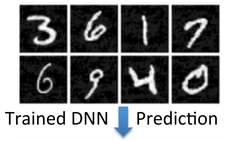


Figure 1. Original Examples

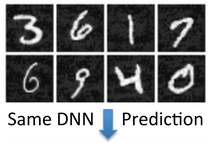


Figure 2. Adversarial Examples.

Although the cause of the DNN adversarial instability is still under debate, training on adversarial samples have been proposed to improve robustness[2,3].

In these studies, however, the widely-used gradient methods for optimizations are too computational expensive, i.e., such data are hard to obtain. In our project, we propose a more efficient way to increase adversarial stability utilizing a novel DNN structure.

Model

Our model consists of two parts: a classical convolutional neural network D and an additional network G, whose source domain is an image and target is the adversarial perturbation.

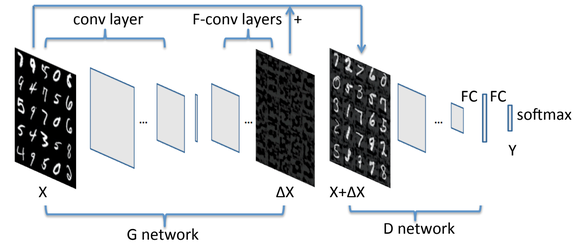


Figure 3. Proposed generative adversarial network (GAN) structure

The network is trained in an adversarial setting:

1) While training D, we freeze the parameters of G and add raw examples in the same time.
2) In the process of training G, we flip the loss function of D for the adversarial purpose, and the norm of ΔX is restricted to ensure it to be a small perturbation . The objective function is shown below

$$\arg\max_{\Theta} \mathbf{CrossEntropy}[D_\Phi(G_\Theta(X)+X, Y)]$$

$$\arg\min_{\Phi} \mathbf{CrossEntropy}[D_\Phi(G_\Theta(X)+X, Y)]$$

During the training process, G and D gradually reach non cooperative equilibrium, and D is expected to become less vulnerable to adversarial examples.

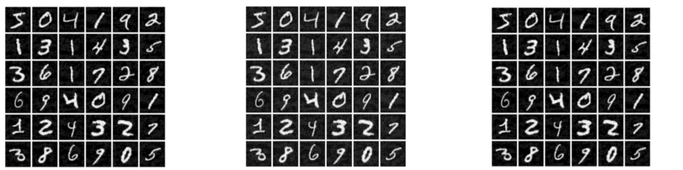
Data & Feature

We used MNIST database: handwritten digits with a training set of 60,000 examples, and a test set of 10,000 examples. It is a subset of a larger set available from NIST. The digits have been size-normalized and centered in a fixed-size image.

Results

High accuracy of 98.4% is achieved after the Network D is training on MNIST for 70 epochs. Then network G is added and successfully generates adversarial examples. After training our model for 200 epochs, the accuracy of network D is decreased from 98.4% to 55.1%.

As a comparison, after one step, the commonly used method - fast gradient - decreases the accuracy from 98.4% to 76.5%[4].



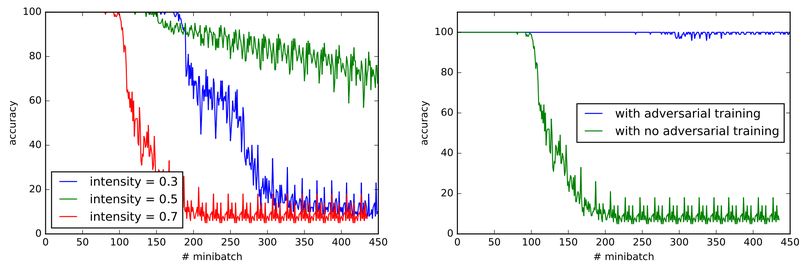
(a) Before G (b) After G (c) Generated using fast gradient

Figure 2 Original image and perturbed image using GAN and FastGrad

Table 1. Accuracy of 1-step perturbation .
To prove our model, we tested accuracy for one step perturbation. For FastGrad, it's a back propagation on D. For our model, it is a forward propagation on G. It shows that better results are achieved using our approach for most intensity levels (defined as maximum perturbation).

Intensity	FastGrad	GAN
0.1	**0.96**	0.97
0.3	0.76	**0.55**
0.5	0.37	**0.15**
0.7	0.06	**0.05**

Figure 1(a) plots the learning accuracy curve for different intensities, when D is frozen and only G is trained. It is found that the larger the perturbation intensity is, the less epochs are required to train a network G till it converges.
Figure 1(b) shows the learning accuracy curve when G and D are trained alternatively. We note that the network remains near 100% accuracy throughout the training process, indicating the robustness of our target network D against the adversarial examples G generated.



(a) Accuracy comparison for different perturbation intensity, without adversarial training

(b) Accuracy comparison for different perturbation intensity, with/without adversarial training

Figure 4. Learning curve.

Discussion & Future work

Discussion

The biggest advantage of our model is that more complicated adversarial patterns can be extracted by training the two networks D and G alternatively.

For most current adversarial researches[1-4], the target neural network is always frozen to be attacked or designed to defend. Relatively simple patterns could be found as shown in Fig. 4. In our scenario, however, the target network D is updated at the same time with the training of the attacker G, which results in more complicated perturbation patterns.

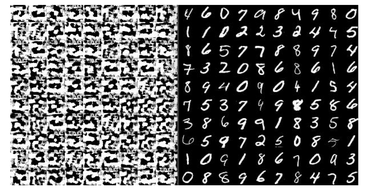


Figure 5. Freeze target network D

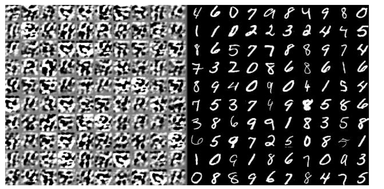


Figure 6. Train D and G alternatively

We address that this complexity feature of our adversarial examples has a great potential in practical applications. For example, captcha with orderly perturbed background is more likely to be observed by the attacker and thus more vulnerable to the real attacks.

We also point out that, in the previous settings, the intensity of the perturbation is defined as ∞– norm, originally used to model the quantization error in images. It should be further discussed cause ∞–norm doesn't capture the 'distance' between true sample and adversarial sample and does not fully capture how human perceive the difference of two image.

Future work

(1) Generalization of adversarial examples: can the adversarial examples generated by G and D performs well against another network D', which is trained on a similar dataset.
(2) Adversarial example defense using regularization: In robust optimization research, adversarial example defense is modeled as a min-max problem related to regularization. It is worth studying how our approach is related to regularization

Contact

Fei Xia
feixia@stanford.edu

Ruishan Liu
ruishan@stanford. edu

Adversarial training livestream, Realtime visualizing the adversarial examples: http://128.12.146.201:8000 (or QR code on the left)
Github Repo (a torch implementation): https://github.com/fxia22/advGAN

References

1. Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. arXiv preprint arXiv: 1312.6199, 2013.
2. Osbert Bastani, Yani Ioannou, Leonidas Lampropoulos, Dimitrios Vytiniotis, Aditya Nori, and Antonio Criminisi. Measuring neural net robustness with constraints. arXiv preprint arXiv: 1605.07262, 2016.
3. Uri Shaham, Yutaro Yamada, and Sahand Negahban. Understanding adversarial training: Increasing local stability of neural nets through robust optimization. arXiv preprint arXiv: 1511.05432, 2015.
4. Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial machine learning at scale. arXiv preprint arXiv:1611.01236, 2016.