Adversarial Examples Generation and Defense Based on Generative Adversarial Network

Fei Xia (06082760), Ruishan Liu (06119690)

December 15, 2016

1 Abstract

We propose a novel generative adversarial network to generate and defend adversarial examples for deep neural networks (DNN). The adversarial stability of a network D is improved by training alternatively with an additional network G. Our experiment is carried out on MNIST, and the adversarial examples are generated in an efficient way compared with wildly-used gradient based methods. After training network D and G alternatively, complicated adversarial patterns are extracted. The target network D is demonstrated to become robust against the adversarial attack from G. Our approach provides a broadly new viewpoint for the adversarial instability researches, and have various critical applications in practice.

2 Introduction

Currently, some machine learning models including deep neural networks (DNN) are known to be susceptible to *adversarial examples*, 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 [1]. Certain adversarial examples can be easily generated by optimizing input to maximize the objective error, through gradient methods [2, 1, 3, 4, 5], evolutionary methods [6], etc. Defense towards adversarial examples is thus meaningful and urgent, both for decreasing the gap between the machines and the human perception, and for the security of DNN applications such as license plate recognition, face recognition and autonomous driving.

Although the cause of the DNN adversarial instability is still under debate [7, 3, 1, 8], different methods have been proposed to improve robustness [9, 10, 11, 5, 12, 13]. Training on adversarial samples has been demonstrated to be effective and considered as a good practice of regularization [12, 13]. 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 an efficient way to increase adversarial stability on the basis of generating and training on adversarial examples.

3 Related Work

The theory about DNN adversarial instability has been studied in many researches. Ref. [3] gave the reason that the summation of small perturbations can be large, i.e., "small effects in hundreds of dimensions adding up to create a large effect". However, Ref. [7] argues that the adversarial instability arises from "the low flexibility of classifiers compared to the difficulty of the classification task". There are also other explanations. Ref. [1] states that the adversarial instability is due to the linear nature of the neural network. Ref. [8] further verifies this viewpoint.

Current experiments on adversarial examples generation and defense adopt different methods to maximize the objective error. Firstly, gradient methods are most frequently used. Other approaches such as evolutionary methods [6] are intrinsically more computational expensive than gradient descent methods.

Secondly, the traditional gradient descend methods [2, 1, 3, 4, 5] have been demonstrated to be successful in generating and defending adversarial examples. Lastly, fast gradient descend method is proposed recently as an improvement of efficiency on traditional gradient descent methods[16]. Thus in our project, we mainly compare our result with the fast gradient descend method, which is highly effective in existing approaches.

4 Dataset and Features

In our experiment, we used MNIST database, a subset of a larger set available from NIST. More specifically, we used handwritten digits with a training set of 60,000 examples, and a test set of 10,000 examples. The digits have been size-normalized and centered in a fixed-size image.

5 Methods

We propose a method to resist adversarial perturbation of DNN in an *adversarial* [14] setting. Instead of carrying out optimization for each example, we train a time-saving network that automatically does this, namely a *pixel level domain transfer* [15]. A tentative architecture of the proposed model is illustrated in Figure 1. For the additional network G, the source domain is an image, and the target domain is the adversarial perturbation.

The objective functions are given by

$$\arg\max_{\Theta} \textbf{CrossEntropy}[D_{\Phi}(G_{\Theta}(X) + X, Y)] \tag{1}$$

$$\arg\min_{\Phi} \textbf{CrossEntropy}[D_{\Phi}(G_{\Theta}(X) + X, Y)] \tag{2}$$

where Θ is parameters of generator G, Φ is parameters of discriminator D, X stands for the input image, Y represents the label of Y, and **CrossEntropy** is the loss function we choose for classification.

The network is trained in an adversarial setting. While training D, we freeze the parameters of G. In addition, we add raw examples when training G. In the process of training G, as indicated by Eq. (1) and (2), we flip the loss function of D for the adversarial purpose. The norm of ΔX is also restricted to ensure it to be a small perturbation . During the training process, G and D gradually reach non cooperative equilibrium, and D is expected to become less vulnerable to adversarial examples.

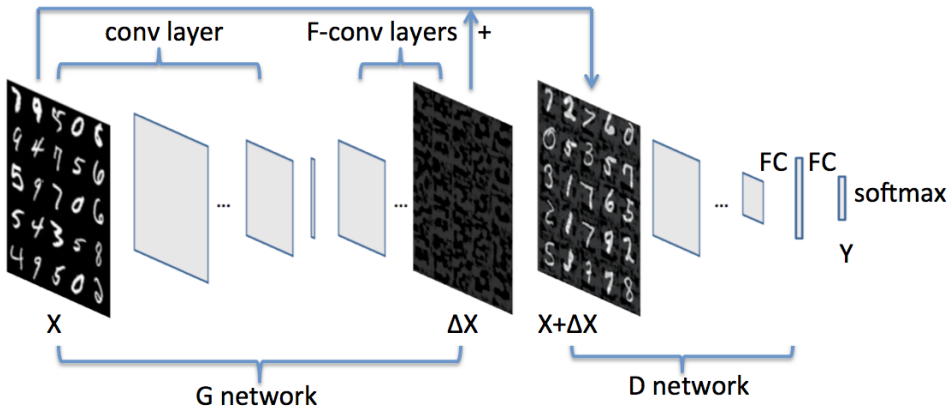
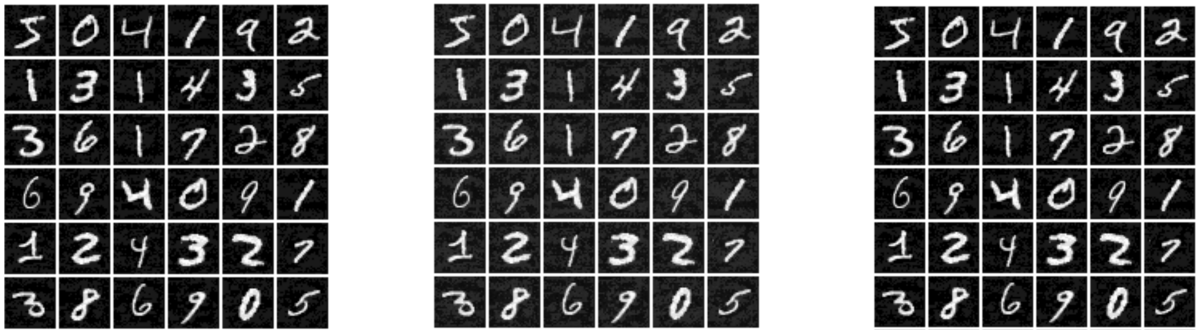


Figure 1: Proposed generative adversarial network (GAN) structure

6 Results

We implemented our proposed network in Torch 7[1]. A classical convolutional neural network structure is used for network D, which is trained on MNIST for 70 epochs and frozen when G is being trained. High accuracy of about 98.4% is achieved after the initial step. In the meantime, for network G, in order to limit the magnitude of perturbation, we add a clamp layer to the perturbation, which is restricted within $[-t, t]$, where t is the predefined intensity of perturbation (defined by $\infty-$ norm).



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

Figure 2: Original image and perturbed image using GAN and FastGrad

We then show that network G successfully generates adversarial examples, and is effective compared with traditional fast gradient method [16]. After training for 200 epochs, the accuracy is decreased from 98.4% to 55.1%. The images before G and after G are shown in Fig 2, indicating the perturbations are imperceptible. As a comparison, after one step, fast gradient decreases the accuracy from 98.4% to 76.5%. The corresponding adversarial examples are also shown in Fig 2, which also denotes a successful adversarial examples generation.

The effectiveness of our model is proven by testing the accuracy test for one step perturbation under different noise intensity, as shown in Table. 1. It is found that better results are achieved using our approach for most intensity levels, which is defined as maximum perturbation. Note that the required time for one step of those two methods is similar. Generative model requires one forward pass of net G, fast gradient compute gradient of net D, which requires a backward pass of net D.

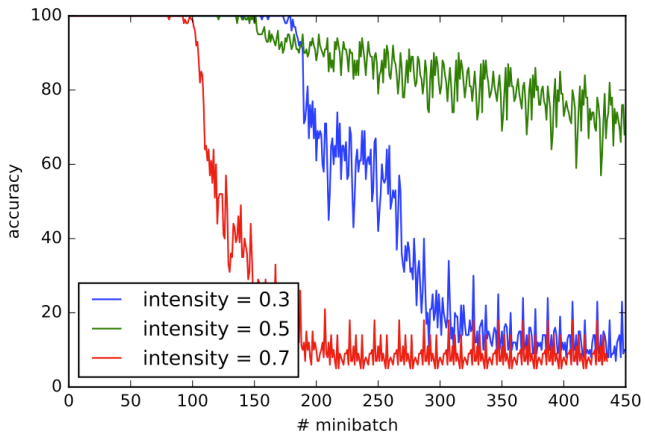
	Accuracy of FastGrad	Accuracy of GAN
Intensity = 0.1	0.96	0.97
Intensity = 0.3	0.76	0.55
Intensity = 0.5	0.37	0.15
Intensity = 0.7	0.06	0.05

Table 1: Accuracy of 1-step perturbation with different noise intensity for two methods.

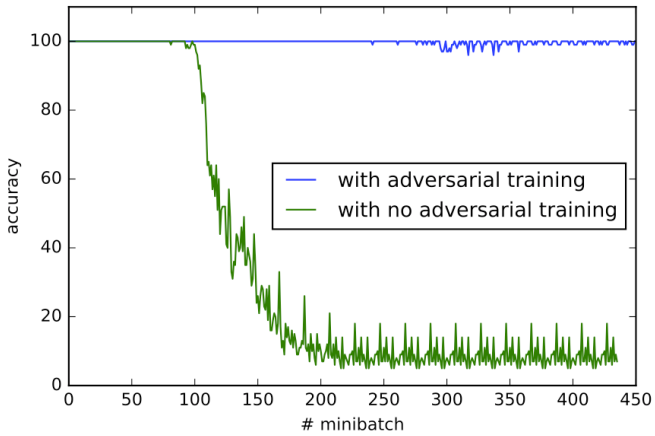
We further analyze the performance of our model for different noise level. As plotted in Fig. 3a, the learning accuracy curve varies for different noise intensities, when D is frozen and G is trained to generate adversarial examples. As expected, the larger the perturbation intensity is, the less epochs are required to train a network G till it converges.

Finally, in order to improve the adversarial stability of the target network D, we trained network D and G alternatively. In this case, the learning accuracy curve is shown in Fig. 3b. 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.

[1] http://torch.ch



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



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

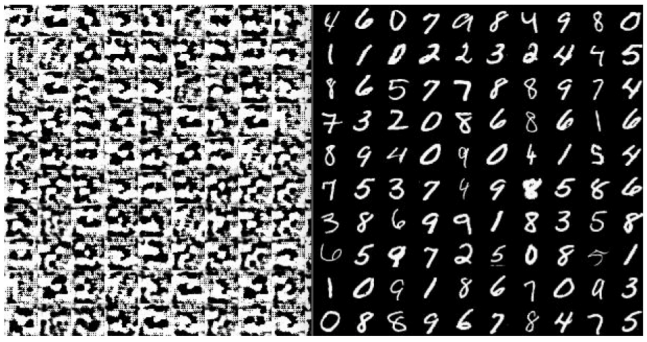
Figure 3: Learning Curve

7 Discussion

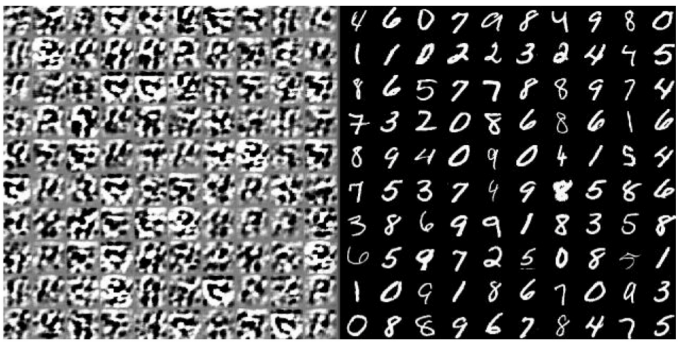
One advantages of our model is that more complicated adversarial patterns can be extracted by training the two networks D and G alternatively. For most current adversarial researches, the target neural network is always frozen to be attacked or designed to defend. Relatively simple patterns could be 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.

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.

It should also be pointed out that, in the previous settings, the intensity of the perturbation is defined as $\infty-$ norm. This is worth discussion because $\infty-$ norm does not capture the 'distance' between true sample and adversarial sample. The notion of $\infty-$ norm for adversarial samples is originally used to model the quantization error in images. However, it does not fully capture how human perceive the difference of two image.



(a) Freeze target network D



(b) Train D and G alternatively

Figure 4: Scaled noise and original image.

8 Future Work

The universality of adversarial examples generation and defense is a hot topic today and is also worth investigating in our model. This problem can be divided into two parts. The first is whether the adversarial examples are universal, i.e., whether the adversarial examples generated by G and D can also performs well against another network D' which is trained on a similar dataset. The second problem is whether the adversarial robustness is universal, i.e., whether the trained-to-be-robust network D can be still robust to another attacker with different network structure or even another attacker using other methods. The latter is actually a critically unsolved problem nowadays, the solution to which may involve finding the true reason of the adversarial instability.

Another thing we can do in the future is to get more theoretical insight of our model. In the current robust optimization researches, adversarial example defense is modeled as a min-max problem which is related to regularization. It is worth studying how our method can be related to more general approaches such as regularization.

9 Conclusion

In the project, we propose to use a generative adversarial network to generate and defend adversarial examples for DNN. The novel model consists of a classical convolutional neural network D and an additional network G to produce the adversarial perturbation. In our experiment on MNIST, adversarial examples are successfully generated by network G. The accuracy of the target network is decreased from 98.4% to 55.1%, indicating a high efficiency compared with wildly-used gradient based methods. We further show that by training network D and G alternatively, our target network D successfully becomes robust against the adversarial examples generated by G. Complicated adversarial patterns are extracted. Our method provides a new approach for adversarial examples generation and defense, and has a great potential in practical applications such as improving DNN security.

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] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. *arXiv preprint arXiv:1608.04644*, 2016.

[3] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

[4] Chunchuan Lyu, Kaizhu Huang, and Hai-Ning Liang. A unified gradient regularization family for adversarial examples. In *Data Mining (ICDM), 2015 IEEE International Conference on*, pages 301–309. IEEE, 2015.

[5] Jiashi Feng, Tom Zahavy, Bingyi Kang, Huan Xu, and Shie Mannor. Ensemble robustness of deep learning algorithms. *arXiv preprint arXiv:1602.02389*, 2016.

[6] Anh Nguyen, Jason Yosinski, and Jeff Clune. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 427–436. IEEE, 2015.

[7] Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. Analysis of classifiers' robustness to adversarial perturbations. *arXiv preprint arXiv:1502.02590*, 2015.

[8] Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. Fundamental limits on adversarial robustness. *ICML 2015*, 2015.

[9] Shixiang Gu and Luca Rigazio. Towards deep neural network architectures robust to adversarial examples. *NIPS Workshop on Deep Learning and Representation Learning*, 2014.

[10] Krzysztof Chalupka, Pietro Perona, and Frederick Eberhardt. Visual causal feature learning. *arXiv preprint arXiv:1412.2309*, 2014.

[11] Nicolas Papernot, Patrick McDaniel, Xi Wu, Somesh Jha, and Ananthram Swami. Distillation as a defense to adversarial perturbations against deep neural networks. *arXiv preprint arXiv:1511.04508*, 2015.

[12] 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.

[13] 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.

[14] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 2672–2680, 2014.

[15] Donggeun Yoo, Namil Kim, Sunggyun Park, Anthony S Paek, and In So Kweon. Pixel-level domain transfer. *arXiv preprint arXiv:1603.07442*, 2016.

[16] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial machine learning at scale. *arXiv preprint arXiv:1611.01236*, 2016.