Iterative Ensemble Adversarial Attack

Jiayang Liu[*]

School of Information Science and Technology, University of Science and Technology of China

Abstract. Deep Neural Networks (DNNs) have recently led to significant improvements in many fields. However, DNNs are vulnerable to adversarial examples which are samples with imperceptible perturbations while dramatically misleading the DNNs. Adversarial attacks can be used to evaluate the robustness of deep learning models before they are deployed. Unfortunately, most of existing adversarial attacks can only fool a black-box model with a low success rate. To improve the success rates for black-box adversarial attacks, we proposed an iterated adversarial attack against an ensemble of image classifiers. With this method, we won the 5th place in CAAD 2018 Targeted Adversarial Attack competition.

1 Introduction

Deep Neural Networks (DNNs) have recently led to significant improvements in many fields, such as image classification [1, 2] and speech recognition [3]. However, the generalization properties of the DNNs have been recently questioned because these machine learning models are vulnerable to adversarial examples [4]. An adversarial example is a slightly modified sample that is intended to cause an error output of the DNN based model. In the context of classification task, the adversarial example is crafted to force a model to classify it into a class different from the legitimate class. In addition, adversarial examples have cross-model transferability property [5]. The adversarial examples crafted for one model remain adversarial for others, thus making black-box attacks practical in real-world applications. Unfortunately, most of existing adversarial attacks can only fool a black-box model with a low success rate. To improve the success rates for black-box adversarial attacks, we proposed an iterated adversarial attack against an ensemble of image classifiers.

Adversarial attacks are divided into two types: targeted attack and untargeted attack. In targeted attack, the attacker generates adversarial examples which are misclassified by the classifier into a particular class. In untargeted attack, the attacker generates adversarial examples which are misclassified by the classifier into any class as long as it is different from the true class.

---

2   Related Work

2.1   Fast Gradient Sign Method

Goodfellow et al. [5] propose the Fast Gradient Sign Method (FGSM) for generating adversarial examples. This method uses the derivative of the loss function of the model pertaining to the input feature vector. Given the input image X, FGSM is to perturb the gradient direction of each feature by the gradient. Then the classification result of the input image will be changed. For a neural network with cross-entropy cost function $J(X, y)$ where X is the input image and y_t is the target class for the input image, the adversarial example is generated as

$$X^{adv} = X - \epsilon \text{sign}(\nabla_X J(X, y_t)) \tag{1}$$

where ϵ is a parameter to determine the perturbation size.

2.2   Iterative Gradient Sign Method

The Iterative Gradient Sign Method (IGSM) is the iterative version of FGSM. This method applies FGSM many times with small perturbation size instead of applying adversarial noise with one large perturbation size. The adversarial example of the iterative gradient sign method is generated as

$$\begin{aligned} X_0^{adv} &= X, \\ X_{N+1}^{adv} &= Clip_{X,\epsilon}\{X_N^{adv} - \alpha \text{sign}(\nabla_X J(X_N^{adv}, y_t))\} \end{aligned} \tag{2}$$

where $Clip_{X,\epsilon}\{X'\}$ represents a clipping of the values of the adversarial example. So the results are within ϵ-neighbourhood of the input image X. This attack is more powerful because the attacker can control how far the adversarial example past the classification boundary. It was demonstrated that the attack of IGSM was better than FGSM on ImageNet dataset [6].

2.3   Carlini & Wagner Method

This method is named after its authors [7]. The attack can be targeted or untargeted, and has three metrics to measure its distortion (l_0 norm, l_2 norm and l_∞ norm). The authors point out that the untargeted l_2 norm version has the best performance. It generates adversarial examples by solving the following optimization problem:

$$\begin{aligned} \underset{\delta}{\text{minimize}} \ \|\delta\|_2 + c \cdot f(x + \delta) \\ s.t. \quad x + \delta \in [0, 1]^n \end{aligned} \tag{3}$$

This attack is to look for the smallest perturbation measured by l_2 norm and make the network classify the image incorrectly at the same time. c is a hyperparameter to balance the two parts of equation (3). The best way to

choose c is to use the smallest value of c for which the resulting solution $x + \delta$ has $f(x + \delta) \leq 0$. $f(x)$ is the loss function to measure the distance between the input image and the adversarial image. $f(x)$ is defined as:

$$f(x) = \max(Z(x)_{true} - \max_{i \neq true} \{Z(x)_i\}, -\kappa) \tag{4}$$

$Z(x)$ is the pre-softmax classification result vector. κ is a hyper-parameter called confidence. Higher confidence encourages the attack to search for adversarial examples that are stronger in classification confidence. High-confidence attacks often have larger perturbations and better transferability to other models.

3   Proposed Method

3.1   Ensemble Attack Method

An intriguing property of adversarial examples is that adversarial examples generated for one classifier transfer to other classifiers [5]. Therefore, a natural method for effective adversarial attacks against unknown classifiers is to generate strong adversarial examples against a large collection of classifiers. We propose to perform an iterated targeted adversarial attack against an ensemble of image classifiers. In our solution we give equal weights to each of classifiers. The classifiers in the ensemble are: Inception V3 [8], ResNet 50 [2], ResNet 101 [2], Inception ResNet V2 [9], ensemble adversarially trained Inception ResNet V2 (EnsAdv Inception ResNet V2) [10] and adversarially trained Inception V3 (Adv Inception V3) [11]. Our method builds iterative attacks on a loss ensemble neural networks focusing on the classifiers that are difficult to perturb.

It is a common way to combine a number of classifiers by probability ensemble. However, it is usually hard to optimize the loss function because the parameters of different classifiers are coupled inside the logarithm.

$$J_{prob}(X, y) = -\sum_{i}^{N} y_j \log \left(\frac{1}{M} \sum_{i}^{M} p_{ij}(X) \right) \tag{5}$$

An upper bound is obtained for the loss function by Jensen's inequality. We propose to optimize the upper bound instead of minimizing $J_{prob}(X, y)$. This way of combining classifiers in Eq.6 is called loss ensemble. The parameters of different neural networks are decoupled in the new loss function $J_{loss}(X, y)$, which is benefit to the optimization.

$$J_{prob}(X, y) \leq -\frac{1}{M} \sum_{j}^{N} \sum_{i}^{M} y_{ij} \log(p_{ij}(X)) = J_{loss}(X, y) \tag{6}$$

4 Jiayang Liu

3.2 Iterative Ensemble Attack

The proposed iterative ensemble attack is based on IGSM. We compare our iterative attack method with different number of iterations. Figure 1 demonstrates that attacking an adversarial trained model at high success rate requires more iterations than a normal model. For tasks where computation is limited, we preassign the number of iterations for each model or dynamically adjust whether a model is included in each step of the attack by observing if the loss function for that model is small enough. In Eq.7, $\delta_{Nk} \in \{0, 1\}$ determines whether to include the loss for model k in the total loss at step N.

$$X_0^{adv} = X,$$

$$X_{N+1}^{adv} = Clip_{X,\epsilon}\left\{X_N^{adv} - \alpha\,\mathrm{sign}\left(\frac{1}{M}\sum_k^M \delta_{Nk}\nabla_X J_k\left(X_N^{adv}, y_t\right)\right)\right\} \qquad (7)$$
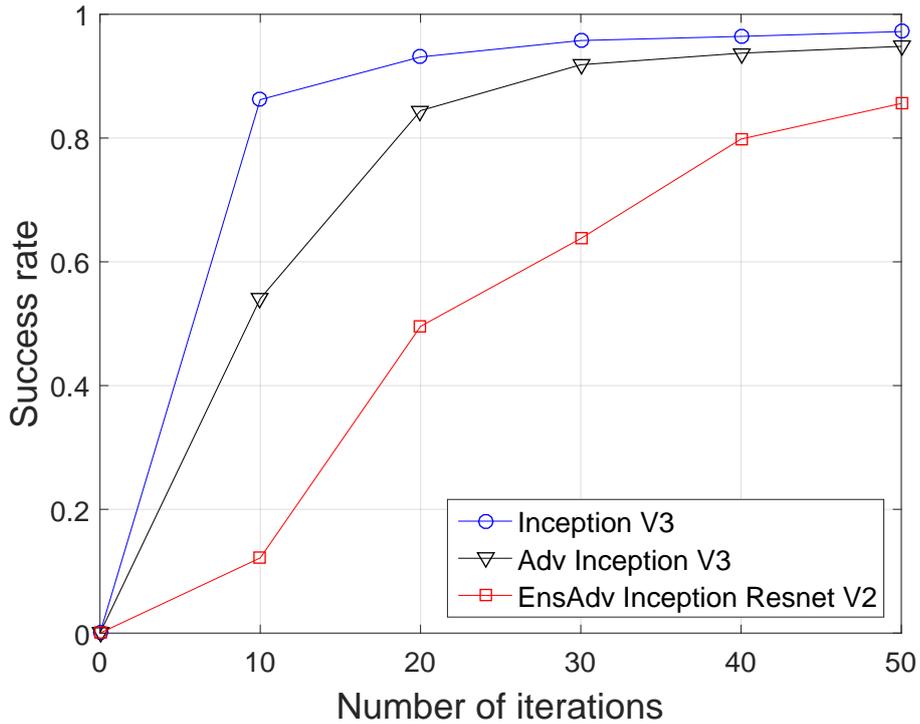
Fig. 1: Iterative ensemble attack results for three selected models.

4   Conclusion

In this paper, we propose an iterated adversarial attack against an ensemble of image classifiers, which can improve the success rates for black-box adversarial attacks. With this method, we won the 5th place in CAAD 2018 Targeted Adversarial Attack competition. It is easy to make our attack stronger by adding more models or increasing the number of iterations.

References

1. O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al., Imagenet large scale visual recognition challenge, International Journal of Computer Vision 115 (3) (2015) 211–252.
2. K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.
3. D. Amodei, R. Anubhai, E. Battenberg, C. Case, J. Casper, B. Catanzaro, J. Chen, M. Chrzanowski, A. Coates, G. Diamos, et al., Deep speech 2: End-to-end speech recognition in english and mandarin, in: International Conference on Machine Learning, 2016, pp. 173–182.
4. C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, R. Fergus, Intriguing properties of neural networks, International Conference on Learning Representations.
5. I. J. Goodfellow, J. Shlens, C. Szegedy, Explaining and harnessing adversarial examples, in: ICML, 2015, pp. 1–10.
6. A. Kurakin, I. Goodfellow, S. Bengio, Adversarial examples in the physical world, International Conference on Learning Representations.
7. N. Carlini, D. Wagner, Towards evaluating the robustness of neural networks, in: Security and Privacy, 2017.
8. C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, Z. Wojna, Rethinking the inception architecture for computer vision, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 2818–2826.
9. C. Szegedy, S. Ioffe, V. Vanhoucke, A. A. Alemi, Inception-v4, inception-resnet and the impact of residual connections on learning., in: AAAI, Vol. 4, 2017, p. 12.
10. F. Tramèr, A. Kurakin, N. Papernot, I. Goodfellow, D. Boneh, P. McDaniel, Ensemble adversarial training: Attacks and defenses, arXiv preprint arXiv:1705.07204.
11. A. Kurakin, I. Goodfellow, S. Bengio, Adversarial machine learning at scale, arXiv preprint arXiv:1611.01236.