Boosting Adversarial Attacks with Momentum

Yinpeng Dong[1], Fangzhou Liao[1], Tianyu Pang[1], Hang Su[1], Jun Zhu[1,*], Xiaolin Hu[1], Jianguo Li[2]
[1] Department of Computer Science and Technology, Tsinghua Lab of Brain and Intelligence
[1] Beijing National Research Center for Information Science and Technology, BNRist Lab
[1] Tsinghua University, 100084 China
[2] Intel Labs China
{dyp17, liaofz13, pty17}@mails.tsinghua.edu.cn, {suhangss, dcszj, xlhu}@mail.tsinghua.edu.cn, jianguo.li@intel.com

Abstract

Deep neural networks are vulnerable to adversarial examples, which poses security concerns on these algorithms due to the potentially severe consequences. Adversarial attacks serve as an important surrogate to evaluate the robustness of deep learning models before they are deployed. However, most of existing adversarial attacks can only fool a black-box model with a low success rate. To address this issue, we propose a broad class of momentum-based iterative algorithms to boost adversarial attacks. By integrating the momentum term into the iterative process for attacks, our methods can stabilize update directions and escape from poor local maxima during the iterations, resulting in more transferable adversarial examples. To further improve the success rates for black-box attacks, we apply momentum iterative algorithms to an ensemble of models, and show that the adversarially trained models with a strong defense ability are also vulnerable to our black-box attacks. We hope that the proposed methods will serve as a benchmark for evaluating the robustness of various deep models and defense methods. With this method, we won the first places in NIPS 2017 Non-targeted Adversarial Attack and Targeted Adversarial Attack competitions.

1. Introduction

Deep neural networks (DNNs) are challenged by their vulnerability to adversarial examples [23, 5], which are crafted by adding small, human-imperceptible noises to legitimate examples, but make a model output attacker-desired inaccurate predictions. It has garnered an increasing attention to generating adversarial examples since it helps to identify the vulnerability of the models before they are launched. Besides, adversarial samples also facilitate various DNN algorithms to assess the robustness by providing more varied training data [5, 10].

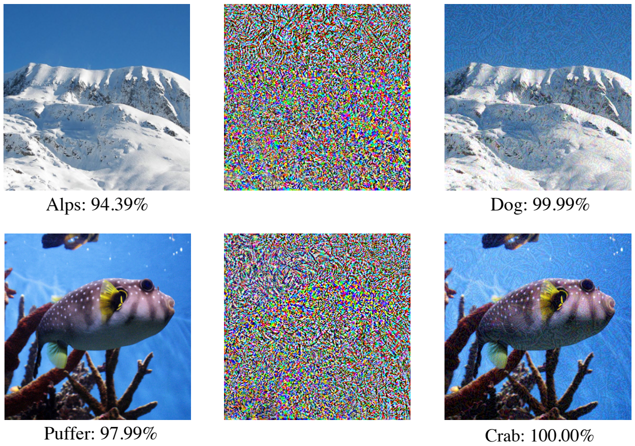
Alps: 94.39%
Dog: 99.99%
Puffer: 97.99%
Crab: 100.00%

Figure 1. We show two adversarial examples generated by the proposed momentum iterative fast gradient sign method (MI-FGSM) for the Inception v3 [22] model. **Left column**: the original images. **Middle column**: the adversarial noises by applying MI-FGSM for 10 iterations. **Right column**: the generated adversarial images. We also show the predicted labels and probabilities of these images given by the Inception v3.

With the knowledge of the structure and parameters of a given model, many methods can successfully generate adversarial examples in the white-box manner, including optimization-based methods such as box-constrained L-BFGS [23], one-step gradient-based methods such as fast gradient sign [5] and iterative variants of gradient-based methods [9]. In general, a more severe issue of adversarial examples is their good transferability [23, 12, 14], *i.e.*, the adversarial examples crafted for one model remain adversarial for others, thus making black-box attacks practical in real-world applications and posing real security issues. The phenomenon of transferability is due to the fact that different machine learning models learn similar decision boundaries around a data point, making the adversarial examples crafted for one model also effective for others.

[*] Corresponding author.

However, existing attack methods exhibit low efficacy when attacking black-box models, especially for those with a defense mechanism. For example, ensemble adversarial training [24] significantly improves the robustness of deep neural networks and most of existing methods cannot successfully attack them in the black-box manner. This fact largely attributes to the trade-off between the attack ability and the transferability. In particular, the adversarial examples generated by optimization-based and iterative methods have poor transferability [10], and thus make black-box attacks less effective. On the other hand, one-step gradient-based methods generate more transferable adversarial examples, however they usually have a low success rate for the white-box model [10], making it ineffective for black-box attacks. Given the difficulties of practical black-box attacks, Papernot *et al.* [16] use adaptive queries to train a surrogate model to fully characterize the behavior of the target model and therefore turn the black-box attacks to white-box attacks. However, it requires the full prediction confidences given by the target model and tremendous number of queries, especially for large scale datasets such as ImageNet [19]. Such requirements are impractical in real-world applications. Therefore, we consider how to effectively attack a black-box model without knowing its architecture and parameters, and further, without querying.

In this paper, we propose a broad class of *momentum iterative gradient-based* methods to boost the success rates of the generated adversarial examples. Beyond iterative gradient-based methods that iteratively perturb the input with the gradients to maximize the loss function [5], momentum-based methods accumulate a velocity vector in the gradient direction of the loss function across iterations, for the purpose of stabilizing update directions and escaping from poor local maxima. We show that the adversarial examples generated by momentum iterative methods have higher success rates in both white-box and black-box attacks. The proposed methods alleviate the trade-off between the white-box attacks and the transferability, and act as a stronger attack algorithm than one-step methods [5] and vanilla iterative methods [9].

To further improve the transferability of adversarial examples, we study several approaches for attacking an ensemble of models, because if an adversarial example fools multiple models, it is more likely to remain adversarial for other black-box models [12]. We show that the adversarial examples generated by the momentum iterative methods for multiple models, can successfully fool robust models obtained by ensemble adversarial training [24] in the black-box manner. The findings in this paper raise new security issues for developing more robust deep learning models, with a hope that our attacks will be used as a benchmark to evaluate the robustness of various deep learning models and defense methods. In summary, we make the following contributions:

- We introduce a class of attack algorithms called momentum iterative gradient-based methods, in which we accumulate gradients of the loss function at each iteration to stabilize optimization and escape from poor local maxima.
- We study several ensemble approaches to attack multiple models simultaneously, which demonstrates a powerful capability of transferability by preserving a high success rate of attacks.
- We are the first to show that the models obtained by ensemble adversarial training with a powerful defense ability are also vulnerable to the black-box attacks.

2. Backgrounds

In this section, we provide the background knowledge as well as review the related works about adversarial attack and defense methods. Given a classifier $f(\boldsymbol{x}) : \boldsymbol{x} \in \mathcal{X} \rightarrow y \in \mathcal{Y}$ that outputs a label y as the prediction for an input $\boldsymbol{x}$, the goal of adversarial attacks is to seek an example $\boldsymbol{x}^*$ in the vicinity of $\boldsymbol{x}$ but is misclassified by the classifier. Specifically, there are two classes of adversarial examples—*non-targeted* and *targeted* ones. For a correctly classified input $\boldsymbol{x}$ with ground-truth label y such that $f(\boldsymbol{x}) = y$, a non-targeted adversarial example $\boldsymbol{x}^*$ is crafted by adding small noise to $\boldsymbol{x}$ without changing the label, but misleads the classifier as $f(\boldsymbol{x}^*) \neq y$; and a targeted adversarial example aims to fool the classifier by outputting a specific label as $f(\boldsymbol{x}^*) = y^*$, where y^* is the target label specified by the adversary, and $y^* \neq y$. In most cases, the L_p norm of the adversarial noise is required to be less than an allowed value ϵ as $\|\boldsymbol{x}^* - \boldsymbol{x}\|_p \leq \epsilon$, where p could be $0, 1, 2, \infty$.

2.1. Attack methods

Existing approaches for generating adversarial examples can be categorized into three groups. We introduce their non-targeted version of attacks here, and the targeted version can be simply derived.

One-step gradient-based approaches, such as the fast gradient sign method (FGSM) [5], find an adversarial example $\boldsymbol{x}^*$ by maximizing the loss function $J(\boldsymbol{x}^*, y)$, where J is often the cross-entropy loss. FGSM generates adversarial examples to meet the L_∞ norm bound $\|\boldsymbol{x}^* - \boldsymbol{x}\|_\infty \leq \epsilon$ as

$$\boldsymbol{x}^* = \boldsymbol{x} + \epsilon \cdot \mathrm{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{x}, y)), \tag{1}$$

where $\nabla_{\boldsymbol{x}} J(\boldsymbol{x}, y)$ is the gradient of the loss function w.r.t. $\boldsymbol{x}$. The fast gradient method (FGM) is a generalization of FGSM to meet the L_2 norm bound $\|\boldsymbol{x}^* - \boldsymbol{x}\|_2 \leq \epsilon$ as

$$\boldsymbol{x}^* = \boldsymbol{x} + \epsilon \cdot \frac{\nabla_{\boldsymbol{x}} J(\boldsymbol{x}, y)}{\|\nabla_{\boldsymbol{x}} J(\boldsymbol{x}, y)\|_2}. \tag{2}$$

Iterative methods [9] iteratively apply fast gradient multiple times with a small step size α. The iterative version of FGSM (I-FGSM) can be expressed as:

$$\boldsymbol{x}_0^* = \boldsymbol{x}, \quad \boldsymbol{x}_{t+1}^* = \boldsymbol{x}_t^* + \alpha \cdot \text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{x}_t^*, y)). \tag{3}$$

To make the generated adversarial examples satisfy the L_∞ (or L_2) bound, one can clip $\boldsymbol{x}_t^*$ into the ϵ vicinity of $\boldsymbol{x}$ or simply set $\alpha = \epsilon/T$ with T being the number of iterations. It has been shown that iterative methods are stronger white-box adversaries than one-step methods at the cost of worse transferability [10, 24].

Optimization-based methods [23] directly optimize the distance between the real and adversarial examples subject to the misclassification of adversarial examples. Box-constrained L-BFGS can be used to solve such a problem. A more sophisticated way [1] is solving:

$$\arg\min_{\boldsymbol{x}^*} \lambda \cdot \|\boldsymbol{x}^* - \boldsymbol{x}\|_p - J(\boldsymbol{x}^*, y). \tag{4}$$

Since it directly optimizes the distance between an adversarial example and the corresponding real example, there is no guarantee that the L_∞ (L_2) distance is less than the required value. Optimization-based methods also lack the efficacy in black-box attacks just like iterative methods.

2.2. Defense methods

Among many attempts [13, 3, 15, 10, 24, 17, 11], adversarial training is the most extensively investigated way to increase the robustness of DNNs [5, 10, 24]. By injecting adversarial examples into the training procedure, the adversarially trained models learn to resist the perturbations in the gradient direction of the loss function. However, they do not confer robustness to black-box attacks due to the coupling of the generation of adversarial examples and the parameters being trained. Ensemble adversarial training [24] augments the training data with the adversarial samples produced not only from the model being trained, but also from other hold-out models. Therefore, the ensemble adversarially trained models are robust against one-step attacks and black-box attacks.

3. Methodology

In this paper, we propose a broad class of **momentum iterative gradient-based methods** to generate adversarial examples, which can fool white-box models as well as black-box models. In this section, we elaborate the proposed algorithms. We first illustrate how to integrate momentum into iterative FGSM, which induces a momentum iterative fast gradient sign method (MI-FGSM) to generate adversarial examples satisfying the L_∞ norm restriction in the non-targeted attack fashion. We then present several methods on how to efficiently attack an ensemble of models. Finally, we extend MI-FGSM to L_2 norm bound and targeted attacks, yielding a broad class of attack methods.

Algorithm 1 MI-FGSM

```
Input: A classifier $f$ with loss function $J$; a real example $\boldsymbol{x}$ and ground-truth label $y$;
Input: The size of perturbation $\epsilon$; iterations $T$ and decay factor $\mu$.
Output: An adversarial example $\boldsymbol{x}^*$ with $\|\boldsymbol{x}^* - \boldsymbol{x}\|_\infty \le \epsilon$.
$\alpha = \epsilon/T$;
$\boldsymbol{g}_0 = 0$; $\boldsymbol{x}_0^* = \boldsymbol{x}$;
for $t = 0$ to $T - 1$ do
    Input $\boldsymbol{x}_t^*$ to $f$ and obtain the gradient $\nabla_{\boldsymbol{x}} J(\boldsymbol{x}_t^*, y)$;
    Update $\boldsymbol{g}_{t+1}$ by accumulating the velocity vector in the gradient direction as
        $\boldsymbol{g}_{t+1} = \mu \cdot \boldsymbol{g}_t + \frac{\nabla_{\boldsymbol{x}} J(\boldsymbol{x}_t^*, y)}{\|\nabla_{\boldsymbol{x}} J(\boldsymbol{x}_t^*, y)\|_1}$;   (6)
    Update $\boldsymbol{x}_{t+1}^*$ by applying the sign gradient as
        $\boldsymbol{x}_{t+1}^* = \boldsymbol{x}_t^* + \alpha \cdot \text{sign}(\boldsymbol{g}_{t+1})$;   (7)
end for
return $\boldsymbol{x}^* = \boldsymbol{x}_T^*$.
```

3.1. Momentum iterative fast gradient sign method

The momentum method [18] is a technique for accelerating gradient descent algorithms by accumulating a velocity vector in the gradient direction of the loss function across iterations. The memorization of previous gradients helps to barrel through narrow valleys, small humps and poor local minima or maxima [4]. The momentum method also shows its effectiveness in stochastic gradient descent to stabilize the updates [20]. We apply the idea of momentum to generate adversarial examples and obtain tremendous benefits.

To generate a non-targeted adversarial example $\boldsymbol{x}^*$ from a real example $\boldsymbol{x}$, which satisfies the L_∞ norm bound, gradient-based approaches seek the adversarial example by solving the constrained optimization problem

$$\arg\max_{\boldsymbol{x}^*} J(\boldsymbol{x}^*, y), \quad \text{s.t. } \|\boldsymbol{x}^* - \boldsymbol{x}\|_\infty \le \epsilon, \tag{5}$$

where ϵ is the size of adversarial perturbation. FGSM generates an adversarial example by applying the sign of the gradient to a real example only once (in Eq. (1)) by the assumption of linearity of the decision boundary around the data point. However in practice, the linear assumption may not hold when the distortion is large [12], which makes the adversarial example generated by FGSM "underfits" the model, limiting its attack ability. In contrast, iterative FGSM greedily moves the adversarial example in the direction of the sign of the gradient in each iteration (in Eq. (3)). Therefore, the adversarial example can easily drop into poor local maxima and "overfit" the model, which is not likely to transfer across models.

In order to break such a dilemma, we integrate momentum into the iterative FGSM for the purpose of stabilizing update directions and escaping from poor local maxima. Therefore, the momentum-based method remains the transferability of adversarial examples when increasing iterations, and at the same time acts as a strong adversary for the white-box models like iterative FGSM. It alleviates the trade-off between the attack ability and the transferability, demonstrating strong black-box attacks.

The momentum iterative fast gradient sign method (MI-FGSM) is summarized in Algorithm 1. Specifically, $\boldsymbol{g}_t$ gathers the gradients of the first t iterations with a decay factor μ, defined in Eq. (6). Then the adversarial example $\boldsymbol{x}_t^*$ until the t-th iteration is perturbed in the direction of the sign of $\boldsymbol{g}_t$ with a step size α in Eq. (7). If μ equals to 0, MI-FGSM degenerates to the iterative FGSM. In each iteration, the current gradient $\nabla_{\boldsymbol{x}} J(\boldsymbol{x}_t^*, y)$ is normalized by the L_1 distance (any distance measure is feasible) of itself, because we notice that the scale of the gradients in different iterations varies in magnitude.

3.2. Attacking ensemble of models

In this section, we study how to attack an ensemble of models efficiently. Ensemble methods have been broadly adopted in researches and competitions for enhancing the performance and improving the robustness [6, 8, 2]. The idea of ensemble can also be applied to adversarial attacks, due to the fact that if an example remains adversarial for multiple models, it may capture an intrinsic direction that always fools these models and is more likely to transfer to other models at the same time [12], thus enabling powerful black-box attacks.

We propose to attack multiple models whose *logit* activations[1] are fused together, and we call this method *ensemble in logits*. Because the logits capture the logarithm relationships between the probability predictions, an ensemble of models fused by logits aggregates the fine detailed outputs of all models, whose vulnerability can be easily discovered. Specifically, to attack an ensemble of K models, we fuse the logits as

$$\boldsymbol{l}(\boldsymbol{x}) = \textstyle\sum_{k=1}^{K} w_k \boldsymbol{l}_k(\boldsymbol{x}), \tag{8}$$

where $\boldsymbol{l}_k(\boldsymbol{x})$ are the logits of the k-th model, w_k is the ensemble weight with $w_k \geq 0$ and $\sum_{k=1}^{K} w_k = 1$. The loss function $J(\boldsymbol{x}, y)$ is defined as the softmax cross-entropy loss given the ground-truth label y and the logits $\boldsymbol{l}(\boldsymbol{x})$

$$J(\boldsymbol{x}, y) = -\mathbf{1}_y \cdot \log(\text{softmax}(\boldsymbol{l}(\boldsymbol{x}))), \tag{9}$$

where $\mathbf{1}_y$ is the one-hot encoding of y. We summarize the MI-FGSM algorithm for attacking multiple models whose logits are averaged in Algorithm 2.

For comparison, we also introduce two alternative ensemble schemes, one of which is already studied [12]. Specifically, K models can be averaged in predictions [12] as $\boldsymbol{p}(\boldsymbol{x}) = \sum_{k=1}^{K} w_k \boldsymbol{p}_k(\boldsymbol{x})$, where $\boldsymbol{p}_k(\boldsymbol{x})$ is the predicted probability of the k-th model given input $\boldsymbol{x}$. K models can also be averaged in loss as $J(\boldsymbol{x}, y) = \sum_{k=1}^{K} w_k J_k(\boldsymbol{x}, y)$. In these three methods, the only difference is where to combine the outputs of multiple models, but they result in different attack abilities. We empirically find that the ensemble in logits performs better than the ensemble in predictions and the ensemble in loss, among various attack methods and various models in the ensemble, which will be demonstrated in Sec. 4.3.1.

Algorithm 2 MI-FGSM for an ensemble of models

```
Input: The logits of $K$ classifiers $\boldsymbol{l}_1, \boldsymbol{l}_2, ..., \boldsymbol{l}_K$; ensemble weights
    $w_1, w_2, ..., w_K$; a real example $\boldsymbol{x}$ and ground-truth label $y$;
Input: The size of perturbation $\epsilon$; iterations $T$ and decay factor $\mu$.
Output: An adversarial example $\boldsymbol{x}^*$ with $\|\boldsymbol{x}^* - \boldsymbol{x}\|_\infty \leq \epsilon$.
$\alpha = \epsilon/T$;
$\boldsymbol{g}_0 = 0$; $\boldsymbol{x}_0^* = \boldsymbol{x}$;
for $t = 0$ to $T-1$ do
    Input $\boldsymbol{x}_t^*$ and output $\boldsymbol{l}_k(\boldsymbol{x}_t^*)$ for $k = 1, 2, ..., K$;
    Fuse the logits as $\boldsymbol{l}(\boldsymbol{x}_t^*) = \sum_{k=1}^{K} w_k \boldsymbol{l}_k(\boldsymbol{x}_t^*)$;
    Get softmax cross-entropy loss $J(\boldsymbol{x}_t^*, y)$ based on $\boldsymbol{l}(\boldsymbol{x}_t^*)$
    and Eq. (9);
    Obtain the gradient $\nabla_{\boldsymbol{x}} J(\boldsymbol{x}_t^*, y)$;
    Update $\boldsymbol{g}_{t+1}$ by Eq. (6);
    Update $\boldsymbol{x}_{t+1}^*$ by Eq. (7);
end for
return $\boldsymbol{x}^* = \boldsymbol{x}_T^*$.
```

3.3. Extensions

The momentum iterative methods can be easily generalized to other attack settings. By replacing the current gradient with the accumulated gradient of all previous steps, any iterative method can be extended to its momentum variant. Here we introduce the methods for generating adversarial examples in terms of the L_2 norm bound attacks and the targeted attacks.

To find an adversarial examples within the ϵ vicinity of a real example measured by L_2 distance as $\|\boldsymbol{x}^* - \boldsymbol{x}\|_2 \leq \epsilon$, the momentum variant of iterative fast gradient method (MI-FGM) can be written as

$$\boldsymbol{x}_{t+1}^* = \boldsymbol{x}_t^* + \alpha \cdot \frac{\boldsymbol{g}_{t+1}}{\|\boldsymbol{g}_{t+1}\|_2}, \tag{10}$$

where $\boldsymbol{g}_{t+1}$ is defined in Eq. (6) and $\alpha = \epsilon/T$ with T standing for the total number of iterations.

For targeted attacks, the objective for finding an adversarial example misclassified as a target class y^* is to minimize the loss function $J(\boldsymbol{x}^*, y^*)$. The accumulated gradient is derived as

$$\boldsymbol{g}_{t+1} = \mu \cdot \boldsymbol{g}_t + \frac{J(\boldsymbol{x}_t^*, y^*)}{\|\nabla_{\boldsymbol{x}} J(\boldsymbol{x}_t^*, y^*)\|_1}. \tag{11}$$

The targeted MI-FGSM with an L_∞ norm bound is

$$\boldsymbol{x}_{t+1}^* = \boldsymbol{x}_t^* - \alpha \cdot \text{sign}(\boldsymbol{g}_{t+1}), \tag{12}$$

[1] Logits are the input values to softmax.

	Attack	Inc-v3	Inc-v4	IncRes-v2	Res-152	Inc-v3$_{\text{ens3}}$	Inc-v3$_{\text{ens4}}$	IncRes-v2$_{\text{ens}}$
Inc-v3	FGSM	72.3*	28.2	26.2	25.3	11.3	10.9	4.8
	I-FGSM	**100.0***	22.8	19.9	16.2	7.5	6.4	4.1
	MI-FGSM	**100.0***	**48.8**	**48.0**	**35.6**	**15.1**	**15.2**	**7.8**
Inc-v4	FGSM	32.7	61.0*	26.6	27.2	13.7	11.9	6.2
	I-FGSM	35.8	**99.9***	24.7	19.3	7.8	6.8	4.9
	MI-FGSM	**65.6**	**99.9***	**54.9**	**46.3**	**19.8**	**17.4**	**9.6**
IncRes-v2	FGSM	32.6	28.1	55.3*	25.8	13.1	12.1	7.5
	I-FGSM	37.8	20.8	**99.6***	22.8	8.9	7.8	5.8
	MI-FGSM	**69.8**	**62.1**	99.5*	**50.6**	**26.1**	**20.9**	**15.7**
Res-152	FGSM	35.0	28.2	27.5	72.9*	14.6	13.2	7.5
	I-FGSM	26.7	22.7	21.2	**98.6***	9.3	8.9	6.2
	MI-FGSM	**53.6**	**48.9**	**44.7**	98.5*	**22.1**	**21.7**	**12.9**

Table 1. The success rates (%) of non-targeted adversarial attacks against seven models we study. The adversarial examples are crafted for Inc-v3, Inc-v4, IncRes-v2 and Res-152 respectively using FGSM, I-FGSM and MI-FGSM. * indicates the white-box attacks.

and the targeted MI-FGM with an L_2 norm bound is

$$\boldsymbol{x}^*_{t+1} = \boldsymbol{x}^*_t - \alpha \cdot \frac{\boldsymbol{g}_{t+1}}{\|\boldsymbol{g}_{t+1}\|_2}. \qquad (13)$$

Therefore, we introduce a broad class of momentum iterative methods for attacks in various settings, whose effectiveness is demonstrated in Sec. 4.

4. Experiments

In this section, we conduct extensive experiments on the ImageNet dataset [19] to validate the effectiveness of the proposed methods. We first specify the experimental settings in Sec. 4.1. Then we report the results for attacking a single model in Sec. 4.2 and an ensemble of models in Sec. 4.3. Our methods won both the NIPS 2017 Non-targeted and Targeted Adversarial Attack competitions, with the configurations introduced in Sec. 4.4.

4.1. Setup

We study seven models, four of which are normally trained models—Inception v3 (Inc-v3) [22], Inception v4 (Inc-v4), Inception Resnet v2 (IncRes-v2) [21], Resnet v2-152 (Res-152) [7] and the other three of which are trained by ensemble adversarial training—Inc-v3$_{\text{ens3}}$, Inc-v3$_{\text{ens4}}$, IncRes-v2$_{\text{ens}}$ [24]. We will simply call the last three models as "adversarially trained models" without ambiguity.

It is less meaningful to study the success rates of attacks if the models cannot classify the original image correctly. Therefore, we randomly choose 1000 images belonging to the 1000 categories from the ILSVRC 2012 validation set, which are all correctly classified by them.

In our experiments, we compare our methods to one-step gradient-based methods and iterative methods. Since optimization-based methods cannot explicitly control the distance between the adversarial examples and the corresponding real examples, they are not directly comparable to ours, but they have similar properties with iterative methods as discussed in Sec. 2.1. For clarity, we only report the results based on L_∞ norm bound for non-targeted attacks, and leave the results based on L_2 norm bound and targeted attacks in the supplementary material. The findings in this paper are general across different attack settings.

4.2. Attacking a single model

We report in Table 1 the success rates of attacks against the models we consider. The adversarial examples are generated for Inc-v3, Inc-v4, InvRes-v2 and Res-152 respectively using FGSM, iterative FGSM (I-FGSM) and MI-FGSM attack methods. The success rates are the misclassification rates of the corresponding models with adversarial images as inputs. The maximum perturbation ϵ is set to 16 among all experiments, with pixel value in $[0, 255]$. The number of iterations is 10 for I-FGSM and MI-FGSM, and the decay factor μ is 1.0, which will be studied in Sec. 4.2.1.

From the table, we can observe that MI-FGSM remains as a strong white-box adversary like I-FGSM since it can attack a white-box model with a near 100% success rate. On the other hand, it can be seen that I-FGSM reduces the success rates for black-box attacks than one-step FGSM. But by integrating momentum, our MI-FGSM outperforms both FGSM and I-FGSM in black-box attacks significantly. It obtains more than 2 times of the success rates than I-FGSM in most black-box attack cases, demonstrating the effectiveness of the proposed algorithm. We show two adversarial images in Fig. 1 generated for Inc-v3.

It should be noted that although our method greatly improves the success rates for black-box attacks, it is still ineffective for attacking adversarially trained models (*e.g.*, less than 16% for IncRes-v2$_{\text{ens}}$) in the black-box manner. Later we show that ensemble-based approaches greatly improve the results in Sec. 4.3. Next, we study several aspects of MI-FGSM that are different from vanilla iterative methods, to further explain why it performs well in practice.

4.2.1 Decay factor μ

The decay factor μ plays a key role for improving the success rates of attacks. If $\mu = 0$, momentum-based iterative

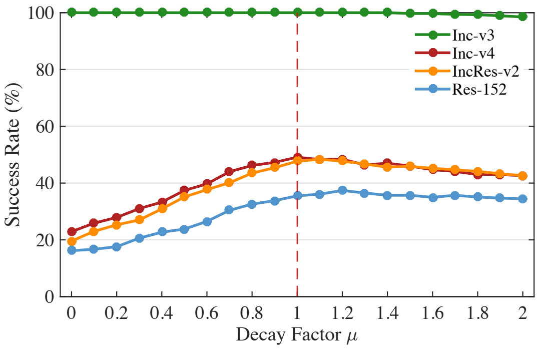

Figure 2. The success rates (%) of the adversarial examples generated for Inc-v3 against Inc-v3 (white-box), Inc-v4, IncRes-v2 and Res-152 (black-box), with μ ranging from 0.0 to 2.0.

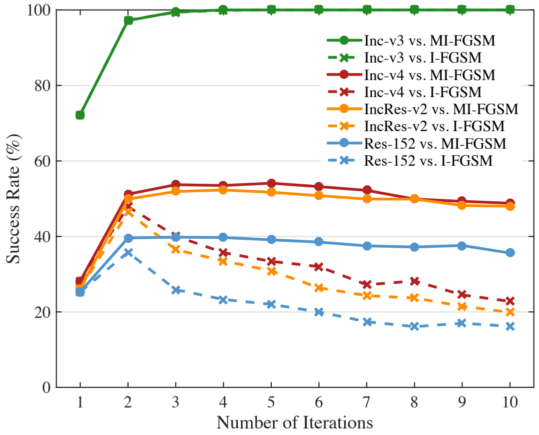

Figure 3. The success rates (%) of the adversarial examples generated for Inc-v3 model against Inc-v3 (white-box), Inc-v4, IncRes-v2 and Res-152 (black-box). We compare the results of I-FGSM and MI-FGSM with different iterations. Please note that the curves of Inc-v3 vs. MI-FGSM and Inc-v3 vs. I-FGSM overlap together.

methods trivially turn to vanilla iterative methods. Therefore, we study the appropriate value of the decay factor. We attack Inc-v3 model by MI-FGSM with the perturbation $\epsilon = 16$, the number of iterations 10, and the decay factor ranging from 0.0 to 2.0 with a granularity 0.1. We show the success rates of the generated adversarial examples against Inc-v3, Inc-v4, IncRes-v2 and Res-152 in Fig. 2. The curve of the success rate against a black-box model is unimodal whose maximum value is obtained at around $\mu = 1.0$. When $\mu = 1.0$, another interpretation of $\boldsymbol{g}_t$ defined in Eq. (6) is that it simply adds up all previous gradients to perform the current update.

4.2.2 The number of iterations

We then study the effect of the number of iterations on the success rates when using I-FGSM and MI-FGSM. We adopt the same hyper-parameters (*i.e.*, $\epsilon = 16$, $\mu = 1.0$) for attacking Inc-v3 model with the number of iterations ranging from 1 to 10, and then evaluate the success rates of adversarial examples against Inc-v3, Inc-v4, IncRes-v2 and Res-152 models, with the results shown in Fig. 3.

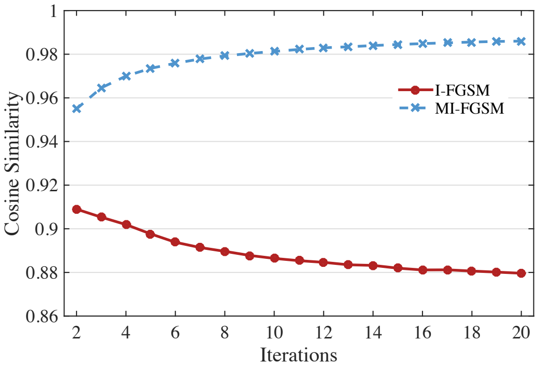

Figure 4. The cosine similarity of two successive perturbations in I-FGSM and MI-FGSM when attacking Inc-v3 model. The results are averaged over 1000 images.

It can be observed that when increasing the number of iterations, the success rate of I-FGSM against a black-box model gradually decreases, while that of MI-FGSM maintains at a high value. The results prove our argument that the adversarial examples generated by iterative methods easily overfit a white-box model and are not likely to transfer across models. But momentum-based iterative methods help to alleviate the trade-off between the white-box attacks and the transferability, thus demonstrating a strong attack ability for white-box and black-box models simultaneously.

4.2.3 Update directions

To interpret why MI-FGSM demonstrates better transferability, we further examine the update directions given by I-FGSM and MI-FGSM along the iterations. We calculate the cosine similarity of two successive perturbations and show the results in Fig. 4 when attacking Inc-v3. The update direction of MI-FGSM is more stable than that of I-FGSM due to the larger value of cosine similarity in MI-FGSM.

Recall that the transferability comes from the fact that models learn similar decision boundaries around a data point [12]. Although the decision boundaries are similar, they are unlikely the same due to the highly non-linear structure of DNNs. So there may exist some exceptional decision regions around a data point for a model (holes as shown in Fig. 4&5 in [12]), which are hard to transfer to other models. These regions correspond to poor local maxima in the optimization process and the iterative methods can easily trap into such regions, resulting in less transferable adversarial examples. On the other hand, the stabilized update directions obtained by the momentum methods as observed in Fig. 4 can help to escape from these exceptional regions, resulting in better transferability for adversarial attacks. Another interpretation is that the stabilized updated directions make the L_2 norm of the perturbations larger, which may be helpful for the transferability.

	Ensemble method	FGSM		I-FGSM		MI-FGSM	
		Ensemble	Hold-out	Ensemble	Hold-out	Ensemble	Hold-out
-Inc-v3	Logits	**55.7**	**45.7**	**99.7**	**72.1**	**99.6**	**87.9**
	Predictions	52.3	42.7	95.1	62.7	97.1	83.3
	Loss	50.5	42.2	93.8	63.1	97.0	81.9
-Inc-v4	Logits	**56.1**	**39.9**	**99.8**	**61.0**	**99.5**	**81.2**
	Predictions	50.9	36.5	95.5	52.4	97.1	77.4
	Loss	49.3	36.2	93.9	50.2	96.1	72.5
-IncRes-v2	Logits	**57.2**	**38.8**	**99.5**	**54.4**	**99.5**	**76.5**
	Predictions	52.1	35.8	97.1	46.9	98.0	73.9
	Loss	50.7	35.2	96.2	45.9	97.4	70.8
-Res-152	Logits	**53.5**	**35.9**	99.6	**43.5**	99.6	**69.6**
	Predictions	51.9	34.6	**99.9**	41.0	**99.8**	67.0
	Loss	50.4	34.1	98.2	40.1	98.8	65.2

Table 2. The success rates (%) of non-targeted adversarial attacks of three ensemble methods. We report the results for an ensemble of white-box models and a hold-out black-box target model. We study four models—Inc-v3, Inc-v4, IncRes-v2 and Res-152. In each row, "-" indicates the name of the hold-out model and the adversarial examples are generated for the ensemble of the other three models by FGSM, I-FGSM and MI-FGSM respectively. Ensemble in logits consistently outperform other methods.

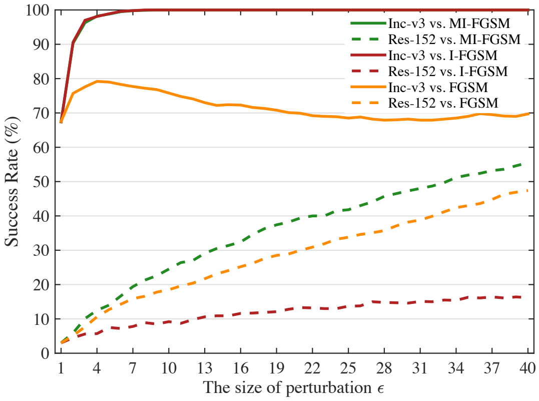

Figure 5. The success rates (%) of the adversarial examples generated for Inc-v3 against Inc-v3 (white-box) and Res-152 (black-box). We compare the results of FGSM, I-FGSM and MI-FGSM with different size of perturbation. The curves of Inc-v3 vs. MI-FGSM and Inc-v3 vs. I-FGSM overlap together.

4.2.4 The size of perturbation

We finally study the influence of the size of adversarial perturbation on the success rates. We attack Inc-v3 model by FGSM, I-FGSM and MI-FGSM with ϵ ranging from 1 to 40 with the image intensity $[0, 255]$, and evaluate the performance on the white-box model Inc-v3 and a black-box model Res-152. In our experiments, we set the step size α in I-FGSM and MI-FGSM to 1, so the number of iterations grows linearly with the size of perturbation ϵ. The results are shown in Fig. 5.

For the white-box attack, iterative methods reach the 100% success rate soon, but the success rate of one-step FGSM decreases when the perturbation is large. The phenomenon largely attributes to the inappropriate assumption of the linearity of the decision boundary when the perturbation is large [12]. For the black-box attacks, although the success rates of these three methods grow linearly with the size of perturbation, MI-FGSM's success rate grows faster. In other words, to attack a black-box model with a required success rate, MI-FGSM can use a smaller perturbation, which is more visually indistinguishable for humans.

4.3. Attacking an ensemble of models

In this section, we show the experimental results of attacking an ensemble of models. We first compare the three ensemble methods introduced in Sec. 3.2, and then demonstrate that the adversarially trained models are vulnerable to our black-box attacks.

4.3.1 Comparison of ensemble methods

We compare the ensemble methods for attacks in this section. We include four models in our study, which are Inc-v3, Inc-v4, IncRes-v2 and Res-152. In our experiments, we keep one model as the hold-out black-box model and attack an ensemble of the other three models by FGSM, I-FGSM and MI-FGSM respectively, to fully compare the results of the three ensemble methods, *i.e.*, ensemble in logits, ensemble in predictions and ensemble in loss. We set ϵ to 16, the number of iterations in I-FGSM and MI-FGSM to 10, μ in MI-FGSM to 1.0, and the ensemble weights equally. The results are shown in Table 2.

It can be observed that the ensemble in logits outperforms the ensemble in predictions and the ensemble in loss consistently among all the attack methods and different models in the ensemble for both the white-box and black-box attacks. Therefore, the ensemble in logits scheme is more suitable for adversarial attacks.

Another observation from Table 2 is that the adversarial examples generated by MI-FGSM transfer at a high rate, enabling strong black-box attacks. For example, by attacking an ensemble of Inc-v4, IncRes-v2 and Res-152 fused in logits without Inc-v3, the generated adversarial examples can fool Inc-v3 with a 87.9% success rate. Normally trained models show their great vulnerability against such an attack.

	Attack	Ensemble	Hold-out
-Inc-v3$_{\text{ens3}}$	FGSM	36.1	15.4
	I-FGSM	**99.6**	18.6
	MI-FGSM	**99.6**	**37.6**
-Inc-v3$_{\text{ens4}}$	FGSM	33.0	15.0
	I-FGSM	99.2	18.7
	MI-FGSM	**99.3**	**40.3**
-IncRes-v2$_{\text{ens}}$	FGSM	36.2	6.4
	I-FGSM	99.5	9.9
	MI-FGSM	**99.7**	**23.3**

Table 3. The success rates (%) of non-targeted adversarial attacks against an ensemble of white-box models and a hold-out black-box target model. We include seven models—Inc-v3, Inc-v4, IncRes-v2, Res-152, Inc-v3$_{\text{ens3}}$, Inc-v3$_{\text{ens4}}$ and IncRes-v2$_{\text{ens}}$. In each row, "-" indicates the name of the hold-out model and the adversarial examples are generated for the ensemble of the other six models.

4.3.2 Attacking adversarially trained models

To attack the adversarially trained models in the black-box manner, we include all seven models introduced in Sec. 4.1. Similarly, we keep one adversarially trained model as the hold-out target model to evaluate the performance in the black-box manner, and attack the rest six model in an ensemble, whose logits are fused together with equal ensemble weights. The perturbation ϵ is 16 and the decay factor μ is 1.0. We compare the results of FGSM, I-FGSM and MI-FGSM with 20 iterations. The results are shown in Table 3.

It can be seen that the adversarially trained models also cannot defend our attacks effectively, which can fool Inc-v3$_{\text{ens4}}$ by more than 40% of the adversarial examples. Therefore, the models obtained by ensemble adversarial training, the most robust models trained on the ImageNet as far as we are concerned, are vulnerable to our attacks in the black-box manner, thus causing new security issues for developing algorithms to learn robust deep learning models.

4.4. Competitions

There are three sub-competitions in the NIPS 2017 Adversarial Attacks and Defenses Competition, which are the Non-targeted Adversarial Attack, Targeted Adversarial Attack and Defense Against Adversarial Attack. The organizers provide 5000 ImageNet-compatible images for evaluating the attack and defense submissions. For each attack, one adversarial example is generated for each image with the size of perturbation ranging from 4 to 16 (specified by the organizers), and all adversarial examples run through all defense submissions to get the final score. We won the first places in both the non-targeted attack and targeted attack by the method introduced in this paper. We will specify the configurations in our submissions.

For the non-targeted attack[2], we implement the MI-FGSM for attacking an ensemble of Inc-v3, Inc-v4, IncRes-v2, Res-152, Inc-v3$_{\text{ens3}}$, Inc-v3$_{\text{ens4}}$, IncRes-v2$_{\text{ens}}$ and Inc-v3$_{\text{adv}}$ [10]. We adopt the ensemble in logits scheme. The ensemble weights are set as $^{1}/_{7.25}$ equally for the first seven models and $^{0.25}/_{7.25}$ for Inc-v3$_{\text{adv}}$. The number of iterations is 10 and the decay factor μ is 1.0.

For the targeted attack[3], we build two graphs for attacks. If the size of perturbation is smaller than 8, we attack Inc-v3 and IncRes-v2$_{\text{ens}}$ with ensemble weights $^{1}/_{3}$ and $^{2}/_{3}$; otherwise we attack an ensemble of Inc-v3, Inc-v3$_{\text{ens3}}$, Inc-v3$_{\text{ens4}}$, IncRes-v2$_{\text{ens}}$ and Inc-v3$_{\text{adv}}$ with ensemble weights $^{4}/_{11}$, $^{1}/_{11}$, $^{1}/_{11}$, $^{4}/_{11}$ and $^{1}/_{11}$. The number of iterations is 40 and 20 respectively, and the decay factor μ is also 1.0.

5. Discussion

Taking a different perspective, we think that finding an adversarial example is an analogue to training a model and the transferability of the adversarial example is also an analogue to the generalizability of the model. By taking a meta view, we actually "train" an adversarial example given a set of models as training data. In this way, the improved transferability obtained by the momentum and ensemble methods is reasonable because the generalizability of a model is usually improved by adopting the momentum optimizer or training on more data. And we think that other tricks (*e.g.*, SGD) for enhancing the generalizability of a model could also be incorporated into adversarial attacks for better transferability.

6. Conclusion

In this paper, we propose a broad class of momentum-based iterative methods to boost adversarial attacks, which can effectively fool white-box models as well as black-box models. Our methods consistently outperform one-step gradient-based methods and vanilla iterative methods in the black-box manner. We conduct extensive experiments to validate the effectiveness of the proposed methods and explain why they work in practice. To further improve the transferability of the generated adversarial examples, we propose to attack an ensemble of models whose logits are fused together. We show that the models obtained by ensemble adversarial training are vulnerable to our black-box attacks, which raises new security issues for the development of more robust deep learning models.

Acknowledgements

The work is supported by the National NSF of China (Nos. 61620106010, 61621136008, 61332007, 61571261 and U1611461), Beijing Natural Science Foundation (No. L172037), Tsinghua Tiangong Institute for Intelligent Computing and the NVIDIA NVAIL Program, and partially funded by Microsoft Research Asia and Tsinghua-Intel Joint Research Institute.

[2] Source code is available at https://github.com/dongyp13/Non-Targeted-Adversarial-Attacks.

[3] Source code is available at https://github.com/dongyp13/Targeted-Adversarial-Attacks.

References

[1] N. Carlini and D. Wagner. Towards evaluating the robustness of neural networks. In *IEEE Symposium on Security and Privacy*, 2017. 3

[2] R. Caruana, A. Niculescu-Mizil, G. Crew, and A. Ksikes. Ensemble selection from libraries of models. In *ICML*, 2004. 4

[3] Y. Dong, H. Su, J. Zhu, and F. Bao. Towards interpretable deep neural networks by leveraging adversarial examples. *arXiv preprint arXiv:1708.05493*, 2017. 3

[4] W. Duch and J. Korczak. Optimization and global minimization methods suitable for neural networks. *Neural computing surveys*, 2:163–212, 1998. 3

[5] I. J. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples. In *ICLR*, 2015. 1, 2, 3

[6] L. K. Hansen and P. Salamon. Neural network ensembles. *IEEE transactions on pattern analysis and machine intelligence*, 12(10):993–1001, 1990. 4

[7] K. He, X. Zhang, S. Ren, and J. Sun. Identity mappings in deep residual networks. In *ECCV*, 2016. 5

[8] A. Krogh and J. Vedelsby. Neural network ensembles, cross validation and active learning. In *NIPS*, 1994. 4

[9] A. Kurakin, I. Goodfellow, and S. Bengio. Adversarial examples in the physical world. *arXiv preprint arXiv:1607.02533*, 2016. 1, 2, 3

[10] A. Kurakin, I. Goodfellow, and S. Bengio. Adversarial machine learning at scale. In *ICLR*, 2017. 1, 2, 3, 8

[11] Y. Li and Y. Gal. Dropout inference in bayesian neural networks with alpha-divergences. In *ICML*, 2017. 3

[12] Y. Liu, X. Chen, C. Liu, and D. Song. Delving into transferable adversarial examples and black-box attacks. In *ICLR*, 2017. 1, 2, 3, 4, 6, 7

[13] J. H. Metzen, T. Genewein, V. Fischer, and B. Bischoff. On detecting adversarial perturbations. In *ICLR*, 2017. 3

[14] S. M. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, and P. Frossard. Universal adversarial perturbations. In *CVPR*, 2017. 1

[15] T. Pang, C. Du, and J. Zhu. Robust deep learning via reverse cross-entropy training and thresholding test. *arXiv preprint arXiv:1706.00633*, 2017. 3

[16] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik, and A. Swami. Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security*, 2017. 2

[17] N. Papernot, P. McDaniel, X. Wu, S. Jha, and A. Swami. Distillation as a defense to adversarial perturbations against deep neural networks. In *IEEE Symposium on Security and Privacy*, 2016. 3

[18] B. T. Polyak. Some methods of speeding up the convergence of iteration methods. *USSR Computational Mathematics and Mathematical Physics*, 4(5):1–17, 1964. 3

[19] 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):211–252, 2015. 2, 5

[20] I. Sutskever, J. Martens, G. Dahl, and G. Hinton. On the importance of initialization and momentum in deep learning. In *ICML*, 2013. 3

[21] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *AAAI*, 2017. 5

[22] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. In *CVPR*, 2016. 1, 5

[23] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus. Intriguing properties of neural networks. In *ICLR*, 2014. 1, 3

[24] F. Tramèr, A. Kurakin, N. Papernot, D. Boneh, and P. McDaniel. Ensemble adversarial training: Attacks and defenses. In *ICLR*, 2018. 2, 3, 5