Improving Adversarial Attacks with Ensemble-Based Approaches

Yapeng Ji[1,2] and Guoxu Zhou[1,3(✉)]

[1] School of Automation, Guangdong University of Technology, Guangzhou 510006, People's Republic of China
2112004090@mail2.gdut.edu.cn, gx.zhou@gdut.edu.cn
[2] Guangdong Key Laboratory of IoT information Technology, Guangzhou 510005, People's Republic of China
[3] Guangdong-HongKong-Macao Joint Laboratory for Smart Discrete Manufacturing, Guangzhou 510006, People's Republic of China

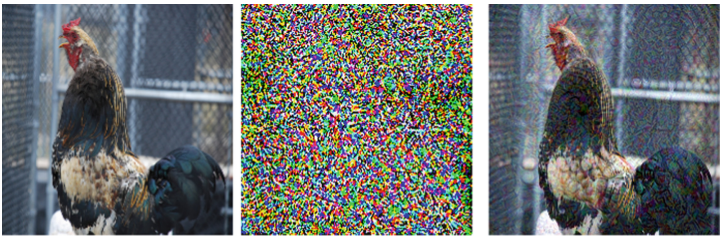
Abstract. Though Deep Neural networks (DNNs) have been applied in solving a wide variety of problems and achieved state-of-the-art performance on various vision tasks, they are vulnerable to adversarial examples which are crafted by adding human-imperceptible perturbations to legitimate inputs. However, most of the existing adversarial attacks have a low success rate under the black-box setting, where the attackers have no information about the model structure and parameters. In particular, targeted adversarial images, which are expected to predict a particular incorrect label, can hardly succeed. To address this, we propose a broad ensemble-based approach to improve the black-box attack. This method aims to find the common properties between all ensemble models. Using it in combination with Nesterov Accelerated Gradient, adversarial examples with higher transferability can be produced by a set of known models, meanwhile, keeping a higher success rate on all original models. In addition, the experiment result illustrates that, for more challenging targeted attacks, our methods exhibit higher transferability than other state-of-the-art attacks.

Keywords: Adversarial examples · Deep neural networks · Black-box attack

1 Introduction

Deep Neural networks have achieved an excellent performance on various computer vision tasks. However, in recent years the vulnerability of those models was discovered [1,2]. To be specific, DNNs will get a wrong result when the clean inputs add some imperceptible, human-imperceptible noises. In addition, adversarial examples show an intriguing transferability [1,3], where they crafted from one model can also fool other models. As a result of adversarial examples can not only evaluate the robustness of networks, but also improve the robustness of networks by adversarial training [4,9]. How to improve the transferability of adversarial examples has attracted a lot of attention (Fig. 1).

L. Fang et al. (Eds.): CICAI 2022, LNAI 13605, pp. 15–29, 2022.
https://doi.org/10.1007/978-3-031-20500-2_2



(a) cock: 95.40% (b) (c) armadillo: 97.05%

Fig. 1. Visualization of one targeted adversarial example generated by the proposed method (EI-NI-FGSM-HAG). A original images (a) is recognized correctly as a "cock" by ResNet-50, "armadillo" is randomly selected from other wrong labels. The adversarial noises (b) crafted by ResNet-152, VGG-19, Densenet-121, Inception-v3. The adversarial images (c) is the addition of the original images (a) and the noise (b), which is recognized as a "armadillo" by ResNet-50.

With the knowledge of the network, several methods have been proposed to generate adversarial examples. Specifically, containing Optimization-based methods like box-constrained L-BFGS [1], Carlini & Wagner attack (C&W) [5], gradient-based methods like fast gradient sign (FGSM) [2] and basic iterative method (I-FGSM) [7]. Those white-box attack methods can achieve high success rates. For black-box attacks, two different kinds of approaches have been proposed to implement it: One is the query-based approach [8], which trains a surrogate model by querying the unknown model. The surrogate model has similar prediction to unknown model, then we can use white-box attack methods to generate adversarial examples. However, in practical applications, it requires a large number of queries when the unknown network is complicated, that is easily detected by the model's defense system. The other is transfer-based approach, Yinpeng Dong et al. [10] utilize white-box attack methods to attack an ensemble of multiple models to generate adversarial images with high transferability. There is no need to query unknown networks. The aim of ensemble-based approaches is to attack their common vulnerability. However, they show low efficacy for targeted attack which requires adversarial examples to be classified by a network as a targeted label [3].

In this work, we improved the transferability of the adversarial image based on the ensemble model in three aspects: ensemble schemes, gradient descent mechanisms, and optimization methods.

- We discovered that different ensemble schemes have different effects on transferability of non-targeted attacks and targeted attacks. To specific, on untarget attacks, ensemble in softmax scheme has a higher success rate than the other two schemes, and the scheme of ensemble in loss is better on targeted attacks.
- In addition, we studied different gradient descent mechanisms. The results show that pixels with higher absolute gradient values are better represented

common properties between models. By attacking common properties between models to improve the transferability of adversarial examples.
- We integrate the Nesterov Iterative Fast Gradient Sign Method (NI-FGSM) [11] to the ensemble of models to avoid falling into local optimum in the optimization process. This method has been verified to be better than Momentum Iterative Fast Gradient Sign (MI-FGSM) [10] on a single model.

Extensive experiments on the ImageNet dataset [6] demonstrate that, on black-box setting, the proposed attack methods assist to improve the success rates of both non-targeted attacks and targeted attacks on a large margin. In targeted attacks, our best attack reaches the highest success rate of 30.1% on top-5 accuracy. This makes targeted attacks possible for black-box systems.

2 Related Works

In this section, we will give a brief introduction to some related works on adversarial attack. Let $\boldsymbol{x}$ and $\boldsymbol{x}^{adv}$ be a benign input and an adversarial input, respectively. Given a classifier $f_\theta(\boldsymbol{x})$, with ground truth label y, the goal of the non-targeted attack is searching for an adversarial image $\boldsymbol{x}^{adv}$ which is predicted by classifier satisfy $f_\theta(\boldsymbol{x}^{adv}) \neq y$. In targeted attack, the attacker aims to search for an adversarial image misclassified into a certain class y^{target}, that is $f_\theta(\boldsymbol{x}^{adv}) = y^{target}$. To limit the distortion, the adversarial images generated by both two kinds of method should satisfy $||\boldsymbol{x}^{adv} - \boldsymbol{x}||_p \leq \varepsilon$, where p could be $0, 1, 2, \infty$ and ε is the maximum value of distortion.

2.1 Optimization-Based Methods

One is directly optimizing the distortion between the benign images and the adversarial images [2,5]. To be specific, for non-targeted attacking, search for an adversarial example $\boldsymbol{x}^{adv}$ by solving:

$$\arg\min_{\boldsymbol{x}^{adv}} ||\boldsymbol{x}^{adv} - \boldsymbol{x}||_p - c \cdot J(\boldsymbol{x}^{adv}, y^{true}) \tag{1}$$

where $J(\boldsymbol{x}^{adv}, y^{true})$ is the loss function of prediction y^{true} and c is a constant to balance constraints the loss and distortion. Though, it is effective to find adversarial images, it is difficult to ensure the distortion between $\boldsymbol{x}^{adv}$ and $\boldsymbol{x}$ is less than ε.

2.2 Gradient-Based Methods

Fast Gradient Sign Method (FGSM): FGSM [2] find an adversarial image $\boldsymbol{x}^{adv}$ by the following equation:

$$\boldsymbol{x}^{adv} = \boldsymbol{x} + \varepsilon \cdot \text{sign}(\nabla_x J(\boldsymbol{x}, y^{true})) \tag{2}$$

This method just needs a one-step update and ε limits the maximum distortion.

Iterative Fast Gradient Sign Method (I-FGSM): I-FGSM [7] is an iterative version of FGSM. The iteration step length is $\alpha = \varepsilon/T$, where T is the number of Iterations.It can be expressed as:

$$\boldsymbol{x}_0 = \boldsymbol{x}, \boldsymbol{x}_{t+1}^{adv} = \boldsymbol{x}_t^{adv} + \alpha \cdot \text{sign}(\nabla_x J(\boldsymbol{x}_t^{adv}, y^{true})) \tag{3}$$

The performance of iterative methods is greatly greater than one-step methods in white-box setting. However, the transferability of adversarial examples is worse.

Momentum Iterative Fast Gradient Sign Method (MI-FGSM) [10]**:** In the optimization process, the momentum [12] is integrated into each iteration, improving the transferability of adversarial images. The broad formalization of this method is as follows:

$$\boldsymbol{g}_{t+1} = \mu \cdot \boldsymbol{g}_t + \frac{\nabla_{\boldsymbol{x}} J(\boldsymbol{x}_t^{adv}, y^{true})}{||\nabla_{\boldsymbol{x}} J(\boldsymbol{x}_t^{adv}, y^{true})||_1} \tag{4}$$

$$\boldsymbol{x}_{t+1}^{adv} = \boldsymbol{x}_t^{adv} + \alpha \cdot \text{sign}(\boldsymbol{g}_{t+1}) \tag{5}$$

μ is the decay factor, and the accumulated gradient is g_t. g_t is starting with $g_0 = 0$.

Nesterov Iterative Fast Gradient Sign Method (NI-FGSM): NI-FGSM [11] considers previous accumulated gradient as a correction to avoid trapping in local optimum. Similar to MI-FGSM, $\boldsymbol{g}_t$ is starting with $\boldsymbol{g_0 = 0}$. The update procedure is carried out as follows:

$$\boldsymbol{x}_t^{nes} = \boldsymbol{x}_t^{adv} + \alpha \cdot \mu \cdot \boldsymbol{g}_t \tag{6}$$

$$\boldsymbol{g}_{t+1} = \mu \cdot \boldsymbol{g}_t + \frac{\nabla_{\boldsymbol{x}} J(x_t^{nes}, y^{true})}{||\nabla_{\boldsymbol{x}} J(\boldsymbol{x}_t^{nes}, y^{true})||_1} \tag{7}$$

$$\boldsymbol{x}_{t+1}^{adv} = \boldsymbol{x}_t^{adv} + \alpha \cdot \text{sign}(\boldsymbol{g}_{t+1}) \tag{8}$$

g_t denotes accumulated gradients [13] at the iteration t, μ denotes the decay factor.

In this paper, the distortion between $\boldsymbol{x}^{adv}$ and $\boldsymbol{x}$ measure by root mean square deviation, i.e., RMSD, Which is calculated as $d(\boldsymbol{x}^{adv}, \boldsymbol{x}) = \sqrt{\sum_i (\boldsymbol{x}_i^{adv} - \boldsymbol{x}_i)^2 / N}$, Where $\boldsymbol{x}_i$ and N represent the dimensionality of $\boldsymbol{x}$ and the pixel value of the i-th dimension of $\boldsymbol{x}$, respectively. The values for each pixel range from 0 to 255.

2.3 Targeted Attacks

The method of generating the target adversarial example is similar to the non-target adversarial example, but the goals of the attackers transform to searching for an instance $\boldsymbol{x_{adv}}$ to satisfy $f_\theta(\boldsymbol{x}^{adv}) = y^{target}$. For the optimization-based methods, we have the following approximate solution to this problem.

$$\arg\min_{\boldsymbol{x}^{adv}} ||\boldsymbol{x}^{adv} - \boldsymbol{x}||_p + c \cdot J(\boldsymbol{x}^{adv}, y^{target}) \tag{9}$$

For I-FGSM, MI-FGSM and NI-FGSM, we make the following changes:

$$\boldsymbol{x}_{t+1}^{adv} = \boldsymbol{x}_t^{adv} + \alpha \cdot \text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{x}_t^{adv}, y^{target}) \text{(I-FGSM)}$$

$$\begin{aligned} \boldsymbol{g}_{t+1} &= \mu \cdot \boldsymbol{g}_t + \frac{\nabla_{\boldsymbol{x}} J(\boldsymbol{x}_t^{adv}, y^{target})}{||\nabla_{\boldsymbol{x}} J(\boldsymbol{x}_t^{adv}, y^{target})||_1} \\ \boldsymbol{x}_{t+1}^{adv} &= \boldsymbol{x}_t^{adv} - \alpha \cdot \text{sign}(\boldsymbol{g}_{t+1}) \end{aligned} \quad \text{(MI-FGSM)}$$

$$\begin{aligned} \boldsymbol{x}_t^{nes} &= \boldsymbol{x}_t^{adv} - \alpha \cdot \mu \cdot \boldsymbol{g}_t \\ \boldsymbol{g}_{t+1} &= \mu \cdot \boldsymbol{g}_t + \frac{\nabla_{\boldsymbol{x}} J(\boldsymbol{x}_t^{nes}, y^{target})}{||\nabla_{\boldsymbol{x}} J(\boldsymbol{x}_t^{nes}, y^{target})||_1} \\ \boldsymbol{x}_{t+1}^{adv} &= \boldsymbol{x}_t^{adv} - \alpha \cdot \text{sign}(\boldsymbol{g}_{t+1}) \end{aligned} \quad \text{(NI-FGSM)}$$

3 Methodology

3.1 Motivation

In the black box case, methods using only one known model to generate adversarial samples have been shown to be effective in non-targeted attacks [3,10]. However, for targeted attacks, the adversarial samples generated by a single known model are virtually untransferable. Attacking multiple models at the same time can be beneficial to improve transferability. Intuitively, if an adversarial example is misidentified by all known models, it is likely to be misidentified by other unknown models. For targeted attacks and non-targeted attacks, there are different ensemble schemes for us to consider. In addition, the process of generating adversarial examples can be seen as an optimization problem [11], so a better optimization algorithm can also improve the transferability of adversarial examples.

3.2 Ensemble Schemes

Let $\boldsymbol{l}_k(\boldsymbol{x})$ denote the logits of k-th model, and we have k known models, the softmax cross-entropy loss of k-th model can be expressed as:

$$J_k(\boldsymbol{x}, y) = -\mathbf{1}_y \cdot \log(\text{softmax}(w_k \boldsymbol{l}_k(\boldsymbol{x}))) \tag{10}$$

where 1_y is the one-hot encoding of ground-truth label y, w_k is the ensemble weight. We employ three ensemble schemes for targeted and non-targeted attacks: ensemble in logits (EI-logits), ensemble in softmax (EI-softmax), ensemble in loss (EI-loss). The ensemble loss of three ensemble schemes can be represented by the following three equations:

$$J(\boldsymbol{x}, y) = -\mathbf{1}_y \cdot \log(\text{softmax}(\textstyle\sum_{k=1}^{K} w_k \boldsymbol{l}_k(\boldsymbol{x}))), \tag{11}$$

$$J(\boldsymbol{x}, y) = -\mathbf{1}_y \cdot \log(\textstyle\sum_{k=1}^{K} \text{softmax}(w_k \boldsymbol{l}_k(\boldsymbol{x}))), \tag{12}$$

$$J(\boldsymbol{x}, y) = \textstyle\sum_{k=1}^{K} (-\mathbf{1}_y \cdot \log(\text{softmax}(w_k \boldsymbol{l}_k(\boldsymbol{x})))), \tag{13}$$

where $\sum_{k=1}^{K} w_k = 1$ and $w_k \geq 0$, and we have K known models. In all ensemble schemes, we set $w_1 = w_2 = ... = w_k$.

3.3 Gradient Descent Mechanisms

We discovered that some pixels did not update during the two iterations, which would affect the iteration direction of other pixels and ultimately affect the transferable of the adversarial samples. Moreover, we found that most unchanged pixels after two iterations have small absolute gradient values and those pixels with high absolute gradient values are more stable in the direction of iteration. In fact, pixels with a higher absolute gradient value have a greater impact on the loss in the white-box setting.

From the perspective of transferability, those pixels with a stable iteration direction are better represented common properties between models. Thus we can only change ones with high gradient absolute values during the iteration, so as to improve the transferability of adversarial examples under the premise of ensuring certain distortion. Based on the above analysis, we propose the higher absolute gradient method (HAG), which optimizes the adversarial perturbations over pixes with higher absolute gradient.

$$\boldsymbol{g}_{t+1}^{*}[i] = \begin{cases} \boldsymbol{g}_{t+1}[i], & \text{if} \quad i \in topk_{index} \\ 0, & \text{if} \quad i \notin topk_{index} \end{cases} \tag{14}$$

where i is the index of the corresponding element, $topk_{index}$ is computed by Eq. 15. The $topk(k, x)$ function returns the index of the first k percent of the largest elements of a given input tensor $\boldsymbol{x}$. For MI-FGSM and NI-FGSM, the updating formulas of non-target attacks and target attacks are Eq. 16 and Eq. 17 respectively.

$$topk_{index} = topk(k, |\boldsymbol{g}_{t+1}|) \tag{15}$$

$$\boldsymbol{x}_{t+1}^{adv} = \boldsymbol{x}_{t}^{adv} + \alpha \cdot sign(\boldsymbol{g}_{t+1}^{*}) \tag{16}$$

$$\boldsymbol{x}_{t+1}^{adv} = \boldsymbol{x}_{t}^{adv} - \alpha \cdot sign(\boldsymbol{g}_{t+1}^{*}) \tag{17}$$

3.4 Optimization Algorithm

Algorithm 1. EI-NI-FGSM-HAG

1: **Input:** A classifier f with softmax crossentopy loss function J; a real image $\boldsymbol{x}$ and ground-truth label y;
2: **Input:** The size of perturbation ε; iterations T and decay factor μ;
3: **Output:** An adversarial example $\boldsymbol{x}^{adv}$
4: Initialize $\alpha = \varepsilon/T; \boldsymbol{g}_0 = 0; \boldsymbol{x}_0^{adv} = \boldsymbol{x}$
5: **for** $t = 0$ **to** $T - 1$ **do**
6: Get $\boldsymbol{x}_t^{nes}$ by Eq. 6
7: Get $J(\boldsymbol{x}, y)$ by Eq. 11 or Eq. 12 or Eq. 13
8: Update $\boldsymbol{g}_{t+1}$ by Eq. 7
9: Get $\boldsymbol{g}_{t+1}^{*}$ by Eq. 14
10: Update $\boldsymbol{x}_{t+1}^{adv}$ by Eq. 16
11: **end for**
12: **return:** $\boldsymbol{x}^{adv} = \boldsymbol{x}_T^{adv}$

We can integrate ensemble schemes into gradient-based methods to generate adversarial examples with strong transferability. Adversarial examples crafted by one step attack method (FGSM) has higher transferability than Iterative attack methods in attacking single model. Nonetheless, when attacking ensemble models, one step method has a lower success rate on all original models so that it is failure to attack ensemble models' common vulnerability. In iterative attack methods, I-FGSM greedily searches for adversarial images in the direction of the sign of the gradient at each iteration, it easily falls into poor local optimum. MI-FGSM adopts momentum [12] which stabilizes the update direction and assists to escape from poor local optimum. NI-FGSM, more than stabilizing the update directions, gives previous accumulated gradient a correction to look ahead. Those properties are helpful to escape from poor local optimum and improve transferability of adverasial images. We merge the three ensemble schemes into NI-FGSM, where $J(\boldsymbol{x}, y)$ can be calculated from Eq. 11, Eq. 12 and Eq. 13. We summarize the NI-FGSM-HAG algorithm for attacking ensemble models in Algorithm 1.

4 Experimental Results

In this section we will present experimental results to demonstrate the effectiveness of the proposed methods. We first discuss the experimental settings and implementation details in Sect. 4.1. Then we report the results of non-targeted attacks and targeted attacks for attacking a single model in Sect. 4.2. We further conduct two trials to study the effects of our methods on attacking an ensemble model on non-targeted attacks and targeted attack in Sect. 4.3.

4.1 Experimental Settings

In this section, we detail the models to be examined, the dataset to be evaluated and the hyperparameters to be used.

Models. For normally trained models, we study five networks, ResNet-50 [17] , ResNet-152 [18],VGG-19 [15], Densenet-121 [16], Inception-v3 [14].

Dataset. We use a dataset which randomly extracted an image from each category of the ILSVRC 2012 validation set, 1000 images in total, and all of them can be classified correctly by all five models in our examination. For targeted attacks, we randomly select a lable from additional lables besides the correct one.

Hyper-Parameters. For the hyper-parameters, we set number of iteration $T = 10$, and step size $\alpha = 2$. For MI-FGSM and NI-FGSM, we adopt the default decay factor $\mu = 10$.

4.2 Attacking a Single Model

We first study the transferability of attacking a single model. Table 1 presents the success rates of non-targeted attacks and Table 2 show the top-1 success

Table 1. Attack success rates (%) of non-targeted adversarial images where we attack a single network. The adversarial examples are generated by Vgg-19, Dens-121, Res-142, Inc-v3 and Res-50 respectively using I-FGSM, MI-FGSM and NI-FGSM. * indicates the white-box attacks.

	Attack	Vgg-19	Dens-121	Res-152	Inc-v3	Res-50
Vgg-19	I-FGSM	**100.0***	43.4	28.3	25.1	41.3
	MI-FGSM	**100.0***	72.0	56.1	53.0	68.5
	NI-FGSM	**100.0***	**75.5**	**58.1**	**55.2**	**71.7**
Dens-121	I-FGSM	54.5	**100.0***	56.3	35.3	67.2
	MI-FGSM	78.2	**100.0***	74.3	60.0	84.0
	NI-FGSM	**82.4**	**100.0***	**80.5**	**64.4**	**88.0**
Res-152	I-FGSM	43.1	55.6	**100.0***	30.5	69.7
	MI-FGSM	65.5	78.2	**100.0***	55.0	85.3
	NI-FGSM	**71.0**	**82.4**	**100.0***	**57.1**	**89.5**
Inc-v3	I-FGSM	22.3	19.9	18.1	**98.7***	20.0
	MI-FGSM	48.8	42.9	36.6	98.2*	40.1
	NI-FGSM	**55.4**	**49.7**	**41.3**	98.4*	**47.1**
Res-50	I-FGSM	41.1	52.3	52.1	27.1	**99.8***
	MI-FGSM	65.3	72.1	68.3	46.2	99.7*
	NI-FGSM	**71.0**	**76.4**	**77.3**	**49.9**	**99.8***

Table 2. Attack success rates (%) of targeted adversarial images where we attack a single network. The adversarial examples are generated by Vgg-19, Dens-121, Res-142, Inc-v3 and Res-50 respectively using I-FGSM, MI-FGSM and NI-FGSM. * indicates the white-box attacks. Results of top-5 accuracy can be found in the Table 3)

	Attack	Vgg-19	Dens-121	Res-152	Inc-v3	Res-50
Vgg-19	I-FGSM	93.2*	0.30	0.10	0.10	0.30
	MI-FGSM	99.8*	**1.50**	**0.80**	**0.20**	0.70
	NI-FGSM	**100.**0*	1.10	0.50	**0.20**	**0.80**
Dens-121	I-FGSM	0.50	97.8*	0.90	0.10	1.50
	MI-FGSM	**1.90**	**100.0***	2.20	**0.60**	3.10
	NI-FGSM	1.60	**100.0***	**2.40**	**0.60**	**4.00**
Res-152	I-FGSM	0.30	1.60	97.5*	0.20	1.50
	MI-FGSM	**0.70**	3.60	**100.0***	**0.80**	3.40
	NI-FGSM	0.60	4.10	**100.0***	0.70	**5.20**
Inc-v3	I-FGSM	0.20	0.20	0.00	65.4*	0.20
	MI-FGSM	0.30	**0.50**	**0.60**	89.9*	**0.60**
	NI-FGSM	**0.70**	0.40	**0.60**	**91.9***	**0.60**
Res-50	I-FGSM	0.30	0.60	0.60	0.10	93.1*
	MI-FGSM	0.50	1.40	**1.90**	0.60	**100.0***
	NI-FGSM	**0.90**	**2.70**	1.70	**0.70**	**100.0***

Table 3. Top-5 accuracy of targeted adversarial images where we attack a single network. The adversarial examples are generated by Vgg-19, Dens-121, Res-142, Inc-v3 and Res-50 respectively using I-FGSM, MI-FGSM and NI-FGSM. * indicates the white-box attacks.

	Attack	Vgg-19	Dens-121	Res-152	Inc-v3	Res-50
Vgg-19	I-FGSM	98.0*	1.80	0.80	0.40	2.00
	MI-FGSM	**100.0***	**4.50**	**3.10**	**1.50**	2.70
	NI-FGSM	**100.0***	4.10	2.60	1.20	**3.60**
Dens-121	I-FGSM	2.30	98.9*	3.20	0.60	5.50
	MI-FGSM	3.90	**100.0***	**4.80**	2.20	8.10
	NI-FGSM	**4.60**	**100.0***	7.50	**2.80**	**10.5**
Res-152	I-FGSM	1.90	5.00	99.3*	0.90	6.70
	MI-FGSM	2.60	9.50	**100.0***	2.60	9.90
	NI-FGSM	**3.40**	**12.0**	**100.0***	**2.90**	**13.5**
Inc-v3	I-FGSM	0.80	1.60	1.20	80.2*	1.40
	MI-FGSM	0.90	**1.80**	**2.10**	96.5*	**2.10**
	NI-FGSM	**1.30**	**1.80**	1.90	**97.5***	1.80
Res-50	I-FGSM	1.10	3.60	3.10	0.70	97.3*
	MI-FGSM	**2.40**	5.70	5.00	1.50	100.0*
	NI-FGSM	2.30	**7.80**	**6.20**	**1.70**	**100.0***

rates of targeted attacks. For non-targeted attacks, the success rates are the misclassification rates against the models we consider. However, for targeted attacks, the success rates are the percentage of the adversarial examples crafted for one model that are classified as the target label by the corresponding model. The adversarial images are generated for Vgg-19, Dens-121, Res-152, Ince-v3 and Res-50 respectively. We use three iterative attack methods: I-FGSM, MI-FGSM, NI-FGSM to implement attack. The diagonal blocks represent white-box attack scenario and off-diagonal ones indicate black-box attack scenario. The models that we attack are arranged in rows, and that we test on in columns.

From the table, we can see that all three iterative attack methods can attack a white-box model with an almost 100% success rate for both non-targeted attacks and targeted attacks. As for the black-box scenario, it can be observed that NI-FGSM has a higher success rate than other iterative attack methods about 60% in non-targeted attacks, indicating the effectiveness of the optimization algorithm. But for target attack in black-box scenario, despite NI-FGSM and MI-FGSM increasing the success rates than I-FGSM, the success rates are still small, less than 1% in most cases, and only ten percent in the highest cases. We show top-5 success rates in Table 3. In the black-box scenario, targeted attacks are much harder than non-targeted attacks since the black-box model needs to classify adversarial images as specific error categories. We can do this by attacking attacking an ensemble of models. We'll cover that in the next section.

4.3 Attacking an Ensemble of Models

Based on the above analysis, we focus on generating more transferable adversarial examples via attacking an ensemble of models. In this section, we display the experimental results of non-targeted attacks in Sect. 4.3 and targeted attacks in Sect. 4.3.

Table 4. The success rates (%) of non-targeted adversarial images where we attack an ensemble networks. We study five models Vgg-19, Dens-121, Res-142, Inc-v3 and Res-50 and attack the ensemble networks by MI-FGSM. "*" indicates the black-box attacks. "-" indicates the name of the hold-out model and the adversarial examples are generated for the ensemble of the other four models by three ensemble schemes: EI-logits, EI-softmax and EI-loss.

	Schemes	RMSE	Vgg-19	Dens-121	Res-152	Inc-v3	Res-50
-Vgg-19	EI-logits	13.97	80.8*	98.3	98.3	98.4	98.4
	EI-softmax	14.23	**89.7***	**100.0**	**100.0**	**99.2**	**100.0**
	EI-loss	14.04	87.9*	98.9	97.1	88.6	98.1
-Dens-121	EI-logits	13.87	95.6	86.4*	96.6	97.6	97.1
	EI-softmax	14.21	**100.0**	**94.9***	**99.9**	**99.1**	**100.0**
	EI-loss	13.97	98.2	89.9*	95.8	85.5	96.3
-Res-152	EI-logits	13.90	96.3	97.1	83.3*	97.9	97.6
	EI-softmax	14.23	**100.0**	**100.0**	**90.9***	**99.2**	**100.0**
	EI-loss	13.99	98.0	99.0	87.8*	86.6	97.1
-Inc-v3	EI-logits	13.84	96.0	97.0	96.9	71.7*	97.5
	EI-softmax	14.17	**99.9**	**100.0**	**100.0**	**83.1***	**100.0**
	EI-loss	13.93	98.5	99.1	97.0	80.6*	98.2
-Res-50	EI-logits	13.98	96.1	97.0	97.2	98.0	87.9*
	EI-softmax	14.23	**100.0**	**100.0**	**100.0**	**99.1**	**95.6***
	EI-loss	14.04	98.0	98.8	96.2	88.7	92.6*

Table 5. The second line shows the percentage of pixels that did not change after two iterations and the third line shows the probability that the absolute gradient of invariant pixels will be in the last 50% after two iterations.

Iterations	2	3	4	5	6	7	8	9	10
Unchanged	25.40	20.57	17.36	15.15	13.21	12.41	12.20	10.80	10.03
Unchanged-Lower	70.98	77.10	82.97	85.87	87.35	88.16	84.06	88.36	90.87

Non-targeted Attack. We consider five models here, which are Vgg-19, Dens-121, Res-152, Inc-v3, Res-50. Adversarial images are crafted by an ensemble of four models, and tested on the another hold-out model. Firstly, we tested

the effects of different ensemble schemes on non-target attack. We compare the results of the three ensemble schemes, ensemble in logits, ensemble in softmax and ensemble in loss using the MI-FGSM attack method. The results are shown in Table 4. It can be found that the ensemble in softmax is better than the other two ensemble schemes for both the white-box and black-box attacks. For example, if adversarial examples are crafted on Vgg-19, Dense-121, Res-152, Inc-v3 has success rates of 95.6% on Res-50 and 100% on Vgg-19, while baselines like EI-logits only obtains the corresponding success rates of 87.9% and 96.1%, respectively.

In Table 5, we show the percentage of pixels whose pixel value has not altered after two iterations. We found that about 25% of pixes values are unchanged after the first two iterations, and most of them have a small absolute gradient values. Intuitively, pixels with a steady iteration direction are better represented common properties between models. So for Eq. 15, k is set to 0.5. As a result, only half of the pixes change in each iteration, which means that less perturbation is added to the adversarial examples. To compare transferability within the same disturbance range, we set the number of iterations to 13 when applying this method. We can combine HAG with EI-softmax naturally to form a much stronger non-targeted attack. We report the results in Table 6. MI-FGSM-HAG method improves the success rates on challenging black-box models and main-

Table 6. The success rates (%) of non-targeted adversarial images where we attack an ensemble networks. Using EI-softmax, we studied five models Vgg-19, Dens-121, Res-142, Inc-v3 and Res-50. "*" indicates the black-box attacks. '-"indicates the name of the hold-out model and the adversarial examples are generated for the ensemble of the other four models by MI-FGSM, MI-FGSM-HAG and NI-FGSM-HAG.

	Attacks	RMSE	Vgg-19	Dens-121	Res-152	Inc-v3	Res-50
-Vgg-19	MI-FGSM	14.23	89.7*	**100.0**	**100.0**	99.2	**100.0**
	MI-FGSM-HAG	13.53	90.6*	**100.0**	**100.0**	99.7	**100.0**
	NI-FGSM-HAG	13.69	**92.5***	**100.0**	**100.0**	**99.7**	**100.0**
-Dens-121	MI-FGSM	14.21	**100.0**	94.9*	99.9	99.1	**100.0**
	MI-FGSM-HAG	13.54	**100.0**	95.5*	**100.0**	99.5	**100.0**
	NI-FGSM-HAG	13.68	**100.0**	**96.4***	**100.0**	**99.9**	**100.0**
-Res-152	MI-FGSM	14.23	**100.0**	**100.0**	90.9*	99.2	**100.0**
	MI-FGSM-HAG	13.55	**100.0**	**100.0**	91.2*	99.4	**100.0**
	NI-FGSM-NAG	13.69	**100.0**	**100.0**	**93.5***	**99.8**	**100.0**
-Inc-v3	MI-FGSM	14.17	99.9	**100.0**	**100.0**	83.1*	**100.0**
	MI-FGSM-HAG	13.48	**100.0**	**100.0**	**100.0**	85.7*	**100.0**
	NI-FGSM-NAG	13.65	**100.0**	**100.0**	**100.0**	**87.6***	**100.0**
-Res-50	MI-FGSM	14.23	**100.0**	**100.0**	**100.0**	99.1	95.6*
	MI-FGSM-HAG	13.54	**100.0**	**100.0**	**100.0**	**99.8**	97.0*
	NI-FGSM-NAG	13.69	**100.0**	**100.0**	**100.0**	**99.8**	**97.3***

tains high success rates on white-box models. It should be noted that although we increase the number of iterations to 13 in MI-FGSM-HAG, the perturbation is still smaller than the MI-FGSM.

We then compare the success rates of NI-FGSM-HAG and MI-FGSM-HAG to see the effectiveness of optimization in Table 6. Experimental results show NI-FGSM-HAG is a stronger attack method than MI-FGSM-HAG. For the strongest attack method in the case of non-target attack NI-FGSM-HAG can fool white-box model at almost 100% and misclassify the balck-box model at almost 93% rate on average.

Targeted Attack. For more challenging targeted attack, we also examine the transferability of targeted adversarial images based on ensemble models. Table 7 presents the results for three esemble schemes using MI-FGSM methods. The results show EI-loss reaches much higher success rates than other two ensemble schemes on both black-box models and white-box models. Under the white-box setting, we see that EI-loss can reach more than 85% success rate. However, the highest success rate is only 11.7% under the black box setting.

Table 7. The success rates (%) of targeted adversarial images where we attack an ensemble networks. We study five models Vgg-19, Dens-121, Res-142, Inc-v3 and Res-50 and attack the ensemble networks by MI-FGSM. "*" indicates the black-box attacks. "-" indicates the hold-out model and the adversarial examples are generated for the ensemble of the other four models by three ensemble schemes: EI-logits, EI-softmax and EI-loss.

	Schemes	RMSE	Vgg-19	Dens-121	Res-152	Inc-v3	Res-50
-Vgg-19	EI-logits	14.44	0.9*	91.4	64.6	17.7	79.5
	EI-softmax	14.31	3.0*	10.3	80.1	31.5	80.8
	EI-loss	14.30	**3.5***	**100.0**	**99.8**	**86.1**	**100.0**
-Dens-121	EI-logits	14.45	49.4	3.0*	62.4	18.2	81.5
	EI-softmax	14.35	63.8	5.6*	3.8	37.0	75.3
	EI-loss	14.28	**98.2**	**11.0***	**99.6**	**86.5**	**100.0**
-Res-152	EI-logits	14.47	42.2	89.4	1.6*	16.3	75.8
	EI-softmax	14.35	63.9	5.6	3.9*	36.9	75.3
	EI-loss	14.31	**98.7**	**100.0**	**5.8***	**85.5**	**100.0**
-Inc-v3	EI-logits	14.51	39.5	92.6	61.4	1.1*	82.0
	EI-softmax	14.29	61.1	13.2	79.3	2.4*	81.0
	EI-loss	14.19	**99.3**	**100.0**	**99.9**	**3.7***	**100.0**
-Res-50	EI-logits	14.48	47.5	90.9	63.0	17.6	3.0 *
	EI-softmax	14.35	64.3	7.7	78.4	31.1	7.6*
	EI-loss	14.32	**98.5**	**100.0**	**99.6**	**86.3**	**11.7***

Table 8. The success rates (%) of targeted adversarial images where we attack an ensemble networks. Using EI-loss, we studied five models Vgg-19, Dens-121, Res-142, Inc-v3 and Res-50. "*" indicates the black-box attacks. "-"indicates the name of the hold-out model and the adversarial examples are generated for the ensemble of the other four models by MI-FGSM, MI-FGSM-HAG and NI-FGSM-HAG.

	Attacks	RMSE	Vgg-19	Dens-121	Res-152	Inc-v3	Res-50
-Vgg-19	MI-FGSM	14.30	3.5*	**100.0**	99.8	86.1	**100.0**
	MI-FGSM-HAG	13.58	4.1*	**100.0**	99.9	95.3	**100.0**
	NI-FGSM-HAG	13.71	**5.7***	**100.0**	**100.0**	**97.5**	**100.0**
-Dens-121	MI-FGSM	14.28	98.2	11.0*	99.6	86.5	**100.0**
	MI-FGSM-HAG	13.59	99.2	15.4*	**99.9**	95.0	**100.0**
	NI-FGSM-HAG	13.71	**99.5**	**16.0***	99.8	**97.9**	**100.0**
-Res-152	MI-FGSM	14.31	98.7	100.0	5.8*	**85.5**	**100.0**
	MI-FGSM-HAG	13.60	98.9	**100.0**	9.0*	95.0	**100.0**
	NI-FGSM-HAG	13.72	**99.7**	**100.0**	**11.5***	**97.8**	**100.0**
-Inc-v3	MI-FGSM	14.19	99.3	**100.0**	99.9	3.7*	**100.0**
	MI-FGSM-HAG	13.51	99.6	**100.0**	99.9	4.9*	**100.0**
	NI-FGSM-HAG	13.70	**100.0**	**100.0**	**100.0**	**6.6***	**100.0**
-Res-50	MI-FGSM	14.32	98.5	**100.0**	99.6	86.3	11.7*
	MI-FGSM-HAG	13.59	99.2	**100.0**	**99.9**	95.9	14.5*
	NI-FGSM-HAG	13.72	**99.8**	**100.0**	**99.9**	**98.0**	**17.3***

Table 9. Top-5 accuracy of targeted adversarial images where we attack an ensemble networks. Using EI-loss, we studied five models Vgg-19, Dens-121, Res-142, Inc-v3 and Res-50. "*" indicates the black-box attacks. "-" indicates the name of the hold-out model and the adversarial examples are generated for the ensemble of the other four models by MI-FGSM, MI-FGSM-HAG and NI-FGSM-HAG.

	Attacks	RMSE	Vgg-19	Dens-121	Res-152	Inc-v3	Res-50
-Vgg-19	MI-FGSM	14.30	8.9*	**100.0**	**100.0**	93.2	**100.0**
	MI-FGSM-HAG	13.58	9.8*	**100.0**	**100.0**	98.4	**100.0**
	NI-FGSM-HAG	13.71	**11.7***	**100.0**	**100.0**	**99.0**	**100.0**
-Dens-121	MI-FGSM	14.28	99.0	21.5*	99.8	93.6	**100.0**
	MI-FGSM-HAG	13.59	99.6	25.9*	**100.0**	98.7	**100.0**
	NI-FGSM-HAG	13.71	**99.7**	**29.2***	**100.0**	**99.7**	**100.0**
-Res-152	MI-FGSM	14.31	99.3	**100.0**	15.1*	93.7	**100.0**
	MI-FGSM-HAG	13.60	99.6	**100.0**	18.6*	98.7	**100.0**
	NI-FGSM-NAG	13.72	**99.9**	**100.0**	**22.0***	**98.9**	**100.0**
-Inc-v3	MI-FGSM	14.19	99.6	**100.0**	**100.0**	9.0*	**100.0**
	MI-FGSM-HAG	13.51	99.6	**100.0**	**100.0**	12.6*	**100.0**
	NI-FGSM-HAG	13.70	**100.0**	**100.0**	**100.0**	**13.4***	**100.0**
-Res-50	MI-FGSM	14.32	99.2	**100.0**	99.7	94.4	21.6*
	MI-FGSM-HAG	13.59	99.5	**100.0**	**100.0**	99.0	28.3*
	NI-FGSM-HAG	13.72	**99.9**	**100.0**	**100.0**	**99.5**	**30.1***

For gradient descent mechanism, we set k to 0.5 like untargeted attack. The results are summarized in Table 8. The HAG yields a maximum black-box success rate of 14.5% with lower distortion. We then conducted experiments to validate the effectiveness of the combination of NI-FGSM and HAG. As Table 8 suggests, NI-FGSM-HAG obtains a significant performance improvement. The best black-box success rate attained 17.3 %, and in white-box models, the lowest success rate reached 98.9%. We also examine targeted attacks based on top-5 accuracy, the highest success rate is 30% in balck-box setting. The results can be found in the Table 9. We found that targeted attacks also have almost the same success rate as non-target attacks in the white box setting, but a low success rate in the black-box models, which means that targeted adversarial examples have a much poor transferability.

5 Conclusion and Future Work

In this paper, we propose three methods to improve the transferability of adversarial examples based on the ensemble models. Specifically, we found that different ensemble schemes have different effects on non-targeted attacks and targeted attack, EI-softmax suitable for non-targeted attacks and EI-loss suitable for targeted attacks. Moreover, we discovered that pixels with higher absolute gradient values have better transferability. By integrating HAG with NI-FGSM, we can further improve the transferability of adversarial examples. We conduct extensive experiments to demonstrate that our methods not only yield higher success rates on untargeted attacks but also enhanced the success rates on more harder targeted attacks.

References

1. Szegedy, C., et al.: Intriguing properties of neural networks. arXiv (2013). https://arxiv.org/abs/1412.6572
2. Goodfellow, I.J., Shlens, J., Szegedy, C.: Intriguing properties of neural networks. arXiv (2014). https://arxiv.org/abs/1412.6572
3. Liu, Y., Chen, X., Liu, C., Song, D.: Delving into transferable adversarial examples and black-box attacks. arXiv (2016). https://arxiv.org/abs/1611.02770
4. Madry, A., Makelov, A., Schmidt, L., Tsipras, D., Vladu, A.: Towards deep learning models resistant to adversarial attacks. arXiv (2017). https://arxiv.org/abs/1706.06083
5. Carlini, N., Wagner, D.: Adversarial examples are not easily detected: bypassing ten detection methods, pp. 3–14. ACM (2017). https://dl.acm.org/doi/abs/10.1145/3128572.3140444
6. Russakovsky, O., et al.: ImageNet large scale visual recognition challenge. Int. J. Comput. Vision **115**(3), 211–252 (2015). https://doi.org/10.1007/s11263-015-0816-y
7. Kurakin, A., Goodfellow, I., Bengio, S., et al.: Adversarial examples in the physical world (2016). https://openreview.net/forum?id=HJGU3Rodl

8. Papernot, N., McDaniel, P., Goodfellow, I., Jha, S., Celik, Z.B., Swami, A.: Practical black-box attacks against machine learning, pp. 506–519 (2017). https://dl.acm.org/doi/abs/10.1145/3052973.3053009
9. Song, C., He, K., Wang, L., Hopcroft, J.E.: Improving the generalization of adversarial training with domain adaptation. arXiv (2018). https://arxiv.org/abs/1810.00740
10. Dong, Y., et al.: Boosting adversarial attacks with momentum. In: CVPR, pp. 9185–9193 (2018). https://openaccess.thecvf.com/content_cvpr_2018/html/Dong_Boosting_Adversarial_Attacks_CVPR_2018_paper.html
11. Lin, J., Song, C., He, K., Wang, L., Hopcroft, J.E.: Nesterov accelerated gradient and scale invariance for adversarial attacks. arXiv (2019). https://arxiv.org/abs/1810.00740
12. Polyak, B.T.: Some methods of speeding up the convergence of iteration methods. USSR Comput. Math. Math. Phys. **4**, 1–17 (1964). https://www.sciencedirect.com/science/article/abs/pii/0041555364901375
13. Nesterov, Y.: A method for unconstrained convex minimization problem with the rate of convergence O (1/k^ 2). Doklady USSR **269**, 543–547 (1983). https://cir.nii.ac.jp/crid/1570572699326076416
14. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: CVPR, pp. 2818–2826 (2016). https://www.cv-foundation.org/openaccess/content_cvpr_2016/html/Szegedy_Rethinking_the_Inception_CVPR_2016_paper.html
15. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv (2014). https://arxiv.org/abs/1409.1556
16. Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. In: CVPR, pp. 4700–4708 (2017). https://openaccess.thecvf.com/content_cvpr_2017/html/Huang_Densely_Connected_Convolutional_CVPR_2017_paper.html
17. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR, pp. 770–778 (2016). https://openaccess.thecvf.com/content_cvpr_2016/html/He_Deep_Residual_Learning_CVPR_2016_paper.html
18. He, K., Zhang, X., Ren, S., Sun, J.: Identity mappings in deep residual networks. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9908, pp. 630–645. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46493-0_38