Boosting Adversarial Attacks with Momentum

Yinpeng Dong¹, Fangzhou Liao¹, Tianyu Pang¹, Hang Su¹, Jun Zhu¹, Xiaolin Hu¹, Jianguo Li²
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 attacks, which 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 momentum-based iterative algorithm to boost adversarial attacks. By integrating the momentum term into the iterative process for attacks, our method 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 algorithm to attack an ensemble of models, and show that the adversarially trained models with a strong defense ability are also vulnerable to our black-box attacks.

1. Introduction

Deep neural networks (DNNs) are challenged by their vulnerability to adversarial examples [3, 1], 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 [1].

In this paper, we propose a momentum iterative gradient-based method to boost the success rates of the generated adversarial examples. Beyond basic iterative method [2] that iteratively perturbs the input with the gradients to maximize the loss function, momentum-based method accumulates 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 our method have higher success rates in both white-box and black-box attacks. The proposed method acts as a stronger attack algorithm than the fast gradient sign method (FGSM) [1] and iterative fast gradient sign method (I-FGSM) [2].

To further improve the transferability of adversarial examples, we propose to attack an ensemble of models whose logits are fused. We show that the adversarial examples generated by the momentum iterative method for multiple models, can successfully fool robust models obtained by ensemble adversarial training [4] 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 deep learning models and defense methods.
2. Methodology

In this section, we elaborate the proposed algorithm. To generate a non-targeted adversarial example \( x^* \) from a real example \( x \), which satisfies the \( L_\infty \) norm bound, gradient-based approaches seek the adversarial example by solving the constrained optimization problem

\[
\arg\max_{x^*} J(x^*, y), \quad \text{s.t.} \quad \|x^* - x\|_\infty \leq \epsilon,
\]

where \( \epsilon \) is the size of adversarial perturbation and \( J \) is often the cross-entropy loss function.

The momentum iterative fast gradient sign method (MI-FGSM) is summarized in Algorithm 1. Specifically, \( g_t \) gathers the gradients of the first \( t \) iterations with a decay factor \( \mu \), defined in Eq. (2). Then the adversarial example \( x^*_t \) until the \( t \)-th iteration is perturbed in the direction of the sign of \( g_t \) with a step size \( \alpha \) in Eq. (3). If \( \mu \) equals to 0, MI-FGSM degenerates to the I-FGSM. In each iteration, the current gradient \( \nabla_x J(x^*_t, y) \) 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.

To attack an ensemble of models, we propose to fuse multiple models in logits. (Logits are the input values to softmax.) Because the logits capture the logarithm relationship 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

\[
I(x) = \sum_{k=1}^{K} w_k l_k(x),
\]

where \( l_k(x) \) are the logits of the \( k \)-th model, \( w_k \) is the ensemble weight with \( w_k \geq 0 \) and \( \sum_{k=1}^{K} w_k = 1 \). We again use the MI-FGSM to attack the ensemble model.

Algorithm 1 MI-FGSM

Input: A classifier \( f \) with loss function \( J \); a real example \( x \) and ground-truth label \( y \);

Output: The size of perturbation \( \epsilon \); iterations \( \tau \) and decay factor \( \mu \).

1. \( \alpha = \epsilon / \tau \);
2. \( g_0 = 0; \quad x_0 = x \);
3. for \( t = 0 \) to \( T - 1 \) do
4. Input \( x^*_t \) to \( f \) and obtain the gradient \( \nabla_x J(x^*_t, y) \);
5. Update \( g_{t+1} \) by accumulating the velocity vector in the gradient direction as
   \[
   g_{t+1} = \mu \cdot g_t + \frac{\nabla_x J(x^*_t, y)}{\|\nabla_x J(x^*_t, y)\|_1};
   \]
6. Update \( x^*_{t+1} \) by applying the sign gradient as
   \[
   x^*_{t+1} = x^*_t + \alpha \cdot \text{sign}(g_{t+1});
   \]
7. end for
8. return \( x^* = x^*_T \).

3. Experiments

We report in Table 1 the success rates of attacks. From the table, we can observe that 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.

<table>
<thead>
<tr>
<th>Attack</th>
<th>Inc-v3</th>
<th>Inc-v4</th>
<th>IncRes-v2</th>
<th>Res-152</th>
</tr>
</thead>
<tbody>
<tr>
<td>FGSM</td>
<td>72.3%</td>
<td>28.2%</td>
<td>26.2%</td>
<td>25.3%</td>
</tr>
<tr>
<td>I-FGSM</td>
<td>100.0%</td>
<td>22.8%</td>
<td>19.9%</td>
<td>16.2%</td>
</tr>
<tr>
<td>MI-FGSM</td>
<td>100.0%</td>
<td>4.8%</td>
<td>4.8%</td>
<td>35.6%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Attack</th>
<th>Inc-v4</th>
<th>Inc-v3</th>
<th>IncRes-v2</th>
<th>Res-152</th>
</tr>
</thead>
<tbody>
<tr>
<td>FGSM</td>
<td>32.7%</td>
<td>61.0%</td>
<td>26.6%</td>
<td>27.2%</td>
</tr>
<tr>
<td>I-FGSM</td>
<td>35.8%</td>
<td>99.9%</td>
<td>24.7%</td>
<td>19.3%</td>
</tr>
<tr>
<td>MI-FGSM</td>
<td>65.6%</td>
<td>99.9%</td>
<td>54.9%</td>
<td>46.3%</td>
</tr>
</tbody>
</table>

Table 1. The success rates (%) of non-targeted adversarial attacks. 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.

In the second part of our experiments, we keep one adversarially trained model as the hold-out target model and attack another six models in an ensemble, whose logits are fused with equal ensemble weights. The results are shown in Table 2. 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.

<table>
<thead>
<tr>
<th>Attack</th>
<th>Ensemble</th>
<th>Hold-out</th>
</tr>
</thead>
<tbody>
<tr>
<td>-Inc-v3_\text{ens3}</td>
<td>FGSM</td>
<td>36.1%</td>
</tr>
<tr>
<td>I-FGSM</td>
<td>99.6%</td>
<td>18.6%</td>
</tr>
<tr>
<td>MI-FGSM</td>
<td>99.6%</td>
<td>37.6%</td>
</tr>
</tbody>
</table>

Table 2. The success rates (%) of adversarial attacks against an ensemble of white-box models and a hold-out black-box model.

References