Boosting Black-Box Adversarial Attacks with Meta Learning

Junjie Fu1,2, Jian Sun1,2, Gang Wang1,2
1. Key Lab of Intelligent Control and Decision of Complex Systems, Beijing Institute of Technology, Beijing 10081, China
2. Beijing Institute of Technology Chongqing Innovation Center, Chongqing 401120, China

Abstract: Deep neural networks (DNNs) have achieved remarkable success in diverse fields. However, it has been demonstrated that DNNs are very vulnerable to adversarial examples even in black-box settings. A large number of black-box attack methods have been proposed to in the literature. However, those methods usually suffer from low success rates and large query counts, which cannot fully satisfy practical purposes. In this paper, we propose a hybrid attack method which trains meta adversarial perturbations (MAPs) on surrogate models and performs black-box attacks by estimating gradients of the models. Our method uses the meta adversarial perturbation as an initialization and subsequently trains any black-box attack method for several epochs. Furthermore, the MAPs enjoy favorable transferability and universality, in the sense that they can be employed to boost performance of other black-box adversarial attack methods. Extensive experiments demonstrate that our method can not only improve the attack success rates, but also reduces the number of queries compared to other methods.

Key Words: Deep neural networks, adversarial examples, black-box attack, meta adversarial perturbation, transferability

1 Introduction

Deep neural networks (DNNs) have achieved significant success in a number of science and engineering fields in recent years, including image classification, object detection, smart cities, protein folding, and human-machine confrontation [2, 10, 11, 13–15, 17, 27]. Yet, while DNNs are widely used in diverse scenarios, serious security risks are revealed. Recently, researchers have shown that DNNs are vulnerable to adversarial examples which lead them to completely wrong classification or recognition results. Adversarial examples are crafted to be misclassified by well-trained classifiers but almost imperceptible to humans. It is very dangerous for DNN systems employed in security-critical applications such as self-driving [29], smart grid [26, 28], and financial recognition security system [23], which may lead to huge economic losses or even catastrophic accidents. After realizing this phenomenon, many efforts have been devoted to studying how to generate adversarial samples, which are critical for explaining and understanding adversarial attacks, evaluating the performance of DNNs, as well as improving the robustness of deep learning models.

In general, depending on whether the attacker has knowledge of the target deep model, adversarial attacks can be divided into white-box attacks and black-box attacks. In the white-box setting, the attacker has full access to the model, including the model architecture and model parameters. Leveraging these information, Goodfellow et al. early proposed the Fast Gradient Sign Method (FGSM) [8] and has aroused considerable attention. After that, a number of adversarial attack methods have been proposed, including DeepFool [19], BIM [12], JSMA [22], C&W [1], PGD [18], and AI-FGSM [30], to name a few. While there exist relatively many well-performing white-box attacks, the more challenging black-box attacks have not been sufficiently studied. In the black-box setting, the attacker cannot access the internal knowledge of the target model, and can only craft adversarial examples using the input-output information. The black-box setting is more practical and closer to real-world applications.

Lately, some progress has been made in black-box attacks. The work [21] proposed a black-box attack method based on training surrogate models. Chen et al. [3] performed black-box attacks by designing a zero-order optimization algorithm which estimate the gradient by querying the model’s output at certain input. Building on the two methods above, researchers have proposed some black-box attack methods with higher efficiency and success rates, such as NES [9], MI-FGSM [6], AutoZoom [25], P-RGF [5], Sign-OPT [16], etc. Although these existing methods have achieved improvement in black-box attacks, there are still drawbacks including e.g., low success rate and low query efficiency. A natural question arises: Can we generate a meta adversarial perturbation which can be used to adversarially manipulative different images while being efficiently trainable on small datasets?

Inspired by MAML [7], we find that meta learning can use little data to solve problems analogously. The purpose of meta learning is to train a model on multiple learning tasks, such that it can be quickly adapted to new learning tasks using only a small amount of training data. And MAML propose a model-agnostic algorithm for meta learning to train the model’s parameters such that it can achieve rapid adaptation. It is exactly what is needed in the black-box setting. Yuan et al. [31] trained a meta adversarial perturbation using MAML in the white-box setting. However, in a black-box model, an attacker cannot obtain the gradient of the model like a white-box model. Therefore it is difficult to simply generalize Yuan’s idea to the black-box model since the gradient cannot be obtained directly to train the perturbations by using MAML. New methods still need to be come up with.

Therefore, to improve the query efficiency and boost black-box attacks, in this paper, a new method is proposed for training meta adversarial perturbations and performing black-box attacks. In a nutshell, the main contributions of this paper are summarized as follows.

- A new algorithm is proposed to generate meta adversar-
ional perturbations in black-box setting, such that it can use less queries to achieve a higher success rate.

• A momentum term is added to the zero order optimization process to accelerate attack and reduce the number of queries.
• The meta adversarial perturbations are shown to have great adaptability and transferability, because the meta adversarial perturbations are trained by the data that is semantically unrelated to the attacked images, the perturbations can still achieve excellent performance.

2 Related Works

There is a great deal of work on adversarial examples. Here the most relevant works about white-box attacks and black-box attacks are reviewed.

2.1 White-box attacks

In white-box attacks, the model is completely known to the attacker, so the attacker can craft the adversarial examples by using internal information of the model directly. Szegedy et al. [24] first proposed a method for generating adversarial examples by solving the optimization problem with L-BFGS. Soon after, Goodfellow et al. [8] proposed a fast algorithm requiring only one step of gradient update and achieved good results. After that, a number of adversarial attack methods have been proposed, including DeepFool [19], BIM [12], JSMA [22], C&W [1], PGD [18], and Al-FGSM [30], etc. Recently, Yuan et al. [31] proposed a method for training universal adversarial perturbations in a white-box model based on meta learning, and achieve better results than the FGSM method. White-box attacks have already achieved high success rates and relatively high attack efficiencies. However, the black-box setting is more practical and closer to real-world applications than the white-box setting.

2.2 Black-box attacks

In the black-box setting, the attacker cannot access the internal knowledge of the target model, and can only craft adversarial examples using the output information. Due to the lack of knowledge of the model, black-box attacks are much more difficult than white-box attacks. There exist three main ideas in black-box attacks, which are transfer-based methods, query-based methods and hybrid methods. The three methods mentioned above are introduced.

• Transfer-based methods
Transfer-based methods perform black-box attacks based on the transferability of the adversarial examples. Transferability means that if an adversarial example can attack the current model, then it is possible to successfully attack other models as well. Papernot et al. [21] trained a surrogate model to perform black-box attacks with a synthetic dataset. When performing an attack, the attacker first generates an adversarial example on the surrogate model by using a white-box method, and then exploit the transferability of the adversarial example to perform the black-box attack on the target model. Later on, Dong et al. [6] proposed MI-FGSM by adding a momentum term to the iterative process, which had higher attack efficiency.

Due to the absence of estimated black-box model information, transfer-based methods have low success rates and large distortions. But they have high efficiency since they don’t require lots of queries on the black-box model.

• Query-based methods
Query-based methods perform black-box attacks based on estimating gradient of models by continuously querying the models. Chen et al. [3] proposed an zero-order optimization algorithm named ZOO to estimate gradient of models by performing finite difference on the query results. And then, Ilyas et al. proposed NES [9] to perform attack in a query limit condition, which improved query efficiency a lot and achieved higher success rate with lower queries. Tu et al. [25] used encoding-decoding method to reduce the query dimension of input and performed zero-order optimization with RGF [20] method.

Query-based methods require a large number of queries on the model, so they’re very inefficient. But they have much higher success rates and lower distortions than transfer-based methods, since they access the gradient by a direct query to the target model.

• Hybrid methods
Building on the two methods above, hybrid methods integrate the advantages of the two methods, which use surrogate model to acquire transfer-based prior knowledge and zero-order optimization to estimate information of target model more accurately. So hybrid methods also have higher success rates and fewer queries. Dong et al. [5] proposed a hybrid method named P-RGF, which used the gradient of surrogate model as prior knowledge to guide the query direction of RGF and obtained the same success rate as RGF with fewer queries.

2.3 Non-targeted and targeted attacks

According to the purpose of attack, adversarial attacks can be divided into non-targeted attacks and targeted attacks. Non-targeted attacks require that the model’s prediction of input image is different from its ground truth. While targeted attacks not only need to meet the above requirements, but also require the image to be classified into the specified target class. Obviously, targeted attacks are more difficult and more threatening. In the existing research on black-box attacks, the non-targeted attacks can be easily performed with high success rate. But the target attacks are still unsatisfactory, which still need large queries and have low success rates. Therefore, this paper focuses on targeted attacks mainly.

3 Methodology

Our method belongs to hybrid methods. Unlike other hybrid methods that use surrogate model gradients as priori knowledge, our method aims at training a more universal adversarial perturbation as an initial value by using surrogate models based on the transferability and adaptation of adversarial examples. According to the setting of black-box attacks, we want to generate a meta adversarial perturbation which can be used to adversarially manipulative different images while being efficiently trainable on small datasets. Inspired by the excellent performance of MAML [7] in many tasks, meta learning method is used and the black-box attacks are modeled analogously. Therefore, due to the transferability of adversarial examples, surrogate models are used.
to generate meta adversarial perturbations (MAPs) based on meta learning methods.

In this section, a method is first proposed which uses meta learning to train meta adversarial perturbations on surrogate models. Then a black-box attack method is proposed which uses the zero-order optimization algorithm to estimate the gradient of target models and adds a momentum term to accelerate attack.

3.1 MAPs training

Formally, consider the black-box model as \( f_\theta \), and the surrogate models as \( f_i, (i = 1, 2, ..., n) \) with known architectures and parameters. A meta adversarial perturbation is denoted as \( v \) which is randomly initialized. \( L \) denotes a loss function which can be a cross-entropy loss, a margin loss or some other loss, and the cross-entropy loss is used in this paper for convenience. \( D \) denotes the train set which is used to generate MAPs. When adapting to a new batch of data points \( B = \{x^t, y^t\} \sim D \), the MAP \( v \) becomes \( v' \) which is adaptable and kind of adversarial. We want to find a universal meta adversarial perturbation \( v \) that can be quickly adapted to any new data point to become a good initial value for attack. In targeted attacks, a perturbation \( v \) is supposed to satisfy

\[
f(x + v) = t, \quad \text{for most } x \sim \mu \quad (1)
\]

Where \( t \) is the target label and \( \mu \) is the data distribution.

It can be found that MAPs are image-agnostic as a trained MAP can adapt to all the data points, no matter whether this point is used as training data for the MAP or not. This is a very good property for a black-box attack setting since the trained MAP can be used directly without contacting the black-box model, which significantly reduces the queries to the black-box model and improves the attack efficiency. In the following, our method is presented for training MAPs in the black-box setting.

The MAPs are trained by using a gradient-based iterative method. So, it is difficult to train MAPs directly on the victim model since the model is black-box and its true gradient cannot be obtained. Exploiting the transferability of adversarial examples and the model-agnosticism of meta learning, we utilize meta learning to train MAPs on the surrogate models and we believe that the MAPs are still well adapted and adversarial in the black-box model. In our method, multi-step gradient descent is used to update perturbation on new data points with surrogate models \( f_i, (i = 1, 2, ..., n) \)

\[
v' \leftarrow v - \alpha \cdot \frac{1}{n} \sum_{i=1}^{n} \nabla_v L(f_i, B + v, t) \quad (2)
\]

Where the learning rate \( \alpha \) is a hyperparameter and usually set to a small value. The perturbation \( v' \) is called adapted perturbation since it has adapted the data points in mini-batch \( B \) and already somewhat adversarial.

The meta perturbation is updated by minimizing the loss corresponding to \( v \) in the new mini-batch of data \( B' \) by adding the adapted perturbation \( v' \). The optimization objective can be described as

\[
\min_v \sum_{\forall u \sim \mathcal{D}} L(f_u, B' + v', t) \quad (3)
\]

The optimization objective as formula (3) is performed over the perturbation \( v \), while the objective is computed on the adapted perturbation \( v' \). In fact, our aim is to first find a better adapted perturbation \( v' \) in data \( B \) and then optimize the meta adversarial perturbation \( v \) at the new data point iteratively. Then, a maximally effective adversarial perturbation \( v \) can be obtained which can perform attack with a high success rate and acquire better adaptability.

Algorithm 1 MAP training

**Input:** Train set \( \mathcal{D} \); learn rate \( \alpha \); learn rate \( \beta \); surrogate model \( f_i \); loss function \( L \); project \( \Pi_l \); target label \( t \).

**Output:** Meta adversarial perturbation \( v \).

1. Randomly initialize \( v \);
2. while not done do
3. for minibatch \( B = \{x', y'\} \sim \mathcal{D} \) do
4. for all \( f_i \) do
5. Evaluate \( \nabla_v L(f_i, B + v, t) \) using minibatch \( B \) with perturbation \( v \);
6. Compute adapted perturbation with gradient descent:
7. \( v' \leftarrow v - \alpha \cdot \frac{1}{n} \sum_{i=1}^{n} \nabla_v L(f_i, B + v, t) \);
end for
9. sample a minibatch of data \( B' \) from \( D \);
10. for all \( f_i \) do
11. Evaluate \( \nabla_{v'} L(f_i, B' + v', t) \) using minibatch \( B \) with adapted perturbation \( v' \);
13. end for
14. project \( v \leftarrow \Pi_l(v) \);
end for
16. end while
17. return \( v \).

Stochastic gradient descent is utilized to update the meta adversarial perturbation \( v \)

\[
v \leftarrow v - \beta \cdot \frac{1}{n} \sum_{i=1}^{n} \nabla_{v'} L(f_i, B' + v', t) \quad (4)
\]

Where \( \beta \) is meta learning rate, which is a hyperparameter like \( \alpha \). The detailed procedure of MAP training is presented in Algorithm 1. At line 14, a projection operation is performed on the updated perturbations to restrict them in a norm bound controlled by \( \epsilon \). A smaller \( \epsilon \) makes an attack less visible to humans.

3.2 Query-efficient attack

In this section, a query-efficient black-box attack algorithm is designed using MAPs trained in section 3.1. When performing an attack, the trained MAP is first added to the input image. Due to the strong effectiveness of MAP, some images can successfully attack the black-box model directly only after adding MAP without other operations. If the attack is not successful, the attack is continuously performed using zero-order optimization algorithms to estimate the gradient of the black-box model. Here the gradient of the model is estimated by using RGF method [20] as

\[
\hat{g} = \frac{1}{q} \sum_{k=1}^{q} \frac{f(x + \sigma u_k, y) - f(x, y)}{\sigma} \cdot u_k \quad (5)
\]

Then the input image \( x \) is updated iteratively using the
estimated gradient \( \hat{\gamma} \) under \( L_\infty \) norm. To accelerate the efficiency of the attack, a momentum term is added which is the difference between two adjacent iterations of the image to the iterative process. This momentum acceleration method was first proposed by Chen et al. [4] and was proved to be effective. The update of image \( x \) is as

\[
x_{i+1} = x_i - \gamma \cdot \text{sign}(\hat{\gamma}) + \eta \cdot (x_i - x_{i-1})
\]

(6)

where \( \gamma \) is step size and \( \eta \) is momentum coefficient, they both should be taken to a reasonably small value. \( \text{sign}(\cdot) \) denotes a symbolic function.

After each update, a projection operation is performed on the current image \( x \) as follows

\[
x = \Pi_c(x)
\]

(7)

where \( \Pi_c(x) \) projects vector \( x \) onto the norm ball \( \|x\|_2 \leq \varepsilon \).

Algorithm 2 outlines the key steps of our proposed black-box meta adversarial attack.

**Algorithm 2** Black-box meta adversarial attack

**Input:** Trained MAP \( v_t \) for target label \( t \); black-box model \( f_b \);

input image \( x \); smooth coefficient \( \sigma \); step size \( \gamma \); momentum coefficient \( \eta \); iterations \( I \); number of queries \( q \); project \( \Pi_c \).

**Output:** Adversarial example \( x_{adv} \).

1. \( x \leftarrow x + \nu \);
2. if \( f_b(x) = t \) then
3. \( x_{adv} = x \);
4. else
5. for \( i = 1 \) to \( I \) do
6. \( \hat{\gamma} = \frac{1}{\beta} \sum_{k=1}^{\beta} \frac{(f(x+\varepsilon u_k,y)-f(x,y)) \cdot u_k}{\varepsilon} \);
7. \( x' = x_i - \gamma \cdot \text{sign}(\hat{\gamma}) + \eta \cdot (x_i - x_{i-1}) \);
8. \( x = \Pi_c(x') \);
9. if \( f_b(x) = t \) then
10. \( x_{adv} = x \);
11. break;
12. else
13. \( x_{i+1} = x \);
14. end if
15. end for
16. end if
17. return \( x_{adv} \).

4 Experiments

In this section, we present the experimental setup and results, and we demonstrate the effectiveness of our method by comparing it with some state-of-the-art black-box attack methods.

4.1 Experimental setup

The representative benchmark dataset CIFAR10 was used to assess our method. The CIFAR10 consists of RGB images with 10 classes containing animal and transportation. The size of the image is \( 32 \times 32 \times 3 \). It includes 50,000 images from the train set and 10,000 images from the test set. The surrogate models and victim models used in our experiment were all trained on the train set, but had different architectures and parameters. Five trained models were used in our experiment, including VGG13, VGG16, GoogleNet, ResNet18 and ResNet34, and all of these models exceeded 85% accuracy on the test set. When performing attack, one model was chosen as the black-box model and others were used to train MAPs as white-box models. The parameters were set as follows: learn rate \( \alpha = 1.5 \times 10^{-5} \) and iterated 20 epochs in MAP training, while smooth coefficient \( \sigma = 0.1 \), number of queries \( q = 14 \), step size \( \gamma = 0.015 \) under \( L_\infty \) norm and momentum coefficient \( \eta = 0.01 \) in black-box attack.

Since the difficulties of non-targeted attacks in the black-box setting have been well addressed, our experiments mainly focused on targeted attacks. For non-targeted attacks, our method can also be applied with a slight modification.

4.2 Targeted attacks

100 images were selected from each of the 10 classes, i.e. 1000 images to form the dataset \( D \), and the other 900 images were selected from 9 classes except the target class as the attacked dataset. When training the MAPs, the minibatches were set to \( B = 600 \) and \( B^2 = 1000 \). Then stochastic gradient descent was used to update MAPs for the target attack. Our method was compared with RGF [20], NES [9] and P-RGF [5] with the same attack settings to be fair. Since our method added a MAP to the input image, a random perturbation was also added with the same distortion as a comparison. We chose target label \( t = 6 \) and set 600 as the maximal number of queries on the black-box model. The results of the attack are as Table 1.

**Table 1:** Targeted attacks

| Model     | Method  | Success | Queries | Avg. \( L_2 \) norm |
|-----------|---------|---------|---------|---------------------|
| VGG13     | RGF     | 85.11%  | 270.53  | 3.32                |
|           | Random  | 88.44%  | 240.43  | 4.00                |
|           | P-RGF   | 82.00%  | 137.16  | 3.18                |
|           | NES     | 76.56%  | 310.48  | 3.62                |
|           | Ours    | 96.33%  | 130.45  | 3.23                |
| VGG16     | RGF     | 87.11%  | 272.03  | 3.32                |
|           | Random  | 89.33%  | 249.51  | 4.03                |
|           | P-RGF   | 87.22%  | 120.50  | 2.96                |
|           | NES     | 78.33%  | 305.93  | 3.58                |
|           | Ours    | 97.11%  | 108.09  | 2.80                |
| GoogleNet | RGF     | 97.11%  | 243.07  | 3.17                |
|           | Random  | 98.22%  | 221.36  | 3.92                |
|           | P-RGF   | 98.89%  | 129.44  | 3.53                |
|           | NES     | 88.33%  | 291.73  | 3.53                |
|           | Ours    | 99.00%  | 148.69  | 3.52                |
| ResNet18  | RGF     | 98.33%  | 202.98  | 2.90                |
|           | Random  | 98.67%  | 184.58  | 3.63                |
|           | P-RGF   | 99.00%  | 107.18  | 2.96                |
|           | NES     | 95.00%  | 238.49  | 3.18                |
|           | Ours    | 99.67%  | 80.67   | 2.75                |
| ResNet34  | RGF     | 94.00%  | 233.56  | 3.09                |
|           | Random  | 95.67%  | 197.14  | 3.80                |
|           | P-RGF   | 96.89%  | 134.11  | 3.29                |
|           | NES     | 86.89%  | 268.90  | 3.37                |
|           | Ours    | 98.33%  | 108.42  | 3.45                |

Table 1 shows the success rate of black-box attacks, the average number of queries needed to generate an adversarial example and the average distortion of all adversarial examples. It can be seen that our method generally leads to higher success rates, fewer queries and relatively smaller distortions than other methods. When the number of queries is limited, our method has around 10% improvement in suc-
cess rates over other methods in some models. And our method usually uses fewer queries to achieve the same success rate. Although the distortions caused by our method are slightly larger in some models, they are still at a very small level. The results also show that the MAPs generated by our method have great transferability and adaptation on the new data points, since they can work well on the attacked dataset and perform the black-box attack well. Experiments prove that our method is much more queries-efficient and more effective than other methods.

Fig. 1 illustrates the adversarial examples generated by our method. The left column shows the original images to be attacked, the middle column shows the adversarial perturbations generated by our method and the right column shows the adversarial examples corresponding to the target class. It can be seen that the distortions caused by our method are almost invisible.

4.3 Adaptability and universality

In this section, the transferability and universality of MAPs has been further investigated. It is intriguing that whether the performance of MAPs in the attack is sensitive to the selection of training data. The animal images in CIFAR10 are chosen as the training dataset while the transportation images as the attacked dataset. This is a great test for the adaptability and versatility of MAPs, because the MAPs have never learnt the features of the attacked images during the training process. If MAPs can still obtain great attack results under such unfavourable conditions, it can be demonstrated that the MAPs trained by our method are well adapted and universal. Here 100 images were selected from the 6 classes of animals to form the dataset $\mathbb{D}$, and the other 300 images were selected from the classes of transportation except the target class. When training the MAPs, the mini-batches were set to $B = 360$ and $B' = 600$. Our method was compared with the methods in section 4.1. We chose target label $t = 9$ and set 600 as the maximal number of queries on the black-box model. The results of the attack are as Table 2.

| Model  | Method | Success Rate | Queries | Avg. $L_2$ norm |
|--------|--------|--------------|---------|----------------|
| VGG13  | RGF    | 86.33%       | 244.16  | 3.14           |
|        | Random | 85.33%       | 230.38  | 4.86           |
|        | P-RGF  | 83.00%       | 111.83  | 2.90           |
|        | NES    | 75.67%       | 286.17  | 3.46           |
|        | Ours   | 96.33%       | 158.67  | 3.39           |
| VGG16  | RGF    | 80.33%       | 243.37  | 3.13           |
|        | Random | 79.00%       | 227.02  | 4.95           |
|        | P-RGF  | 79.33%       | 115.87  | 2.76           |
|        | NES    | 69.33%       | 281.14  | 3.48           |
|        | Ours   | 95.67%       | 165.81  | 3.46           |
| GoogleNet | RGF | 97.67%       | 265.29  | 3.30           |
|        | Random | 98.33%       | 252.49  | 4.98           |
|        | P-RGF  | 96.33%       | 177.86  | 3.43           |
|        | NES    | 82.67%       | 316.18  | 3.77           |
|        | Ours   | 98.67%       | 186.98  | 3.72           |
| ResNet18 | RGF | 93.33%       | 280.94  | 3.41           |
|        | Random | 95.33%       | 265.21  | 4.95           |
|        | P-RGF  | 91.67%       | 242.34  | 3.51           |
|        | NES    | 77.67%       | 329.93  | 3.75           |
|        | Ours   | 96.33%       | 230.34  | 4.01           |
| ResNet34 | RGF | 91.33%       | 285.94  | 3.43           |
|        | Random | 90.67%       | 209.19  | 5.00           |
|        | P-RGF  | 91.00%       | 218.18  | 3.58           |
|        | NES    | 77.00%       | 339.51  | 3.81           |
|        | Ours   | 95.00%       | 201.25  | 3.76           |

As shown in Table 2, our method achieves a higher success rate than other methods without knowing the features of the attacked class. Our method achieves the highest attack success rate in all models, and the success rate of our method is even improved by more than 10% in some models. And our method still maintains few queries for the black-box model. The queries of our methods are kept to the fewest or nearly the fewest. Due to the absence of feature knowledge, our method causes slightly larger distortions in some models. However, our method still outperforms other methods in terms of overall performance. It also demonstrates the excellent adaptability and universality of the MAPs generated by our method.

From the experimental results, it can be concluded that the MAPs generated by our method have excellent adaptability and universality in different network structures and different
data points. They can execute attacks independently of the network and data, which is very suitable for the black-box setting. Therefore, it can be said that the MAPs generated by our method are truly universal adversarial perturbations for black-box attacks.

5 Conclusion

In this paper, we have proposed a new black-box attack method to utilize the meta adversarial perturbations (MAPs) for boosting black-box attack. The method contains two stages: (1) training MAPs on the surrogate models using meta learning method; (2) performing black-box attack with the trained MAPs and estimating the gradient of the black-box models efficiently. The experimental results show that our method can achieve higher success rates and fewer queries than other methods. And the MAPs trained by our method have excellent transferability and universality. In future work, we will try to train more efficient MAPs on large datasets and further boost black-box attacks.

References

[1] N. Carlini and D. Wagner, “Towards evaluating the robustness of neural networks,” in Proc. IEEE Symp. Secur. Priv., 2017, pp. 39–57.
[2] J. Chen, J. Sun, and G. Wang, “From unmanned systems to autonomous intelligent systems,” Eng., vol. 8, pp. 1–5, 2022.
[3] P.-Y. Chen, H. Zhang, Y. Sharma, J. Yi, and C.-J. Hsieh, “Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models,” in Proc. ACM Workshop Artif. Intell. Secur., 2017, pp. 15–26.
[4] X. Chen, Y. Tang, and N. Li, “Improve single-point zeroth-order optimization using high-pass and low-pass filters from extremum seeking control,” arXiv:2111.01701, 2021.
[5] S. Cheng, Y. Dong, T. Pang, H. Su, and J. Zhu, “Improving black-box adversarial attacks with a transfer-based prior,” in Proc. Int. Conf. Neural Inf. Proc. Syst., 2019, pp. 10934–10944.
[6] Y. Dong, F. Liao, T. Pang, H. Su, J. Zhu, X. Hu, and J. Li, “Boosting adversarial attacks with momentum,” in Proc. IEEE Conf. Comput. Vision Pattern Recognit., 2018, pp. 9185–9193.
[7] C. Finn, P. Abbeel, and S. Levine, “Model-agnostic meta-learning for fast adaptation of deep networks,” in Int. Conf. Mach. Learn., 2017, pp. 1126–1135.
[8] I. Goodfellow, J. Shlens, and C. Szegedy, “Explaining and harnessing adversarial examples,” arXiv:1412.6572, 2014.
[9] A. Ilyas, L. Engstrom, A. Athalye, and J. Lin, “Black-box adversarial attacks with limited queries and information,” in Int. Conf. Mach. Learn., 2018, pp. 2137–2146.
[10] S. Khan, M. Tufail, M. T. Khan, Z. A. Khan, J. Iqbal, and A. Wasiem, “A novel framework for multiple ground target detection, recognition and inspection in precision agriculture applications using a UAV,” Unmanned Syst., vol. 10, no. 1, pp. 45–56, 2021.
[11] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” Adv. Neural Inf. Process. Syst., vol. 25, no. 2, pp. 1097–1105, 2012.
[12] A. Kurakin, I. Goodfellow, and S. Bengio, “Adversarial examples in the physical world,” arXiv:1607.02533, 2016.
[13] J. Li, M. Ran, H. Wang, and L. Xie, “A behavior-based mobile robot navigation method with deep reinforcement learning,” Unmanned Syst., vol. 9, no. 3, pp. 201–209, 2021.
[14] K. Li, X. Xie, W. Xue, and X. Chen, “Hybrid teaching—learning artificial neural network for city-level electrical load prediction,” Sci. China Inf. Sci., vol. 63, no. 5, pp. 212–214, 2020.
[15] Z. Li, Z. Chen, W.-S. Zheng, S. Oh, and N. Kien, “Ar-cnn: An attention ranking network for learning urban perception,” Sci. China Inf. Sci., vol. 65, no. 1, pp. 164–174, 2022.
[16] S. Liu, J. Sun, and J. Li, “Query-efficient hard-label black-box attacks using biased sampling,” in 2020 Chinese Automation Congress, 2020, pp. 3872–3877.
[17] W. Liu, J. Sun, G. Wang, F. Bullo, and J. Chen, “Data-driven resilient predictive control under denial-of-service,” IEEE Trans. Autom. Control, 2022.
[18] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, “Towards deep learning models resistant to adversarial attacks,” arXiv:1706.06083, 2017.
[19] S.-M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, “Deepfool: a simple and accurate method to fool deep neural networks,” in Proc. IEEE Conf. Comput. Vision Pattern Recognit., 2016, pp. 2574–2582.
[20] Y. Nesterov and V. Spokoiny, “Random gradient-free minimization of convex functions,” Found. Comut. Math., vol. 17, no. 2, pp. 527–566, 2017.
[21] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik, and A. Swami, “Practical black-box attacks against machine learning,” in Proc. ACM Asia Conf. Comput. Commun. Secur., 2017, pp. 506–519.
[22] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami, “The limitations of deep learning in adversarial settings,” in Proc. IEEE Eur. Symp. Secur. Priv., 2016, pp. 372–387.
[23] M. Sharif, S. Bhagavatula, L. Bauer, and M. K. Reiter, “Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition,” in Proc. ACM SIGSAC Conf. Comput. Commun. Secur., 2016, pp. 1528–1540.
[24] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, “Intriguing properties of neural networks,” arXiv:1312.6199, 2013.
[25] C.-C. Tu, P. Ting, P.-Y. Chen, S. Liu, H. Zhang, J. Yi, C.-J. Hsieh, and S.-M. Cheng, “Autozoom: Autoencoder-based zeroth order optimization method for attacking black-box neural networks,” in Proc. AAAI Conf. Artif. Intell., vol. 33, no. 1, 2019, pp. 742–749.
[26] G. Wang, B. Giannakis, Georgios, and J. Chen, “Robust and scalable power system state estimation via composite optimization,” IEEE Trans. Smart Grid, vol. 10, no. 6, pp. 6137–6147, 2019.
[27] G. Wang, G. B. Giannakis, and I. Chen, “Learning ReLU networks on linearly separable data: Algorithm, optimality, and generalization,” IEEE Trans. Signal Process., vol. 67, no. 9, pp. 2357–2370, 2019.
[28] G. Wu, G. Wang, J. Sun, and L. Xiong, “Optimal switching attacks and countermeasures in cyber-physical systems,” IEEE Trans. Syst. Man Cybern. Syst., vol. 51, no. 8, pp. 4825–4835, 2019.
[29] W. Xiao, G. C. Candiasaras, and C. Bela, “Safety-critical optimal control for autonomous systems,” J. Complex Syst. Complexity, vol. 34, no. 5, pp. 1723–1742, 2021.
[30] Y. Xiao, C.-M. Pun, and B. Liu, “Adversarial example generation with adaptive gradient search for single and ensemble deep neural network,” Inf. Sci., vol. 528, pp. 147–167, 2020.
[31] C.-H. Yuan, P.-Y. Chen, and C.-M. Yu, “Meta adversarial perturbations,” arXiv:2111.10291, 2021.