I.J. Image, Graphics and Signal Processing, **2022, 2, 16-24**
Published Online April 2022 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijigsp.2022.02.02

Towards Query Efficient and Derivative Free Black Box Adversarial Machine Learning Attack

Amir F. Mukeri
AISSMS College of Engineering, Pune, 411001, India
E-mail: mukeriamir@gmail.com

Dwarkoba P. Gaikwad
AISSMS College of Engineering, Pune, 411001, India
E-mail: dpgaikwad@aissmscoe.com

Received: 20 November 2021; Revised: 15 December 2021; Accepted: 08 January 2022; Published: 08 April 2022

Abstract: While deep learning has shown phenomenal success in many critical applications such as in autonomous driving and medical diagnosis, it is vulnerable to black box adversarial machine learning attacks. Objective of these attacks is to mislead a classifier in making mistakes. Hard Label attacks are those in which an adversary has access only to the top-1 prediction label and has no knowledge about model parameters or gradient loss. Secondly, for security concerns, the number of model queries that an attacker can perform for evaluation are restricted. In this paper, we propose a novel nature-inspired optimization algorithm for generating adversarial examples. Proposed algorithm is derivative-free, meta-heuristic algorithm. It searches for optimum adversarial examples in high-dimensional image space using simple arithmetic operations inspired by Brownian motion of molecules in fluids and gases. Experiments with CIFAR-10 image dataset yielded encouraging results with a query budget of less than 1000 and with a minimal distortion to original image. Its performance was determined to be comparable and exceeded in some cases compared to previous state of the art attacks.

Index Terms: Adversarial Machine Learning, Robust Deep Learning, Nature Inspired Optimization, Computer Vision, Security & Privacy.

1. Introduction

Deep learning has emerged as the preferred approach for a wide range of object detection applications, from facial recognition to autonomous driving. Complex deep neural network designs, such as Convolution Neural Networks (CNN) and its variants, have been implemented or are being explored for use in a numerous mission-critical application. However, it has been shown that Deep Neural Networks (DNNs) are not without flaws. DNNs, like conventional statistical machine learning models, are vulnerable to adversarial example attacks. Adversarial examples are created from natural images by subtly altering them. Although adversarial samples appear identical to the original image to human eye, image classifiers fail to accurately classify them. As shown in Fig. 1, leftmost image shows an unmodified image of a bird that is classified correctly by a CNN model, image in the center indicates the perturbation added to the original image. Rightmost image shows the resulting adversarial example that is misclassified by CNN as a frog.

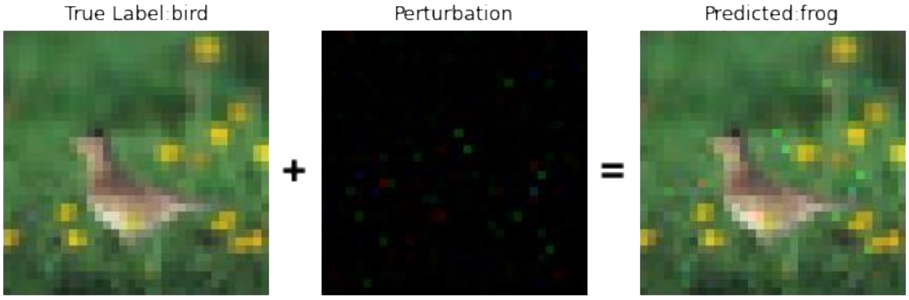

Fig.1. Adversarial Example Attack on image classifier

Adversarial attacks have been demonstrated to be effective in real word setting. For example, in case of autonomous driving, adversarial stop traffic sign is predicted to be a speed limit sign leading to catastrophic consequences [1]

1.1. Taxonomy of Adversarial Attacks on Deep Neural Network

Adversarial attacks are characterized as either white box or black box attacks based on the degree of information accessible to an attacker. In white box attack, an attacker has access to model parameters such as neural network weights or gradient of loss. In contrast, in black box attacks, an attacker has little or no knowledge of the trained neural network parameters.

As illustrated in Fig. 2, black box attacks are further divided into soft label score (output probability) based attacks and hard label attacks.

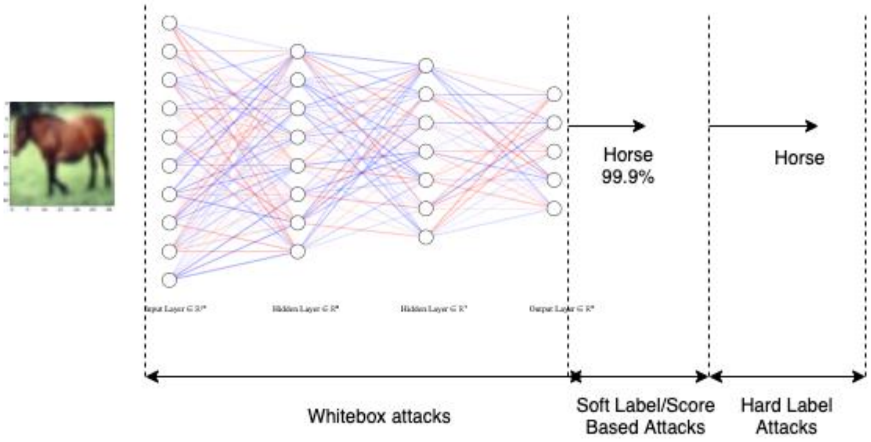
Input Layer $\in \mathbb{R}^{10}$
Hidden Layer $\in \mathbb{R}^{8}$
Hidden Layer $\in \mathbb{R}^{7}$
Output Layer $\in \mathbb{R}^{5}$
Horse
99.9%
Horse
Whitebox attacks
Soft Label/Score Based Attacks
Hard Label Attacks

Fig.2. Adversarial Example Attack Classification

In Soft label attacks, attackers have access to only the highest probability score of a neural network output i.e. confidence scores and in hard label attacks, attackers have access to only final label. Soft label attacks rely on the neural network's probability output to generate adversarial instances, whereas hard label attacks rely only on the final label (top-1 prediction). The only information available to an attacker in a hard label attacks is binary, i.e. whether an attack is successful or not. There is no information on the probability score or gradient.

Hard label attacks are more realistic in practice since attacker typically does not have access to model parameters. However, in order to produce adversarial samples with the least amount of distortion, black box attacks end up querying models for far too many times. As a result, the target model may identify a high number of incoming requests and reject any new queries, thereby preventing future attacks. To be a realistic attack in practice, the black box attack must meet two constraints, namely,

- Limitation on the number of model queries and
- The least amount of distortion to the original image as determined by a distance metric such as an L_p norm.

Difference between the original and adversarial images decreases as the number of queries for model evaluation increases. However, if there are high number of incoming requests from the same source, the target system may prohibit any further inquiries. A trade-off must be struck between the number of queries for model evaluation and the distortion to original image. This is the most challenging task for any adversarial black box attack. In this paper, we present novel hard label black box attack with limited query budget.

1.2. Main Contributions

Main contributions of this paper are as follows:

• Query efficient hard-label and derivative free black box attack
• Query efficient soft-label and derivative free black box attack
• Brownian Arithmetic Optimization Algorithm (BAOA) for high dimensional continuous, non-convex optimization problems based on an optimization algorithm proposed by Abualigah et al. [2]
• Evaluation of efficacy of proposed attack on real world CIFAR-10 image dataset [3].

Remainder of the paper is organized as follows. Section 2 presents recent research on black box attacks, followed by our proposed optimization algorithm in Section 3. We discuss experimental findings in Section 4, and finally conclusions of the research are presented in Section 5.

2. Related Work

Following is an overview of the recent research work in the area of adversarial black box attacks. As mentioned in Section 1.1. black box attacks are classified into Soft Label and Hard Label attack.

2.1. Soft Label Black Box Attack

Soft Label attack is dependent on the model's output probability score, although not all of them are derivative free. For instantiating adversarial examples, Zeroth Order Optimization (ZOO) [4] utilizes the probability score from the logit layer to estimate the gradient of DNN. This technique searches for adversarial instances in an image space using output score and coordinated descent algorithm. As a result, it actually ends up making a massive number of queries, numbering in the hundreds of thousands to the target model. Ilyas et al. [5] suggested Neural Evolution strategy (NES) framework to estimate the gradient of loss using bandit optimization from control theory. Tau et. al. [6,7] proposed autoencoder to generate adversarial images rather than estimating approximate gradient. Another attack in this category is attack that formulates the problem as an optimization problem of finding adversarial examples, however it uses gradient based optimization methods to optimize the loss [8]. Mosli et al. [9] introduced the AdvPSO attack in this context, which makes use of the Particle Swarm Optimization (PSO) method. This is a derivative-free method, although it does rely on the output probability score for generating adversarial examples.

2.2. Hard Label Black Box Attack

This is the most challenging and realistically feasible threat model. In this configuration, an attacker has only access to the final prediction, which is a binary value denoting whether an attack is successful or not. Brendel et al. [10] proposed a boundary attack that utilizes a random walk around the decision boundary to look for adversarial examples. Cheng et al. [11] presented a unique OPT approach based on Random Gradient Free optimization for optimizing the non-convex step function, i.e. a hard label output. They discovered that by employing this approach, they could minimize the number of queries by 3 to 4 times when compared to the Boundary attack. Bruner et al. [12] introduced the Biased Boundary attack, which is based on the previous Boundary attack and makes use of image frequency, regional masks, and surrogate gradients. They were able to minimize the number of queries and perturbations required to produce an adversarial example even more. Chen et al.'s [13] Sign-OPT attack expands on earlier work on OPT attack utilizing directional derivative to estimate the sign of the derivative instead of an actual gradient.

3. Proposed Methodology

This section present the methodology beginning with problem formulation. In the proposed setting, search of an adversarial example is formulated as an optimization problem. The output of target function is defined as noted in Equation 1. We primarily focus on untargeted attacks, in which the goal of an attack is to cause the model to misclassify an input sample as anything other than its true label.

3.1. Attack Fomulation

Formulation of hard label attack is depicted in Equation 1.

$$g(x') = \begin{cases} K & f(x') = C \\ L_2(x, x') & f(x') \neq C \end{cases} \tag{1}$$

$$\text{Such that } \mathrm{x}' = \mathrm{x} + \delta, \delta \leq \epsilon, \epsilon \text{ is maximum } \mathrm{L}_\infty \text{ norm}$$

In above Equation 1, we define a new function g(.) and x′ as an adversarial example perturbed by small amount of distortion δ. The perturbation δ is bound by threshold L∞ norm specified by an attacker. C is true label of the original clean image x correctly classified by model $f(.)$. If target model f classifies x' correctly then a large constant K is returned i.e. if an attack is not successful large constant K is returned. In case attack is successful i.e. output label is not same as true label then Euclidean L_2 distance between original image and adversarial image is returned by $g(.)$.

This formulation effectively turns g into a *step function*. Step functions cannot be optimized using gradient-based methods such as Stochastic Gradient Descent (SGD) or its variants. As a result, for such optimization problems, gradient free techniques are preferred.

Since Soft-Label attack has access to the ultimate probability score of the model output, it is stated as a continuous optimization problem, as seen in Equation 2

$$g(x') = \begin{cases} K + c * f(x') & f(x' = C \\ L_2(x, x') & f(x') \neq C \end{cases} \tag{2}$$
$$Such\ that\ x' = x + \delta, \delta\ \leq\ \epsilon\ , \epsilon\ is\ maximum\ L_\infty\ norm$$

In contrast to hard-label settings, if an attack fails, instead of returning a large constant K, the sum of K and the output score multiplied by another controlling constant c is retuned. If an attack is successful, similar to a hard-label attack, distance between the original and adversarial images is returned.

3.2. *Brownian Arithmatic Optimization Algorithm (BAOA)*

On a set of random test images, nature inspired optimization methods such as Genetic Algorithms (GA), Ant Bee Colony (ABC), Particle Swarm Optimization (PSO), Crow Search Algorithm, Firefly Algorithm(FA), Flower Pollination Algorithm(FPA), Differential Evolution (DE), and Arithmetic Optimization Algorithm (AOA) were evaluated. Implementation of this algorithm from Opytimizer[1] and Facebook's Nevergrad[2] package was used. AOA was chosen as the preferred technique due to the less number of queries required to search adversarial examples and lesser L_2 distortion compared to other derivative free algorithms [2].

In case of meta-heuristic algorithms such as AOA, optimization process is divided into two stages: *exploration* and *exploitation*. Exploration phase refers to large search space coverage utilizing search agents of an algorithm to avoid local optima. Exploitation refers to the enhancement of the correctness of discovered solutions during the exploration phase. Also, this process is stochastic in nature. However, we must modify AOA to meet the requirements of the situation at hand. AOA is a population based algorithm inspired by natural arithmetic processes such as addition, subtraction, multiplication, and division. As illustrated in Fig. 3, it relies on multiplication and division operations for exploration and addition and subtraction operations for exploitation.

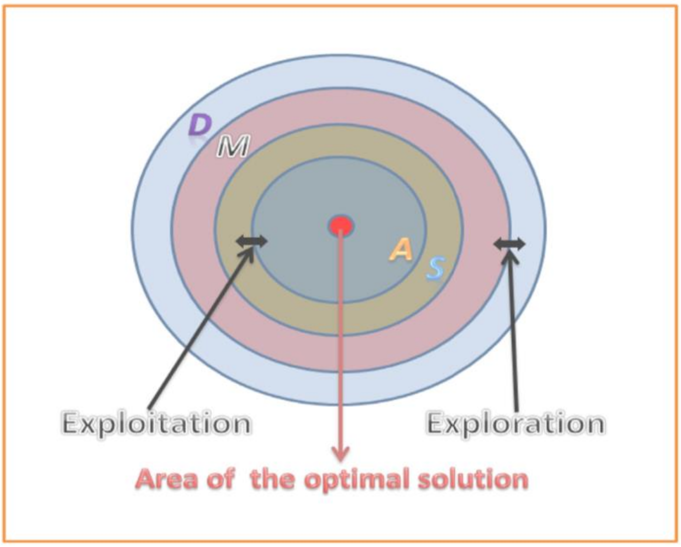

Fig.3. AOA: Exploration and exploitation using basic arithmetic operation [2]

The number of iterations is budgeted in advance. A transition from exploration to exploitation search is conditioned by a function, denoted as MOA, that favors exploration in earlier iterations and exploitation in later iterations. as shown in Equation 3.

$$MOA(C_{iter}) = Min + C_{iter} \times \frac{Max - Min}{M_{iter}} \tag{3}$$

In Equation 3, for MOA, C_{Iter} is current iteration, Max and Min are maximum and minimum bounds of a variable, and M_{Iter} is maximum number of allowed iterations.

Exploration phase is kept unchanged from original AOA method. Its defined as in Equation 4,

$$x_{ij}(C_{iter} + 1) = \begin{cases} best(x_i) \div MOP \times ((UB_j - LB_j) \times \mu + LB_j),\ r_2 < 0.5 \\ best(x_i)\ \times\ MOP \times ((UB_j - LB_j) \times \mu + LB_j),\ otherwise \end{cases} \tag{4}$$

[1] https://github.com/gugarosa/opytimizer
[2] https://facebookresearch.github.io/nevergrad/

where,

C_{iter} : Current Iteration
UB & LB: Upper and Lower bounds of variables
μ: Control parameter set to 0.499
r_2: Random number generated from uniform distribution in [0,1].

With equal probability, the uniform random number determines whether the next operation performed is multiplication or division.

MOP defines the step size as in (5).

$$MOP(C_{iter}) = 1 - \frac{C_{iter}^{1/\alpha}}{M_{iter}^{1/\alpha}} \tag{5}$$

According to AOA, the exploitation phase involves performing addition and subtraction operations on the best agent location discovered thus far. However, in our research it was discovered experimentally that the original method of exploitation might be fine-tuned further particularly for high-dimensional problems like the current one. Brownian motion, which is inspired by molecular motions, is utilized to fine tune the exploitation phase with an update rule, as illustrated in Equation 6. Adding the Brownian motion to exploitation phase is the major contribution of our research to original AOA algorithm which does not have this step.

$$x_{ij}(C_{iter}+1) = \begin{cases} best(x_i) - Normal(0,\delta^2) \times ((UB_j - LB_j) \times \mu + LB_j),\ r_2 < 0.5 \\ best(x_i) +\ Normal(0,\delta^2) \times ((UB_j - LB_j) \times \mu + LB_j),\ otherwise \end{cases} \tag{6}$$

Instead of utilizing MOP as step size, random number generated from Normal distribution with mean 0 and as variance is used for addition or subtraction operation from best known position so far. δ is a control parameter which is set to 0.25.

Overall, Brownian AOA algorithm is shown in Algorithm 1.

Algorithm 1: Brownian Arithmetic Optimization Algorithm (BAOA)

```
Initialize the Arithmetic Optimization Algorithm parameters α, μ.
Initialize the solutions’ positions randomly. (Solutions: i=1, ..., N.)
while (C_Iter < M_Iter) do
  Calculate the Fitness Function (FF) for the given solutions
  Find the best solution (Determined best so far).
  Update the MOA value using Eq. (2).
  Update the MOP value using Eq. (4).
  for (i = 1 to Solution) do
    for j = 1 to Positions) do
      Generate a random value between [0, 1] (r1, r2, and r3)
      if (r1 > MOA) then
        Exploration Phase
        if (r2 > 0.5) then
          (1) Apply the Division math operator (D "÷").
          Update the ith solutions’ positions using the first rule in Eq. (3).
        else
          (2) Apply the Multiplication math operator (M "×")
          Update the ith solutions’ positions using the second rule in Eq. (3).
        end if
      else
       Exploitation phase using Brownian motion
       if (r3 >0.5) then
         Apply the Subtraction math operator (S "-").
     Update the ith solutions’ positions using the first rule in Eq. (5)
      else
        (2) Apply the Addition math operator (A " + ").
        Update the ith solutions’ positions using the second rule in Eq. (5).
      end if
    end if
  end for
 end for
C_iter = C_iter + 1
end while
Return best solution (x)
```

4. Results and Discussion

Experiments employ the Convolution Neural Network (CNN), which has been used in previous investigations of hard label attacks. CNN under investigation is composed of four convolutional layers, two max-pooling layers, and two fully connected layers. This is the same CNN architecture that has been utilized in previous attacks. This facilitates in comparing the outcomes with previous attacks.

The CIFAR-10 image dataset was leveraged for benchmarking. The dataset was divided into two parts: training and testing. On clean, unmodified test data, classification accuracy was found to be around 83%. The maximum allowed L∞ norm threshold was set to be 0.25.

4.1. Hard Label Attack Performance

For the adversarial attack, 100 correctly classified images by CNN model under attack were chosen at random. Table 1 compares proposed BAOA attack to other state-of-the-art hard label attacks. BAOA could converge to the optimal area relatively rapidly with a small number of queries, in this example, 820 queries with a higher attack success rate (ASR) than that of the Boundary and OPT attacks. BAOA achieves lesser average L2 distortion than Boundary attack with a far less number of queries. To compare BAOA with different population methods, a Particle Swarm Optimization (PSO) attack was carried out. While PSO produces equivalent ASR, it has the highest distortion to image of 7.3, violating the threat model requirements set in Section 1.1 .

Table 1. BOA attack comparison with other attacks under hard label setting

Attack	**Attack Success Rate (%)**	**Average L_2 Distance**	**#Queries**
Boundary	2.3	3.12	4000
OPT	37	0.77	4000
Sign-OPT	73	0.26	4000
PSO	49	7.3	820
BAOA (ours)	46	1.61	**820**

Table 2 shows the results of additional tests with different query budgets for the BAOA attack. Even with a query budget of only 210 queries, BAOA achieves a 39 % ASR with an average L_2 distortion of only 1.8, which is similar to other recent attacks.

Table 2. BAOA attack performance on different query budgets

Queries	**Attack Success Rate (%)**	**Average L_2 Distance**
210	39	1.8
465	41	1.29
820	47	1.61
1620	46	1.15
4020	**53**	**1.11**

Fig. 4 depicts a few successful adversarial images with varying query budgets. As seen in this figure, as number of queries allowed for an attack increases the L_2 distortion becomes smaller and smaller and it becomes difficult to notice the difference between original clean image and corresponding perturbed adversarial image.

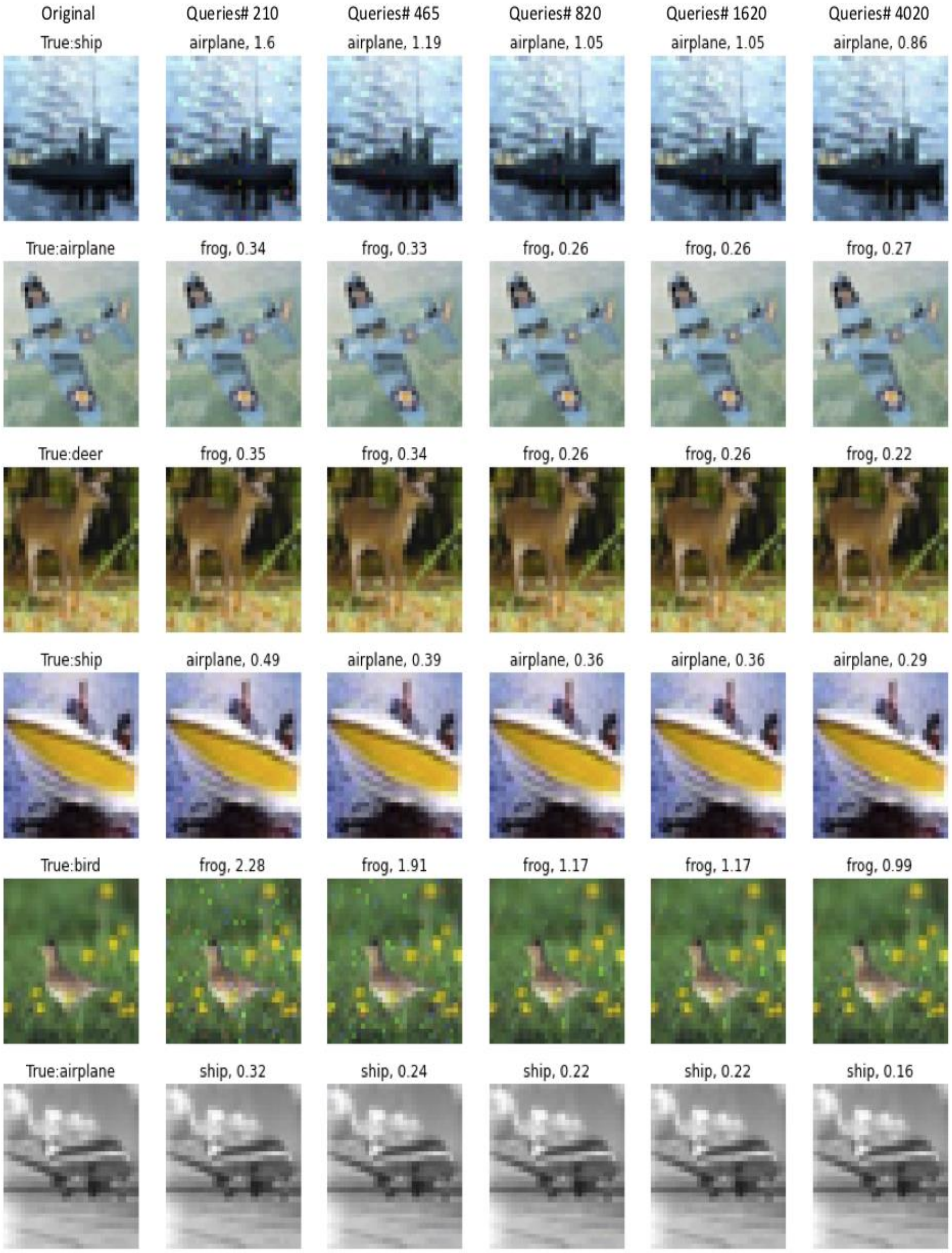

Fig. 4. Depicts a few successful adversarial images with varying query budgets.

4.2. Soft Label Attack Performance

Score-based soft label attacks were also evaluated, in which the attacker has access to a final output label's probability score. Table 3 compares the performance of the BAOA attack with other attack types. On the number of queries parameter, BAOA beats all other attacks.

Table 3. Comparison with other Soft Label black box attacks

Attack	Attack Success Rate (%)	Average L_2 Distance	#Queries
ZOO	100	0.199	128,000
Adv PSO	99.6	1.414	1224
PSO	77	7.949	820
BAOA (ours)	69	**1.99**	**820**

5. Conclusions

Deep learning has made its way into a number of critical applications. As a result, awareness of adversarial example attacks is important for the safety and security of systems, assets, and, most importantly, people. The work presented in this paper addresses the most challenging adversarial evasion threat model, namely hard label black box attacks. In this configuration, an attacker only has access to the final predicted label and no additional gradient or model parameter information. In addition, there is a limit to the number of times models may be queried for evaluation. This is the most realistic and practical attack scenario. The suggested attack is based on the meta-heuristic, nature-inspired Brownian Arithmetic Optimization Algorithm (BAOA). BAOA explores and exploits optimum search space for adversarial examples using simple arithmetic operations such as addition, subtraction, multiplication, and division combined with Brownian motion to perturb original image. It is a derivative-free optimization approach that depends solely on binary output from model assessment, namely whether or not an attack is successful. Despite its simplicity, BAOA performs a successful adversarial attack with a query budget of as little as 210 queries and with a minimal distortion to original image.

References

[1] I. Evtimov, K. Eykholt, E. Fernandes, T. Kohno, B. Li, A. Prakash, A. Rahmati and D. Song, "Robust physical-world attacks on machine learning models," arXiv preprint arXiv:1707.08945, vol. 2, p. 4, 2017.

[2] L. Abualigah, A. Diabat, S. Mirjalili, M. Abd Elaziz and A. H. Gandomi, "The arithmetic optimization algorithm," Computer methods in applied mechanics and engineering, vol. 376, p. 113609, 2021.

[3] A. Krizhevsky, V. Nair and G. Hinton, "Cifar-10 (canadian institute for advanced research)," URL http://www. cs. toronto.edu/kriz/cifar.html, vol. 5, p. 4, 2010.

[4] 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 Proceedings of the 10th ACM workshop on artificial intelligence and security, 2017.

[5] A. Ilyas, L. Engstrom and A. Madry, "Prior convictions: Black-box adversarial attacks with bandits and priors," arXiv preprint arXiv:1807.07978, 2018.

[6] P. Tao, Z. Sun and Z. Sun, "An improved intrusion detection algorithm based on GA and SVM," Ieee Access, vol. 6, p. 13624–13631, 2018.

[7] 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 Proceedings of the AAAI Conference on Artificial Intelligence, 2019.

[8] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in 2017 ieee symposium on security and privacy (sp), 2017.

[9] R. Mosli, M. Wright, B. Yuan and Y. Pan, "They might not be giants: crafting black-box adversarial examples with fewer queries using particle swarm optimization," arXiv preprint arXiv:1909.07490, 2019.

[10] W. Brendel, J. Rauber and M. Bethge, "Decision-based adversarial attacks: Reliable attacks against black-box machine learning models," arXiv preprint arXiv:1712.04248, 2017.

[11] M. Cheng, S. Singh, P. Chen, P.-Y. Chen, S. Liu and C.-J. Hsieh, "Sign-opt: A query-efficient hard-label adversarial attack," arXiv preprint arXiv:1909.10773, 2019.

[12] T. Brunner, F. Diehl, M. T. Le and A. Knoll, "Guessing smart: Biased sampling for efficient black-box adversarial attacks," in Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019.

[13] M. Cheng, T. Le, P.-Y. Chen, J. Yi, H. Zhang and C.-J. Hsieh, "Query-efficient hard-label black-box attack: An optimization-based approach," arXiv preprint arXiv:1807.04457, 2018.

[14] H. A. Kholidy and F. Baiardi, "Cidd: A cloud intrusion detection dataset for cloud computing and masquerade attacks," in 2012 Ninth International Conference on Information Technology-New Generations, 2012.

Authors' Profiles

Amir F. Mukeri received his Diploma in Computer Engineering from AISSMS Polytechnic, Pune, India, B.E. degree in Information Technology from P.V.G.'s College of Engineering & Technology, Pune, India and M.E. Computer Engineering from AISSMS's College of Engineering, Pune, India. He has more than 15 years of experience working in the software products & SaaS industry in the domain of cloud computing, security, data storage and protection, virtualization and IOT in India and US. He is a member of IEEE & ACM.

Dr. Dwarkoba. P. Gaikwad received his B.E. degree in Computer Science and Engineering from Shri Guru Gobind Singhji Institute of Engineering and Technology, Nanded, Maharashtra, India and M.S. degrees in Electrical Engineering from College of Engineering, Pune, Maharashtra, India in 1996 and 2006 respectively. He has been awarded Ph.D. degree in Computer Science and Engineering in 2017. Currently, he is working as an Associate Professor and Head of Department of Computer Engineering in AISSMS College of Engineering, Pune, Maharashtra, India. He has published more than 40 papers in International journal and conferences. He received best researcher award in International Scientist Award Conference held at Vishakhapatnam, India.

How to cite this paper: Amir F. Mukeri, Dwarkoba P. Gaikwad, "Towards Query Efficient and Derivative Free Black Box Adversarial Machine Learning Attack", International Journal of Image, Graphics and Signal Processing(IJIGSP), Vol.14, No.2, pp. 16-24, 2022.DOI: 10.5815/ijigsp.2022.02.02