UNIVERSITY OF CALIFORNIA

Los Angeles

Query-Efficient Black-box Adversarial Attacks

A thesis submitted in partial satisfaction
of the requirements for the degree
Master of Science in Computer Science

by

Simranjit Singh

2020

ABSTRACT OF THE THESIS

Query-Efficient Black-box
Adversarial Attacks

by

Simranjit Singh
Master of Science in Computer Science
University of California, Los Angeles, 2020
Professor Cho-Jui Hsieh, Chair

Machine learning systems have been shown to be vulnerable to adversarial examples. We study the most practical problem setup for evaluating adversarial robustness of a machine learning system with limited access: the hard-label black-box attack setting for generating adversarial examples, where limited model queries are allowed and only the decision is provided to a queried data input. Several algorithms have been proposed for this problem but they typically require huge amount (>20,000) of queries for attacking one example. Among them, one of the state-of-the-art approaches (Cheng et al., 2019) showed that hard-label attack can be modeled as an optimization problem where the objective function can be evaluated by binary search with additional model queries, thereby a zeroth order optimization algorithm can be applied. In this thesis, we adopt the same optimization formulation but propose to directly estimate the sign of gradient at any direction instead of the gradient itself, which enjoys the benefit of single query. Using this single query oracle for retrieving sign of directional derivative, we develop a novel query-efficient Sign-OPT approach for hard-label black-box attack. We provide a convergence analysis of the new algorithm and conduct experiments on several models on MNIST, CIFAR-10 and ImageNet. We find that Sign-OPT attack consistently requires 5X to 10X fewer queries when compared to the current state-of-the-art approaches and usually converges to an adversarial example with smaller perturbation.

The thesis of Simranjit Singh is approved.

Kai-Wei Chang

Quanquan Gu

Cho-Jui Hsieh, Committee Chair

University of California, Los Angeles

2020

To my parents . . .

who—despite many hardships—

saw to it that I grew up in a

positive environment

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

I wish to express my sincere gratitude towards all the people who played a role in my academic accomplishments, some of whom deserve a special mention.

I cannot begin to express my thanks to my advisor, Professor Cho-Jui Hsieh, for providing me with this research opportunity and guiding me throughout. The work presented in this thesis is a joint effort published as [12]. I would like to thank my colleague Minhao Cheng for his significant contributions to this work. I am grateful to and want to acknowledge the contributions of my co-authors Patrick Chen, Pin-Yu Chen and Sijia Liu who made this work possible.

Most of all, I am indebted to my parents, Harcharan Singh and Satvinder Kaur, whose upbringing and unconditional support has enabled me to pursue this journey. And my sister, Ramneek Kaur, for her love and support. I express my heartfelt gratitude towards my friends, Nikhil Mahajan and Ranti Dev Sharma, for their encouragement and invaluable support during the hardest of times. I want to thank my roomies and besties, Jaspreet Arora and Siddharth Verma, who were an integral part of my time at UCLA.

Lastly, I am grateful to UCLA for providing me with a wonderful time during my Masters.

CHAPTER 1

Introduction

It has been shown that neural networks are vulnerable to adversarial examples [29, 15, 7, 3]. As an example, a neural network like Resnet-50 [16] might correctly classify an image of a Panda, but when the pixels are perturbed slightly, the network might classify it as a Gibbon [15]. These perturbations are ubiquitous in many machine learning systems and are so small that these are imperceptible to humans and the algorithms to find such perturbations are called adversarial attacks. Given a victim neural network model and a correctly classified example, an adversarial attack aims to compute a small perturbation such that with this perturbation added, the original example will be misclassified.

Adversarial examples are a threat to the security of machine learning models and hence have gained a lot of attention in the past years. An attacker can fool an autonomous driving system by changing a traffic sign slightly, for example changing a stop sign with stickers and paints so that it is classified as 45 mph sign but still looks like a stop sign to a human, and can cause serious damages. Adversarial examples can be developed against voice-based agents which appear like noise to humans but can give specific commands to the model. A miscreant can try to sell banned substances on an e-commerce platform by fooling the machine learning based security systems. Hence, evaluating robustness of models and developing defense strategies against adversarial attacks becomes very important before these models are deployed in real environment.

Many adversarial attacks have been proposed in the literature. Most of them consider the white-box setting, where the attacker has full knowledge about the victim model. In this setting, the gradient of the objective function with respect to the input can be computed

by back-propagation and perturbations can be crafted easily. This is not realistic setting but does provide a way to evaluate the robustness of the model. Popular Examples include C&W [7] and PGD [25] attacks.

On the other hand, some more recent attacks have considered the probability black-box or score-based setting where the attacker does not know the victim model's structure and weights, but can iteratively query the model and get the corresponding probability output or the logits. In this setting, although gradient (of output probability to the input layer) is not computable, it can still be estimated using finite differences. Each directional derivative estimation only requires one or two queries and these estimate of gradient can be used to device adversarial attack. Algorithms based on this finite-difference estimator include [10, 17, 30, 19].

In this thesis, we consider the most challenging and practical attack setting – hard-label black-box setting – where the model is hidden to the attacker and the attacker can only make queries and get the corresponding hard-label decisions (e.g., predicted labels) of the model. A commonly used algorithm proposed in this setting, also called Boundary attack [5], is based on random walks on the decision surface, but it does not have any convergence guarantee. More recently, [11] showed that finding the minimum adversarial perturbation in the hard-label setting can be reformulated as another optimization problem (we call this Cheng's formulation in this thesis). This new formulation is based on evaluating the distance of decision boundary in a particular direction and finding a direction that has minimum distance to the boundary. It enjoys the benefit of having a smooth boundary in most tasks and the function value is computable using hard-label queries via binary search. Therefore, the authors of [11] are able to use standard zeroth order optimization to solve the new formulation. Although their algorithm converges quickly, it still requires large number of queries (e.g., 20,000 for a CIFAR-10 image) for attacking a single image since every function evaluation of Cheng's formulation has to be computed using binary search requiring tens of queries.

In this thesis, we follow the same optimization formulation of [11] which has the advantage of smoothness. Here, instead of using finite differences to estimate the magnitude of directional

derivative, which requires tens of queries, we propose to evaluate its sign with just **a single query**. With this single-query sign oracle, we design novel algorithms for solving the Cheng's formulation, and we theoretically prove and empirically demonstrate the significant reduction in the number of queries required for hard-label black box attack.

The contributions of the thesis are summarized below:

- We elucidate an efficient approach to compute the sign of directional derivative of Cheng's formulation using a single query, and based on this technique we develop a novel optimization algorithm called Sign-OPT for hard-label black-box attack.
- Our novel optimization method can be viewed as a new zeroth order optimization algorithm that features fast convergence of signSGD [4]. Instead of directly taking the sign of gradient estimation, our algorithm utilizes the scale of random direction. This make existing analysis inappropriate to our case, and we provide a new recipe to prove the convergence of this new optimizer. We give a formal theoretical analysis showing that our algorithm, although using a non-standard update rule, converges to a stationary point with convergence rate similar to previous zeroth order methods.
- We conduct comprehensive experiments on several datasets and models. We show that the proposed algorithm consistently reduces the query count by 5–10 times across different models and datasets, suggesting a practical and query-efficient robustness evaluation tool. Furthermore, on most datasets our algorithm can find an adversarial example with smaller distortion compared with previous approaches.

CHAPTER 2

Background

2.1 White-box attacks

In white-box setting, the attacker has complete access to the model including architecture, weights etc. This means that for neural networks, under this assumption, back-propagation can be conducted on the target model because both network structure and weights are known by the attacker. This setting is not applicable for most real scenarios since, internals of a machine learning model are rarely exposed. Nonetheless, this setting provides important insights into the model's robustness as well as feature representation. Many white-box attacks are used in defence strategies like the adversarial training wherein adversarial examples are generated for the model during training and included in the training dataset.

Since it was firstly found that neural networks are easy to be fooled by adversarial examples [15], a lot of work has been proposed in the white-box attack setting, where the classifier f is completely exposed to the attacker. One of the first attacks, Fast Gradient Sign Method (FGSM) introduced in [15] is based on calculating the gradient of the loss function with respect to the input and adding it to the input aiming at maximizing the loss and hence misclassifying the resulting example. Given a correctly classified example, $\boldsymbol{x}_0$, with label y_0 and a model with parameters $\boldsymbol{\theta}$ and a loss function J the adversarial example can be calculated as

$$\boldsymbol{x}_{adv} = \boldsymbol{x}_0 + \epsilon * \text{sign}(\nabla_x J(\boldsymbol{\theta}, \boldsymbol{x}_0, y_0))$$

where ϵ is a small multiplier to ensure the perturbation is imperceptible. This is illustrated in Figure 2.1. I-FGSM [21] extends this idea and conducts FGSM iteratively to achieve a smaller distortion and a higher success rate.

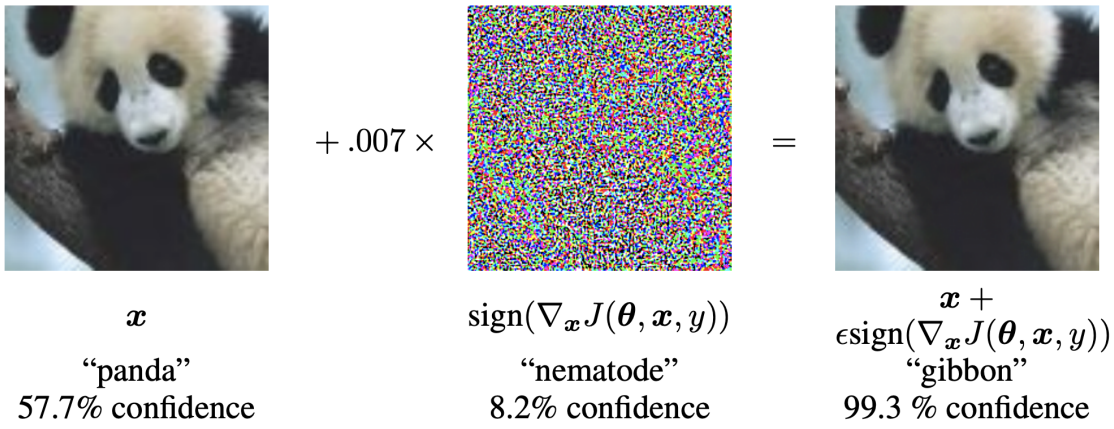


Figure 2.1: Illustratoin of the FGSM attack [15]

The C&W [7] attack is based on optimizing an objective function that is a combination of the distance from original input and the loss function of the classifier. It can be formulated as

$$\boldsymbol{x}_{adv} = \underset{\boldsymbol{x}}{\text{argmin}}\{\mathcal{D}(\boldsymbol{x}_0, \boldsymbol{x}) + c \cdot \mathcal{L}(Z(\boldsymbol{x}))\}$$

where $\mathcal{D}(\cdot, \cdot)$ is a distance measure (e.g. l_2, l_∞, etc.), $\mathcal{L}$ is some loss function and $Z(\boldsymbol{x})$ is the final (logit) layer output. For an untargeted attack, $\mathcal{L}$ can be

$$\mathcal{L}(Z(\boldsymbol{x})) = \max\{[Z(\boldsymbol{x})]_{y_0} - \max_{i \neq y_0}[Z(\boldsymbol{x})]_i, -\kappa\}$$

where $[\cdot]_i$ denotes the i^{th} component in the vector. EAD [9] further uses the elastic net to combine the l_2 and l_1 penalty.

Projected Gradient Descent (PGD) [25] is another famous white-box attack. It is an iterative method that tries to find an adversarial example using the following update:

$$\boldsymbol{x}_k = \Pi_{B_p(\boldsymbol{x}, \epsilon)}(\boldsymbol{x}_{k-1} + \eta \boldsymbol{s}_l)$$

$$\boldsymbol{s}_l = \Pi_{\partial B_p(0,1)} \nabla_x J(\boldsymbol{\theta}, \boldsymbol{x}_k, y_0)$$

where $\boldsymbol{x}_k$ is the adversarial output at k^{th} step, Π_S is the projection onto the set S, $B_p(\boldsymbol{x}', \epsilon')$ is an l_p ball of radius ϵ' around $\boldsymbol{x}'$, η is the step size, ∂U is the boundary of set U [18]. Intuitively, PGD update perturbs input in the direction of increasing loss and the projection ensures that at every step, perturbation is within a given distance from original input.

Recently, the BPDA attack introduced by [3] bypasses some models with obfuscated gradients and is shown to successfully circumvent many defenses. In addition to typical attacks based on small ℓ_p norm perturbation, non-ℓ_p norm perturbations such as scaling or shifting have also been considered [31].

2.2 Black-box attacks

Recently, black-box setting is drawing rapidly increasing attention. In black-box setting, the attacker can query the model but has no (direct) access to any internal information inside the model. Depending on the model's feedback for a given query, an attack can be classified as a soft-label or hard-label attack. In the soft-label setting, the model outputs a probability score for each decision while in hard-label attack only decision labels are available as output from the model.

The black-box attacks can be broadly divided into two kinds - **transfer-based** and **query-based**. The former uses a substitute model to devise attacks on target model while later simply relies on queries to the target model to generate adversarial examples.

2.2.1 Transfer-based attacks

A very common strategy in this setting is to train a substitute model and then use it to devise adversarial examples. This model acts as a representative substitute of the target model and then white-box attacks can be used to craft adversarial examples. This was first introduced in [28] where the attacker generates a synthetic dataset of examples which are labeled using black-box queries to the target model. The attacker then trains a substitute model on this dataset. Since the substitute model is a representative of the target model in terms of classification and features, the attacks on substitute model tend to be highly transferable on target model. However there are disadvantages to this method, major one being the need of a training dataset. Also overall, the query efficiency using substitute models tends to be worse than query-based methods [18].

2.2.2 Soft-label attacks

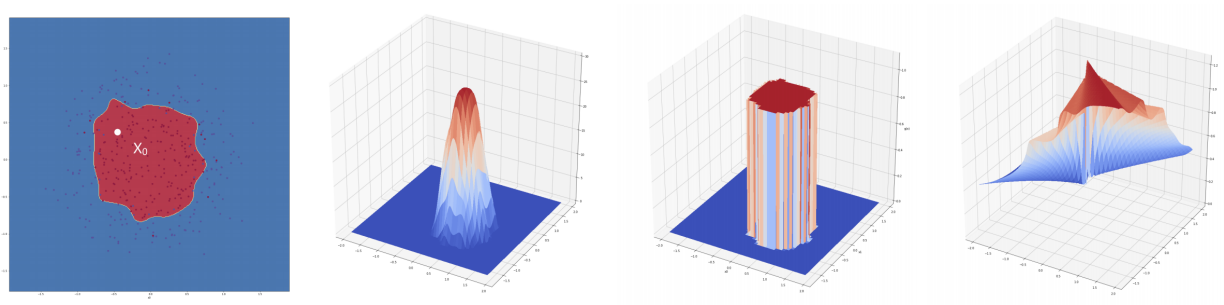
In the soft-label setting, the model outputs a probability score for each decision. [10] uses a finite difference in a coordinate-wise manner to approximately estimate the output probability changes and does a coordinate descent to conduct the attack. [17] uses Neural evolution strategy (NES) to approximately estimate the gradient directly. Later, some variants [18, 30] were proposed to utilize the side information to further speed up the attack procedure. [2] uses a evolutionary algorithm as a black-box optimizer for the soft-label setting. Recently, [1] proposes SignHunter algorithm based on signSGD [4] to achieve faster convergence in the soft-label setting. The recent work [1] proposes SignHunter algorithm to achieve a more query-efficent sign estimate when crafting black-box adversarial examples through soft-label information.

2.2.3 Hard-label attacks

In the hard-label case, only the final decision, i.e. the top-1 predicted class, is observed. As a result, the attacker can only make queries to acquire the corresponding hard-label decision instead of the probability outputs.

2.2.3.1 Difficulty of hard-label attacks

Hard-label black-box attacks are very challenging due to the limited information available to the attacker. The direct extension of any white-box or gradient-based attacks does not apply since the objective function is a discontinuous step function. This is illustrated in Figure 2.2 taken from [11] which shows the extension of these methods onto a simple 3-layer neural network. Figure 2.2 (a) shows a decision boundary learned by the network. We see that the loss function on the logits layer ($Z(\boldsymbol{x})$) is continuous as shown in Figure 2.2 (b). Zeroth-order methods can therefore be applied to optimize such an objective funtion. Figure 2.2 (c) shows the loss function if only final decisions are available (hard-label setting). It is clear that the objective functions is a step function and no first-order or zeroth-order methods can be



(a) $f(\boldsymbol{x})$ (b) $\mathcal{L}(Z(\boldsymbol{x}))$ (c) $\mathcal{L}(f(\boldsymbol{x}))$ (d) $g(\boldsymbol{\theta})$

Figure 2.2: Difficulty of hard-label attack [11].

applied to optimize such a function.

2.2.3.2 Cheng's reformulation

Cheng et. al in [11] reformulated this optimization problem to make the objective function continuous. The authors formulate it into a problem that finds a direction which could produce the shortest distance to decision boundary. We refer to [11] for the explanation of the reformulation in this section.

For a given example $\boldsymbol{x}_0$, true label y_0 and the hard-label black-box function $f : \mathbb{R}^d \to \{1, \ldots, K\}$, we define our objective function $g : \mathbb{R}^d \to \mathbb{R}$ depending on the type of attack:

$$\textbf{Untargeted attack:} \quad g(\boldsymbol{\theta}) = \min_{\lambda>0} \lambda \quad \texttt{s.t} \quad f(\boldsymbol{x}_0 + \lambda \frac{\boldsymbol{\theta}}{||\boldsymbol{\theta}||}) \neq y_0 \tag{2.1}$$

$$\textbf{Targeted attack:} \quad g(\boldsymbol{\theta}) = \min_{\lambda>0} \lambda \quad \texttt{s.t} \quad f(\boldsymbol{x}_0 + \lambda \frac{\boldsymbol{\theta}}{||\boldsymbol{\theta}||}) = t \tag{2.2}$$

wher t in Equation 2.2 is a given target class. In this formulation, $\boldsymbol{\theta}$ represents the search direction and $g(\boldsymbol{\theta})$ is the distance from $\boldsymbol{x}_0$ to the nearest adversarial example along the direction $\boldsymbol{\theta}$. Equation (2.1) and (2.2) correspond to Untargeted and Targeted attacks respectively. For untargeted attack, $g(\boldsymbol{\theta})$ also corresponds to the distance to the decision boundary along the direction $\boldsymbol{\theta}$. For our example, Figure 2.2 (d) shows the $g(\boldsymbol{\theta})$ corresponding to the neural network. We can see that it is a continuous function and such is the case for most of the applications.

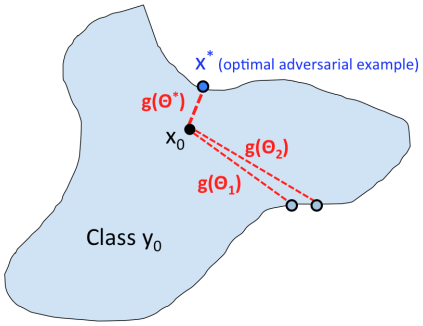


Figure 2.3: Cheng's formulation of adversarial attack [11]

Instead of searching for an adversarial example, we search the direction $\boldsymbol{\theta}$ to minimize the distortion $g(\boldsymbol{\theta})$, which leads to the following optimization problem:

$$\min_{\boldsymbol{\theta}} \; g(\boldsymbol{\theta}). \tag{2.3}$$

Finally, the adversarial example can be found by $\boldsymbol{x}^* = \boldsymbol{x}_0 + g(\boldsymbol{\theta}^*)\frac{\boldsymbol{\theta}^*}{\|\boldsymbol{\theta}^*\|}$, where $\boldsymbol{\theta}^*$ is the optimal solution of (2.3). This is illustrated in Figure 2.3

2.2.3.3 Other Hard-label attacks

[5] first studied this problem and proposed an algorithm based on random walks near the decision boundary. By selecting a random direction and projecting it onto a boundary sphere in each iteration, it aims to generate a high-quality adversarial example. Query-Limited attack [17] tries to estimate the output probability scores with model query and turn the hard-label into a soft-label problem.

The recent arXiv paper [8] applied the zeroth-order sign oracle to improve Boundary attack, and also demonstrated significant improvement. The major differences to our algorithm are that we propose a new zeroth-order gradient descent algorithm, provide its algorithmic convergence guarantees, and aim to improve the query complexity of the attack formulation proposed in [11]. For completeness, we also compare with this method in Section 5.5. Moreover, [8] uses one-point gradient estimate, which is unbiased but may encounter larger variance compared with the gradient estimate in our work. Thus, we can observe in Section

5.5 that although they are slightly faster in the initial stage, Sign-OPT will catch up and eventually lead to a slightly better solution.

CHAPTER 3

Sign-OPT

We follow the same formulation explained in subsubsection 2.2.3.2 and consider the hard-label attack as the problem of finding the direction with shortest distance to the decision boundary. Specifically, for a given example $\boldsymbol{x}_0$, true label y_0 and the hard-label black-box function $f : \mathbb{R}^d \to \{1, \ldots, K\}$, the objective function $g : \mathbb{R}^d \to \mathbb{R}$ (for the untargeted attack) can be written as:

$$\min_{\boldsymbol{\theta}} g(\boldsymbol{\theta}) \quad \text{where} \quad g(\boldsymbol{\theta}) = \arg\min_{\lambda>0} \left(f(x_0 + \lambda \frac{\boldsymbol{\theta}}{\|\boldsymbol{\theta}\|}) \neq y_0 \right). \tag{3.1}$$

It has been shown that this objective function is usually smooth and the objective function g can be evaluated by a binary search procedure locally. At each binary search step, we query the function $f(\boldsymbol{x}_0 + \lambda \frac{\boldsymbol{\theta}}{\|\boldsymbol{\theta}\|})$ and determine whether the distance to decision boundary in the direction $\boldsymbol{\theta}$ is greater or smaller than λ based on the hard-label prediction[1].

As the objective function is computable, the directional derivative of g can be estimated by finite differences:

$$\hat{\nabla} g(\boldsymbol{\theta}; \mathbf{u}) := \frac{g(\boldsymbol{\theta} + \epsilon \mathbf{u}) - g(\boldsymbol{\theta})}{\epsilon} \mathbf{u} \tag{3.2}$$

where $\boldsymbol{u}$ is a random Gaussian vector and $\epsilon > 0$ is a very small smoothing parameter. This is a standard zeroth order oracle for estimating directional derivative and based on this we can apply many different zeroth order optimization algorithms to minimize g. For example, [11] used the Random Derivative Free algorithm [27] to solve problem (3.1). However, each computation of (3.2) requires many hard-label queries due to binary search, so [11] still requires a huge number of queries despite having fast convergence.

[1]Note that binary search only works in a small local region; in more general case $g(\boldsymbol{\theta})$ has to be computed by a fine-grained search plus binary search, as discussed in [11].

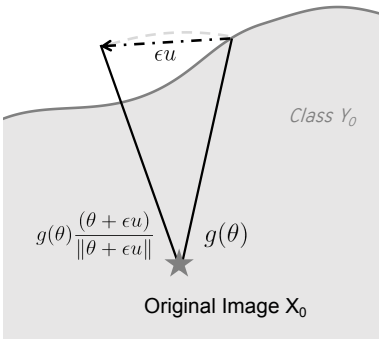


Figure 3.1: Illustration of Single-Query oracle

In this work, we introduce an algorithm that hugely improves the query complexity over [11]. Our algorithm is based on the following key ideas: (i) one does not need very accurate values of directional derivative in order to make the algorithm converge, and (ii) there exists an **imperfect but informative estimation** of directional derivative of g that can be computed by a single query.

3.1 A Single-Query oracle

As mentioned before, the previous approach requires computing $g(\boldsymbol{\theta} + \epsilon \boldsymbol{u}) - g(\boldsymbol{\theta})$ which consumes a lot of queries. However, based on the definition of $g(\cdot)$, we can compute the sign of this value $\text{sign}(g(\boldsymbol{\theta} + \epsilon \boldsymbol{u}) - g(\boldsymbol{\theta}))$ using a single query. Considering the untargeted attack case, the sign can be computed by

$$\text{sign}(g(\boldsymbol{\theta} + \epsilon \mathbf{u}) - g(\boldsymbol{\theta})) = \begin{cases} +1, & f(x_0 + g(\boldsymbol{\theta})\frac{(\boldsymbol{\theta}+\epsilon \mathbf{u})}{\|\boldsymbol{\theta}+\epsilon \mathbf{u}\|}) = y_0, \\ -1, & \text{Otherwise.} \end{cases} \tag{3.3}$$

This is illustrated in Figure 3.1. Essentially, for a new direction $\boldsymbol{\theta} + \epsilon \mathbf{u}$, we test whether a point at the original distance $g(\boldsymbol{\theta})$ from x_0 in this direction lies inside or outside the decision boundary, i.e. if the produced perturbation will result in a wrong prediction by classifier. If the produced perturbation is outside the boundary i.e. $f(x_0 + g(\boldsymbol{\theta})\frac{(\boldsymbol{\theta}+\epsilon \mathbf{u})}{\|\boldsymbol{\theta}+\epsilon \mathbf{u}\|}) \neq y_0$, the new direction has a smaller distance to decision boundary, and thus giving a smaller value of g. It indicates that $\mathbf{u}$ is a descent direction to minimize g.

Algorithm 1: Sign-OPT attack

Input: Hard-label model f, original image $\boldsymbol{x}_0$, initial $\boldsymbol{\theta}_0$;

for $t = 1, 2, \ldots, T$ **do**

Randomly sample $\mathbf{u}_1, \ldots, \mathbf{u}_Q$ from a Gaussian or Uniform distribution;

Compute $\hat{\boldsymbol{g}} \leftarrow \frac{1}{Q}\sum_{q=1}^{Q} \text{sign}(g(\boldsymbol{\theta}_t + \epsilon \mathbf{u}_q) - g(\boldsymbol{\theta}_t)) \cdot \mathbf{u}_q$;

Update $\boldsymbol{\theta}_{t+1} \leftarrow \boldsymbol{\theta}_t - \eta \hat{\boldsymbol{g}}$;

Evaluate $g(\boldsymbol{\theta}_{t+1})$ using the same search algorithm in [11] ;

end

3.2 Sign-OPT attack

By sampling random Gaussian vector Q times, we can estimate the imperfect gradient by

$$\hat{\nabla} g(\boldsymbol{\theta}) \approx \hat{\boldsymbol{g}} := \sum\nolimits_{q=1}^{Q} \text{sign}(g(\boldsymbol{\theta} + \epsilon \mathbf{u}_q) - g(\boldsymbol{\theta}))\mathbf{u}_q, \tag{3.4}$$

which only requires Q queries. We then use this imperfect gradient estimate to update our search direction $\boldsymbol{\theta}$ as $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \eta \hat{\mathbf{g}}$ with a step size η and use the same search procedure to compute $g(\boldsymbol{\theta})$ up to a certain accuracy. The detailed procedure is shown in algorithm 1. Note that in our implementation the step size η is selected for each iteration using exactly the line search procedure in the implementation provided by [11].

We note that [23] designed a Zeroth Order SignSGD algorithm for soft-label black box attack (not hard-label setting). They use $\hat{\nabla} g(\boldsymbol{\theta}) \approx \hat{\boldsymbol{g}} := \sum_{q=1}^{Q} \text{sign}(g(\boldsymbol{\theta} + \epsilon \mathbf{u}_q) - g(\boldsymbol{\theta})\mathbf{u}_q)$ and shows that it could achieve a comparable or even better convergence rate than zeroth order stochastic gradient descent by using only sign information of gradient estimation. Although it is possible to combine ZO-SignSGD with our proposed single query oracle for solving hard-label attack, their estimator will take sign of the whole vector and thus ignore the direction of $\mathbf{u}_q$, which leads to slower convergence in practice (please refer to section 5.4 and Figure 5.4 (c) for more details). In the next section we talk about other estimates of the gradients.

3.3 Other gradient estimations

We observe that the value $\text{sign}(g(\boldsymbol{\theta}+\epsilon\mathbf{u})-g(\boldsymbol{\theta}))$ computed by our single query oracle is actually the sign of the directional derivative:

$$\text{sign}(\langle\nabla g(\boldsymbol{\theta}),\mathbf{u}\rangle)=\text{sign}(\lim_{\epsilon\to\infty}\frac{g(\boldsymbol{\theta}+\epsilon\mathbf{u})-g(\boldsymbol{\theta})}{\epsilon})=\text{sign}(g(\boldsymbol{\theta}+\epsilon\mathbf{u})-g(\boldsymbol{\theta}))\text{ for a small }\epsilon.$$

Therefore, we can use this information to estimate the original gradient. Let $y_q := \text{sign}(\langle\nabla g(\boldsymbol{\theta}),\mathbf{u}_q\rangle)$, a more accurate gradient estimation can be cast as the following constraint optimization problem:

$$\text{Find a vector } \boldsymbol{z} \text{ such that } \text{sign}(\langle\boldsymbol{z},\mathbf{u}_q\rangle)=y_q \quad \forall q=1,\dots,Q.$$

Therefore, this is equivalent to a hard constraint SVM problem where each $\mathbf{u}_q$ is a training sample and y_q is the corresponding label. The gradient can then be recovered by solving the following quadratic programming problem:

$$\min_{\boldsymbol{z}} \ \boldsymbol{z}^T\boldsymbol{z} \quad \text{s.t.} \quad \boldsymbol{z}^T\mathbf{u}_q\geq y_q, \quad \forall q=1,\dots,Q. \tag{3.5}$$

By solving this problem, we can get a good estimation of the gradient. As explained earlier, each y_q can be determined with a single query. Therefore, we propose a variant of Sign-OPT, which is called SVM-OPT attack. The detailed procedure is shown in algorithm 2. We will present an empirical comparison of our two algorithms in section 5.1.

Algorithm 2: SVM-OPT attack

Input: Hard-label model f, original image $\boldsymbol{x}_0$, initial $\boldsymbol{\theta}_0$;

for $t=1,2,\dots,T$ **do**

- Sample $\mathbf{u}_1,\dots,\mathbf{u}_Q$ from Gaussian or orthogonal basis ;
- Solve $\boldsymbol{z}$ defined by (3.5) ;
- Update $\boldsymbol{\theta}_{t+1}\leftarrow\boldsymbol{\theta}_t-\eta\boldsymbol{z}$;
- Evaluate $g(\boldsymbol{\theta}_{t+1})$ using search algorithm in [11] ;

end

CHAPTER 4

Theoretical Results

To the best of our knowledge, no previous analysis can be used to prove convergence of Algorithm 1. In the following, we show that Algorithm 1 can in fact converge and furthermore, with similar convergence rate compared with [23] despite using a different gradient estimator.

Assumption 1. *Function $g(\theta)$ is L-smooth with a finite value of L.*

Assumption 2. *At any iteration step t, the gradient of the function g is upper bounded by $\|\nabla g(\boldsymbol{\theta}_t)\|_2 \leq \sigma$.*

Theorem 4.0.1. *Suppose that the conditions in the assumptions hold, and the distribution of gradient noise is unimodal and symmetric. Then, Sign-OPT attack with learning rate $\eta_t = O(\frac{1}{Q\sqrt{dT}})$ and $\epsilon = O(\frac{1}{dT})$ will give following bound on $\mathbb{E}[\|\nabla g(\boldsymbol{\theta})\|_2]$:*

$$\mathbb{E}[\|\nabla g(\boldsymbol{\theta})\|_2] = O(\frac{\sqrt{d}}{\sqrt{T}} + \frac{d}{\sqrt{Q}}).$$

The proof is provided below. The main difference with the original analysis provided by [23] is that they only only deal with sign of each element, while our analysis also takes the magnitudes of each element of $\boldsymbol{u}_q$ into account.
Define following notations:

$$\hat{\nabla} g(\boldsymbol{\theta}_t; \boldsymbol{u}_q) \coloneqq \text{sign}(g(\boldsymbol{\theta}_t + \epsilon \boldsymbol{u}_q) - g(\boldsymbol{\theta}_t))\boldsymbol{u}_q$$
$$\dot{\nabla} g(\boldsymbol{\theta}_t; \boldsymbol{u}_q) \coloneqq \frac{1}{\epsilon}(g(\boldsymbol{\theta}_t + \epsilon \boldsymbol{u}_q) - g(\boldsymbol{\theta}_t))\boldsymbol{u}_q$$
$$\bar{\nabla} g(\boldsymbol{\theta}_t; \boldsymbol{u}_q) \coloneqq \text{sign}(\frac{1}{\epsilon}(g(\boldsymbol{\theta}_t + \epsilon \boldsymbol{u}_q) - g(\boldsymbol{\theta}_t))\boldsymbol{u}_q)$$

Thus we could write the corresponding estimate of gradients as follow:

$$\hat{\boldsymbol{g}}_t = \frac{1}{Q}\sum_{q=1}^{Q} \text{sign}(g(\boldsymbol{\theta}_t + \epsilon \mathbf{u}_q) - g(\boldsymbol{\theta}_t))\boldsymbol{u}_q = \frac{1}{Q}\sum_{q=1}^{Q} \hat{\nabla} g(\boldsymbol{\theta}_t; \boldsymbol{u}_q)$$

$$\dot{\boldsymbol{g}}_t = \frac{1}{Q}\sum_{q=1}^{Q} \frac{1}{\epsilon}(g(\boldsymbol{\theta}_t + \epsilon \boldsymbol{u}_q) - g(\boldsymbol{\theta}_t))\boldsymbol{u}_q = \frac{1}{Q}\sum_{q=1}^{Q} \dot{\nabla} g(\boldsymbol{\theta}_t; \boldsymbol{u}_q)$$

$$\bar{\boldsymbol{g}}_t = \frac{1}{Q}\sum_{q=1}^{Q} \text{sign}(\frac{1}{\epsilon}(g(\boldsymbol{\theta}_t + \epsilon \boldsymbol{u}_q) - g(\boldsymbol{\theta}_t))\boldsymbol{u}_q) = \frac{1}{Q}\sum_{q=1}^{Q} \bar{\nabla} g(\boldsymbol{\theta}_t; \boldsymbol{u}_q)$$

Clearly, we have $\bar{\nabla} g(\boldsymbol{\theta}_t; \boldsymbol{u}_q) = \text{sign}(\dot{\nabla} g(\boldsymbol{\theta}_t; \boldsymbol{u}_q))$ and we can relate $\bar{\nabla} g(\boldsymbol{\theta}_t; \boldsymbol{u}_q)$ and $\hat{\nabla} g(\boldsymbol{\theta}_t; \boldsymbol{u}_q)$ by writing $\hat{\nabla} g(\boldsymbol{\theta}_t; \boldsymbol{u}_q) = G_q \odot \bar{\nabla} g(\boldsymbol{\theta}_t; \boldsymbol{u}_q)$ where $G_q \in \mathbb{R}^d$ is absolute value of vector $\boldsymbol{u}_q$ i.e.

$$G_q = (|\boldsymbol{u}_{q,1}|, |\boldsymbol{u}_{q,2}|, \cdots, |\boldsymbol{u}_{q,d}|)^T$$

Note that Zeroth-order gradient estimate $\dot{\nabla} g(\boldsymbol{\theta}_t; \boldsymbol{u}_q)$ is a biased approximation to the true gradient of g. Instead, it becomes unbiased to the gradient of the randomized smoothing function $g_\epsilon(\boldsymbol{\theta}) = \mathbb{E}_{\boldsymbol{u}}[g(\boldsymbol{\theta} + \epsilon \boldsymbol{u})]$ [14].

To prove the convergence of proposed method, we need the information on variance of the update $\dot{\nabla} g(\boldsymbol{\theta}_t; \boldsymbol{u}_q)$. Here, we introduce a lemma from previous works.

Lemma 4.0.2. *The variance of Zeroth-Order gradient estimate $\dot{\nabla} g(\boldsymbol{\theta}_t; \boldsymbol{u}_q)$ is upper bounded by*

$$\mathbb{E}\big[\|\dot{\nabla} g(\boldsymbol{\theta}_t; \boldsymbol{u}_q) - \nabla g_\epsilon(\boldsymbol{\theta}_t)\|_2^2\big] \leq \frac{4(Q+1)}{Q}\sigma^2 + \frac{2}{Q}C(d, \epsilon),$$

where $C(d, \epsilon) \coloneqq 2d\sigma^2 + \epsilon^2 L^2 d^2/2$

Proof. This lemma could be proved by using proposition 2 in [23] with b = 1 and q = Q. When b = 1 there is no difference between with/without replacement, and we opt for with replacement case to obtain above bound. □

By talking Q = 1, we know that $\mathbb{E}\big[\|\dot{\nabla} g(\boldsymbol{\theta}_t; \boldsymbol{u}_q) - \nabla g_\epsilon(\boldsymbol{\theta}_t)\|_2^2\big]$ is upper bounded. And by Jensen's inequality, we also know that the

$$\mathbb{E}\big[|(\dot{\nabla} g(\boldsymbol{\theta}_t; \boldsymbol{u}_q) - \nabla g_\epsilon(\boldsymbol{\theta}_t))_l|\big] \leq \sqrt{\mathbb{E}\big[((\dot{\nabla} g(\boldsymbol{\theta}_t; \boldsymbol{u}_q) - \nabla g_\epsilon(\boldsymbol{\theta}_t)))_l^2\big]} \coloneqq \delta_l, \tag{4.1}$$

where δ_l denotes the upper bound of lth coordinate of $\mathbb{E}\big[|\dot{\nabla} g(\boldsymbol{\theta}_t; \boldsymbol{u}_q) - \nabla g_\epsilon(\boldsymbol{\theta}_t)|\big]$, and δ_l is finite since $\mathbb{E}\big[\|\dot{\nabla} g(\boldsymbol{\theta}_t; \boldsymbol{u}_q) - \nabla g_\epsilon(\boldsymbol{\theta}_t)\|_2^2\big]$ is upper bounded.

Next, we want to show the $\mathbb{P}[\text{sign}((\bar{\boldsymbol{g}}_t)_l) \neq \text{sign}((\nabla g_\epsilon(\boldsymbol{\theta}_t))_l)]$ by following lemma.

Lemma 4.0.3. $|(\nabla g_\epsilon(\boldsymbol{\theta}_t))_l| \cdot \mathbb{P}[sign((\bar{\boldsymbol{g}}_t)_l) \neq sign((\nabla g_\epsilon(\boldsymbol{\theta}_t))_l)] \leq \frac{\delta_l}{\sqrt{Q}}$

Proof. Similar to [4], we first relax $\mathbb{P}[\text{sign}((\dot{\nabla} g(\boldsymbol{\theta}_t; \boldsymbol{u}_q))_l) \neq \text{sign}(\nabla g_\epsilon(\boldsymbol{\theta}_t))_l]$ by Markov inequality:

$$\begin{aligned}
\mathbb{P}[\text{sign}((\dot{\nabla} g(\boldsymbol{\theta}_t; \boldsymbol{u}_q))_l) \neq \text{sign}((\nabla g_\epsilon(\boldsymbol{\theta}_t))_l)] &\leq \mathbb{P}[|\dot{\nabla} g(\boldsymbol{\theta}_t; \boldsymbol{u}_q)_l)| \geq |\nabla g_\epsilon(\boldsymbol{\theta}_t)_l|] \\
&\leq \frac{\mathbb{E}\big[|(\dot{\nabla} g(\boldsymbol{\theta}_t; \boldsymbol{u}_q) - \nabla g_\epsilon(\boldsymbol{\theta}_t))_l|\big]}{|\nabla g_\epsilon(\boldsymbol{\theta}_t)_l|} \\
&\leq \frac{\delta_l}{|\nabla g_\epsilon(\boldsymbol{\theta}_t)_l|},
\end{aligned}$$

where the last inequality comes from eq (4.1).

Recall that $(\dot{\nabla} g(\boldsymbol{\theta}_t; \boldsymbol{u}_q))_l)$ is an unbiased estimation to $(\nabla g_\epsilon(\boldsymbol{\theta}_t))_l$. Under the assumption that the noise distribution is unimodal and symmetric, from [4] Lemma D1, we will have

$$\mathbb{P}[\text{sign}((\dot{\nabla} g(\boldsymbol{\theta}_t; \boldsymbol{u}_q))_l) \neq \text{sign}(\nabla g_\epsilon(\boldsymbol{\theta}_t))_l] \coloneqq M \leq \begin{cases} \frac{2}{9}\frac{1}{S^2}, & \text{S } \geq \frac{2}{\sqrt{3}} \\ \frac{1}{2} - \frac{S}{2\sqrt{3}}, & otherwise \end{cases} < \frac{1}{2},$$

where $S \coloneqq |\nabla g_\epsilon(\boldsymbol{\theta}_t)_l| / \delta_l$.

Note that this probability bound applies uniformly to all $q \in Q$ regardless of the magnitude $|(\boldsymbol{u}_q)_l|$. That is,

$$\begin{aligned}
&\mathbb{P}[\text{sign}(\sum_{q=1}^{Q} |(\boldsymbol{u}_q)_l| \text{sign}((\dot{\nabla} g(\boldsymbol{\theta}_t; \boldsymbol{u}_q))_l) \neq \text{sign}(\nabla g_\epsilon(\boldsymbol{\theta}_t))_l] = \\
&\qquad \mathbb{P}[\text{sign}((\sum_{q=1}^{Q} \text{sign}(\dot{\nabla} g(\boldsymbol{\theta}_t; \boldsymbol{u}_q))_l) \neq \text{sign}(\nabla g_\epsilon(\boldsymbol{\theta}_t))_l]. \qquad (4.2)
\end{aligned}$$

This is true as when all $|(\boldsymbol{u}_q)_l| = 1$, $\mathbb{P}[\text{sign}((\sum_{q=1}^{Q} \text{sign}(\dot{\nabla} g(\boldsymbol{\theta}_t; \boldsymbol{u}_q))_l) \neq \text{sign}(\nabla g_\epsilon(\boldsymbol{\theta}_t))_l]$ is equivalent to majority voting of each estimate q yielding correct sign. This is the same as sum of Q bernoulli trials (i.e. binomial distribution) with error rate M. And since error probability

M is independent of sampling of $|(\boldsymbol{u}_q)_l|$, calculating $\mathbb{P}[\text{sign}(\sum_{q=1}^{Q}|(\boldsymbol{u}_q)_l|\text{sign}((\dot{\nabla}g(\boldsymbol{\theta}_t;\boldsymbol{u}_q))_l) \neq \text{sign}(\nabla g_\epsilon(\boldsymbol{\theta}_t))_l]$ could be thought as taking Q bernoulli experiments and then independently draw a weight from unit length for each of Q experiment. Since the weight is uniform, we will have expectation of weights on correct counts and incorrect counts are the same and equal to 1/2. Therefore, the probability of $\mathbb{P}[\text{sign}(\sum_{q=1}^{Q}|(\boldsymbol{u}_q)_l|\text{sign}((\dot{\nabla}g(\boldsymbol{\theta}_t;\boldsymbol{u}_q))_l) \neq \text{sign}(\nabla g_\epsilon(\boldsymbol{\theta}_t))_l]$ is still the same as original non-weighted binomial distribution. Notice that by our notation, we will have $\text{sign}(\dot{\nabla}g(\boldsymbol{\theta}_t;\boldsymbol{u}_q)_l) = \bar{\nabla}g(\boldsymbol{\theta}_t;\boldsymbol{u}_q)_l$ thus $\frac{1}{Q}\sum_{q=1}^{Q}\text{sign}(\dot{\nabla}g(\boldsymbol{\theta}_t;\boldsymbol{u}_q))_l = (\bar{\boldsymbol{g}}_t)_l$. Let Z counts the number of estimates $\dot{\nabla}g(\boldsymbol{\theta}_t;\boldsymbol{u}_q)_l$ yielding correct sign of $\nabla g_\epsilon(\boldsymbol{\theta}_t)_l$. Probability in eq (4.2) could be written as:

$$\mathbb{P}[\text{sign}(\text{sign}((\bar{\boldsymbol{g}}_t)_l) \neq \text{sign}(\nabla g_\epsilon(\boldsymbol{\theta}_t))_l] = \mathbb{P}[Z \leq \frac{Q}{2}].$$

Following the derivation of theorem 2b in [4], we could get

$$\begin{aligned} &\mathbb{P}[Z \leq \frac{Q}{2}] \leq \frac{1}{\sqrt{Q}S} \\ \Rightarrow\ &|(\nabla g_\epsilon(\boldsymbol{\theta}_t))_l| \cdot \mathbb{P}[\text{sign}((\bar{\boldsymbol{g}}_t)_l) \neq \text{sign}((\nabla g_\epsilon(\boldsymbol{\theta}_t))_l)] \leq \frac{\delta_l}{\sqrt{Q}} \end{aligned} \tag{4.3}$$

□

We also need few more lemmas on properties of function g.

Lemma 4.0.4. $g_\epsilon(\boldsymbol{\theta}_1) - g_\epsilon(\boldsymbol{\theta}_T) \leq g_\epsilon(\boldsymbol{\theta}_1) - g^* + \epsilon^2 L$

Proof. The proof can be found in [24] Lemma C. □

Lemma 4.0.5. $\mathbb{E}[\|\nabla g(\boldsymbol{\theta})\|_2] \leq \sqrt{2}\mathbb{E}[\|\nabla g_\epsilon(\boldsymbol{\theta})\|_2] + \frac{\epsilon L d}{\sqrt{2}}$, *where* $g^* = \min_{\boldsymbol{\theta}} g(\boldsymbol{\theta})$.

Proof. The proof can be found in [23]. □

Proof of Theorem 4.0.1 From L-smoothness assumption we could have

$$
\begin{aligned}
g_\epsilon(\boldsymbol{\theta}_{t+1}) &\leq g_\epsilon(\boldsymbol{\theta}_t) + \langle \nabla g_\epsilon(\boldsymbol{\theta}_t), \boldsymbol{\theta}_{t+1} - \boldsymbol{\theta}_t \rangle + \frac{L}{2}\|\boldsymbol{\theta}_{t+1} - \boldsymbol{\theta}_t\|_2^2 \\
&= g_\epsilon(\boldsymbol{\theta}_t) - \eta_k \langle \nabla g_\epsilon(\boldsymbol{\theta}_t), \hat{\boldsymbol{g}}_t \rangle + \frac{L}{2}\eta_t^2 \|\hat{\boldsymbol{g}}_t\|_2^2 \\
&= g_\epsilon(\boldsymbol{\theta}_t) - \eta_t \odot \bar{G}_t \|\nabla g_\epsilon(\boldsymbol{\theta}_t)\|_1 + \frac{dL}{2}\eta_t^2 \odot \bar{G}_t{}^2 \\
&+ 2\eta_t \odot \bar{G}_t \sum_{l=1}^{d} |(\nabla g_\epsilon(\boldsymbol{\theta}_t))_l| \mathbb{P}[\mathrm{sign}((\bar{\boldsymbol{g}}_t)_l) \neq \mathrm{sign}((\nabla g_\epsilon(\boldsymbol{\theta}_t))_l)]
\end{aligned}
$$

where $\bar{G}_t$ is defined as $(\bar{G}_t)_l = \sum_{q=1}^{Q} (G_q)_l \bar{\nabla} g(\boldsymbol{\theta}_t; \boldsymbol{u}_q)_l = \sum_{q=1}^{Q} |(\boldsymbol{u}_q)_l| \bar{\nabla} g(\boldsymbol{\theta}_t; \boldsymbol{u}_q)_l$. Continuing the inequality,

$$
\begin{aligned}
& g_\epsilon(\boldsymbol{\theta}_t) - \eta_t \odot \bar{G}_t \|\nabla g_\epsilon(\boldsymbol{\theta}_t)\|_1 + \frac{dL}{2}\eta_t^2 \odot \bar{G}_t{}^2 \\
& + 2\eta_t \odot \bar{G}_t \sum_{l=1}^{d} |(\nabla g_\epsilon(\boldsymbol{\theta}_t))_l| \mathbb{P}[\mathrm{sign}((\bar{\boldsymbol{g}}_t)_l) \neq \mathrm{sign}((\nabla g_\epsilon(\boldsymbol{\theta}_t))_l)] \\
& \leq g_\epsilon(\boldsymbol{\theta}_t) - \eta_t \odot \bar{G}_t \|\nabla g_\epsilon(\boldsymbol{\theta}_t)\|_1 + \frac{dL}{2}\eta_t^2 \odot \bar{G}_t{}^2 + 2\eta_t \odot \bar{G}_t \sum_{l=1}^{d} \frac{\delta_l}{\sqrt{Q}} && \text{using (4.3)} \\
& \leq g_\epsilon(\boldsymbol{\theta}_t) - \eta_t \odot \bar{G}_t \|\nabla g_\epsilon(\boldsymbol{\theta}_t)\|_1 + \frac{dL}{2}\eta_t^2 \odot \bar{G}_t{}^2 + 2\eta_t \odot \bar{G}_t \frac{\|\delta_l\|_1}{\sqrt{Q}} \\
& \leq g_\epsilon(\boldsymbol{\theta}_t) - \eta_t \odot \bar{G}_t \|\nabla g_\epsilon(\boldsymbol{\theta}_t)\|_1 + \frac{dL}{2}\eta_t^2 \odot \bar{G}_t{}^2 + 2\eta_t \odot \bar{G}_t \frac{\sqrt{d}\sqrt{\|\delta_l\|_2^2}}{\sqrt{Q}} \\
& = g_\epsilon(\boldsymbol{\theta}_t) - \eta_t \odot \bar{G}_t \|\nabla g_\epsilon(\boldsymbol{\theta}_t)\|_1 + \frac{dL}{2}\eta_t^2 \odot \bar{G}_t{}^2 \\
& + 2\eta_t \odot \bar{G}_t \frac{\sqrt{d}\sqrt{\mathbb{E}\big[((\dot{\bar{\nabla}} g(\boldsymbol{\theta}_t; \boldsymbol{u}_q) - \nabla g_\epsilon(\boldsymbol{\theta}_t)))_l^2\big]}}{\sqrt{Q}} && \text{using(4.1).}
\end{aligned}
$$

Thus we will have,

$$
\begin{aligned}
g_\epsilon(\boldsymbol{\theta}_{t+1}) - g_\epsilon(\boldsymbol{\theta}_t) &\leq -\eta_t \odot \bar{G}_t \|\nabla g_\epsilon(\boldsymbol{\theta}_t)\|_1 + \frac{dL}{2}{\eta_t^2} \odot {\bar{G}_t}^2 \\
&\quad + 2\eta_t \odot \bar{G}_t \frac{\sqrt{d}\sqrt{\mathbb{E}\big[((\dot{\nabla} g(\boldsymbol{\theta}_t; \boldsymbol{u}_q) - \nabla g_\epsilon(\boldsymbol{\theta}_t))_l^2\big]}}{\sqrt{Q}} \\
\Rightarrow \eta_t \odot \bar{G}_t \|\nabla g_\epsilon(\boldsymbol{\theta}_t)\|_1 &\leq g_\epsilon(\boldsymbol{\theta}_t) - g_\epsilon(\boldsymbol{\theta}_{t+1}) + \frac{dL}{2}{\eta_t^2} \odot {\bar{G}_t}^2 \\
&\quad + 2\eta_t \odot \bar{G}_t \frac{\sqrt{d}\sqrt{\mathbb{E}\big[((\dot{\nabla} g(\boldsymbol{\theta}_t; \boldsymbol{u}_q) - \nabla g_\epsilon(\boldsymbol{\theta}_t))_l^2\big]}}{\sqrt{Q}} \\
\Rightarrow \hat{\eta}_t \|\nabla g_\epsilon(\boldsymbol{\theta}_t)\|_1 &\leq g_\epsilon(\boldsymbol{\theta}_t) - g_\epsilon(\boldsymbol{\theta}_{t+1}) + \frac{dL}{2}{\hat{\eta}_t}^2 + 2\hat{\eta}_t\sqrt{d}\frac{\sqrt{\mathbb{E}\big[((\dot{\nabla} g(\boldsymbol{\theta}_t; \boldsymbol{u}_q) - \nabla g_\epsilon(\boldsymbol{\theta}_t))_l^2\big]}}{\sqrt{Q}}
\end{aligned}
$$

where we define $\hat{\eta}_t \coloneqq \eta_t \odot \bar{G}_t$. Sum up all inequalities for all ts and take expectation on both side, we will have

$$
\begin{aligned}
\sum_{t=1}^{T} \hat{\eta}_t \mathbb{E}[\|\nabla g_\epsilon(\boldsymbol{\theta}_t)\|_1] &\leq \mathbb{E}[g_\epsilon(\boldsymbol{\theta}_1) - g_\epsilon(\boldsymbol{\theta}_T)] + \frac{dL}{2}\sum_{t=1}^{T}{\hat{\eta}_t}^2 \\
&\quad + \sum_{t=1}^{T} 2\hat{\eta}_t\sqrt{d}\sqrt{\mathbb{E}\big[((\dot{\nabla} g(\boldsymbol{\theta}_t; \boldsymbol{u}_q) - \nabla g_\epsilon(\boldsymbol{\theta}_t))_l^2\big]} \\
&\leq \mathbb{E}[g_\epsilon(\boldsymbol{\theta}_1) - g_\epsilon(\boldsymbol{\theta}_T)] + \frac{dL}{2}\sum_{t=1}^{T}{\hat{\eta}_t}^2 \\
&\quad + \sum_{t=1}^{T} 2\hat{\eta}_t\sqrt{d}\sqrt{\frac{4(Q+1)}{Q}\sigma^2 + \frac{2}{Q}C(d,\epsilon)} \qquad \text{using Lemma 1.}
\end{aligned}
$$

Substitute Lemma 3 into above inequality, we get

$$
\begin{aligned}
\sum_{t=1}^{T} \hat{\eta}_t \mathbb{E}[\|\nabla g_\epsilon(\boldsymbol{\theta}_t)\|_1] &\leq g_\epsilon(\boldsymbol{\theta}_1) - g^* + \epsilon^2 L + \frac{dL}{2}\sum_{t=1}^{T}{\hat{\eta}_t}^2 \\
&\quad + \sum_{t=1}^{T} 2\hat{\eta}_t\sqrt{d}\sqrt{\frac{4(Q+1)}{Q}\sigma^2 + \frac{2}{Q}C(d,\epsilon)}.
\end{aligned}
$$

Since $\|\cdot\|_2 \leq \|\cdot\|_1$ and we can divide $\sum_{t=1}^{T}\hat{\eta}_t$ on both side to get

$$\sum_{t=1}^{T} \frac{\hat{\eta}_t}{\sum_{t=1}^{T} \hat{\eta}_t} \mathbb{E}[\|\nabla g_\epsilon(\boldsymbol{\theta}_t)\|_2] \leq \frac{g_\epsilon(\boldsymbol{\theta}_1) - g^* + \epsilon^2 L}{\sum_{t=1}^{T} \hat{\eta}_t} + \frac{dL}{2} \frac{\sum_{t=1}^{T} \hat{\eta}_t{}^2}{\sum_{t=1}^{T} \hat{\eta}_t} + \sum_{t=1}^{T} \frac{2\sqrt{d}}{\sqrt{Q}} \sqrt{4(Q+1)\sigma^2 + 2C(d,\epsilon)}.$$

Define a new random variable R with probability $\mathbb{P}[R = t] = \frac{\eta_t}{\sum_{t=1}^{T} \eta_t}$, we will have

$$\mathbb{E}[\|\nabla g_\epsilon(\boldsymbol{\theta}_R)\|_2] = \mathbb{E}[\mathbb{E}_R[\|\nabla g_\epsilon(\boldsymbol{\theta}_R)\|_2]] = \mathbb{E}\Big[\sum_{t=1}^{T} \mathbb{P}[R = t] \|\nabla g_\epsilon(\boldsymbol{\theta}_t)\|_2\Big].$$

Substitute all the quantities into Lemma 4, we will get

$$\mathbb{E}[\|\nabla g(\boldsymbol{\theta})\|_2] \leq \frac{\sqrt{2}(g_\epsilon(\boldsymbol{\theta}_1) - g^* + \epsilon^2 L)}{\sum_{t=1}^{T} \hat{\eta}_t} + \frac{dL}{\sqrt{2}} \frac{\sum_{t=1}^{T} \hat{\eta}_t{}^2}{\sum_{t=1}^{T} \hat{\eta}_t} + \frac{\epsilon L d}{\sqrt{2}} + \sum_{t=1}^{T} \frac{2\sqrt{2}\sqrt{d}}{\sqrt{Q}} \sqrt{4(Q+1)\sigma^2 + 2C(d,\epsilon)}.$$

By choosing $\epsilon = O(\frac{1}{dT})$ and $\eta_t = O(\frac{1}{Q\sqrt{dT}})$, then the convergence rate as shown in above is $O(\frac{d}{T} + \frac{d}{\sqrt{Q}})$. □

CHAPTER 5

Experimental Results

We evaluate the Sign-OPT algorithm for attacking black-box models in a hard-label setting on three different standard datasets - MNIST [22], CIFAR-10 [20] and ImageNet-1000 [13] and compare it with existing methods. For fair and easy comparison, we use the CNN networks provided by [7], which have also been used by other previous hard-label attacks as well. Specifically, for both MNIST and CIFAR-10, the model consists of nine layers in total - four convolutional layers, two max-pooling layers and two fully-connected layers. Further details about implementation, training and parameters are available on [7]. As reported in [7] and [11], we were able to achieve an accuracy of 99.5% on MNIST and 82.5% on CIFAR-10. We use the pretrained Resnet-50 [16] network provided by torchvision [26] for ImageNet-1000, which achieves a Top-1 accuracy of 76.15%.

In our experiments, we found that Sign-OPT and SVM-OPT perform quite similarly in terms of query efficiency. Hence we compare only Sign-OPT attack with previous approaches and provide a comparison between Sign-OPT and SVM-OPT in section 5.1. We compare the following attacks:

- **Sign-OPT attack** (black-box): The approach presented in this thesis.
- **Opt-based attack** (black-box): The method proposed in [11] where they use Randomized Gradient-Free method to optimize the same objective function. We use the implementation provided at `https://github.com/LeMinhThong/blackbox-attack`.
- **Boundary attack** (black-box): The method proposed in [5]. This is compared only in L_2 setting as it is designed for the same. We use the implementation provided in Foolbox (`https://github.com/bethgelab/foolbox`).

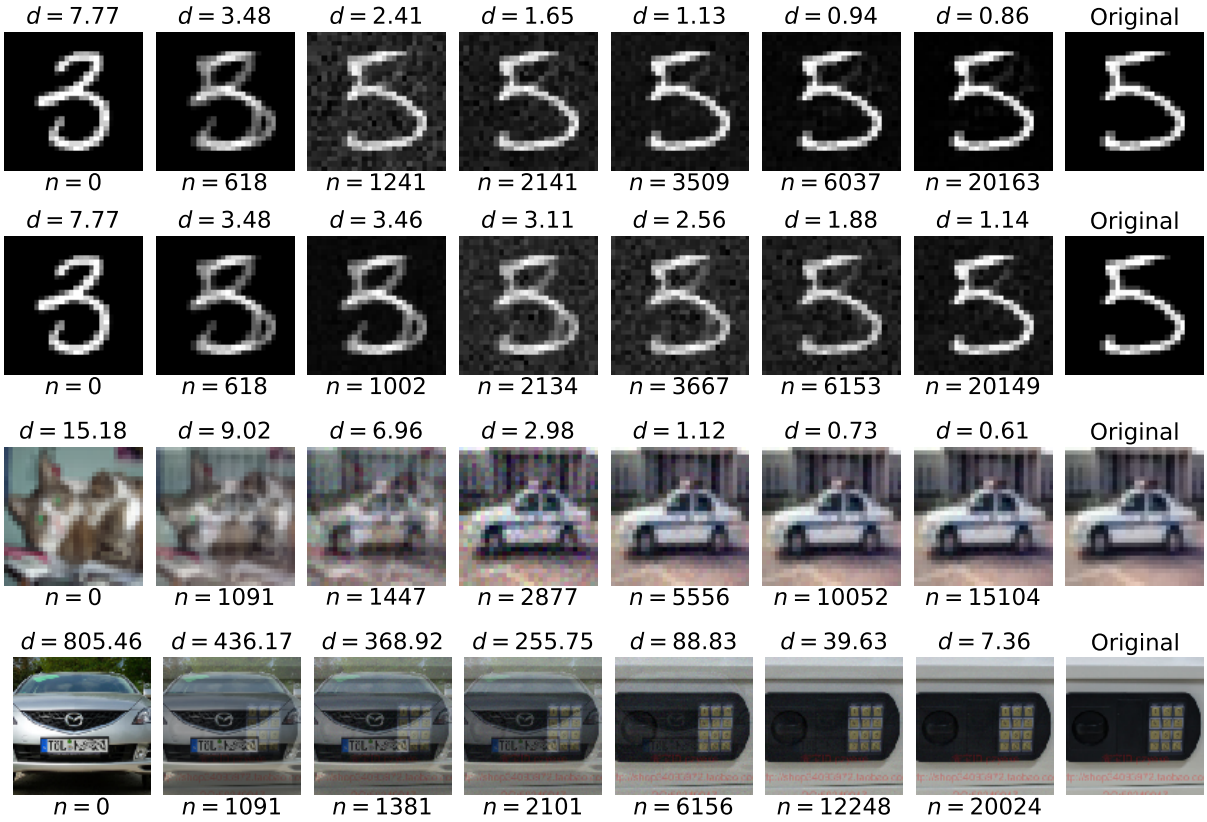


Figure 5.1: Example of Sign-OPT targeted attack. L_2 distortions and queries used are shown above and below the images. First two rows: Example comparison of Sign-OPT attack and OPT attack. Third and fourth rows: Examples of Sign-OPT attack on CIFAR-10 and ImageNet

- **Guessing Smart Attack** (black-box): The method proposed in [6]. This attack enhances boundary attack by biasing sampling towards three priors. Note that one of the priors assumes access to a similar model as the target model and for a fair comparison we do not incorporate this bias in our experiments. We use the implementation provided at `https://github.com/ttbrunner/biased_boundary_attack`.

- **C&W attack** (white-box): One of the most popular methods in the white-box setting proposed in [7]. We use C&W L_2 norm attack as a baseline for the white-box attack performance.

For each attack, we randomly sample 100 examples from validation set and generate adversarial perturbations for them. For untargeted attack, we only consider examples that are correctly predicted by model and for targeted attack, we consider examples that are already not predicted as target label by the model. To compare different methods, we mainly use *median distortion* as the metric. Median distortion for x queries is the median adversarial perturbation of all examples achieved by a method using less than x queries. Since all the hard-label attack algorithms will start from an adversarial exmample and keep reduce the distortion, if we stop at any time they will always give an adversarial example and medium distortion will be the most suitable metric to compare their performance. Besides, we also show *success rate (SR)* for x queries for a given threshold (ϵ), which is the percentage of number of examples that have achieved an adversarial perturbation below ϵ with less than x queries. We evaluate success rate on different thresholds which depend on the dataset being used. For comparison of different algorithms in each setting, we chose the same set of examples across all attacks.

Implementation details[1]: To optimize algorithm 1, we estimate the step size η using the same line search procedure implemented in [11]. At the cost of a relatively small number of queries, this provides significant speedup in the optimization. Similar to [11], $g(\boldsymbol{\theta})$ in last step of algorithm 1 is approximated via binary search. The initial $\boldsymbol{\theta}_0$ in algorithm 1 is

[1]Code available at https://github.com/cmhcbb/attackbox

calculated by evaluating $g(\boldsymbol{\theta})$ on 100 random directions and taking the best one.

5.1 Comparison between Sign-OPT and SVM-OPT

In our experiments, we found that the performance in terms of queries of both these attacks is remarkably similar in all settings (both L_2/L_∞ & Targeted/Untargeted) and datasets. We present a comparison for MNIST and CIFAR-10 (L_2 norm-based) for both targeted and untargeted attacks in Figure 5.2. We see that the median distortion achieved for a given number of queries is quite on part for both Sign-OPT and SVM-OPT.

Number of queries per gradient estimate: In Figure 5.2, we show the comparison of Sign-OPT attack with different values of Q. Our experiments suggest that Q does not have an impact on the convergence point reached by the algorithm. Although, small values of Q provide a noisy gradient estimate and hence delayed convergence to an adversarial perturbation. Large values of Q, on the other hand, require large amount of time per gradient estimate. After fine tuning on a small set of examples, we found that $Q = 200$ provides a good balance between the two. Hence, we set the value of $Q = 200$ for all our experiments in this section.

5.2 Untargeted attack

In this attack, the objective is to generate an adversary from an original image for which the prediction by model is different from that of original image. Figure 5.3 provides an elaborate comparison of different attacks for L_2 case for the three datasets. Sign-OPT attack consistently outperforms the current approaches in terms of queries. Not only is Sign-OPT more efficient in terms of queries, in most cases it converges to a lower distortion than what is possible by other hard-label attacks. Furthermore, we observe Sign-OPT converges to a solution comparable with C&W white-box attack (better on CIFAR-10, worse on MNIST, comparable on ImageNet). This is significant for a hard-label attack algorithm since we are given very limited information.

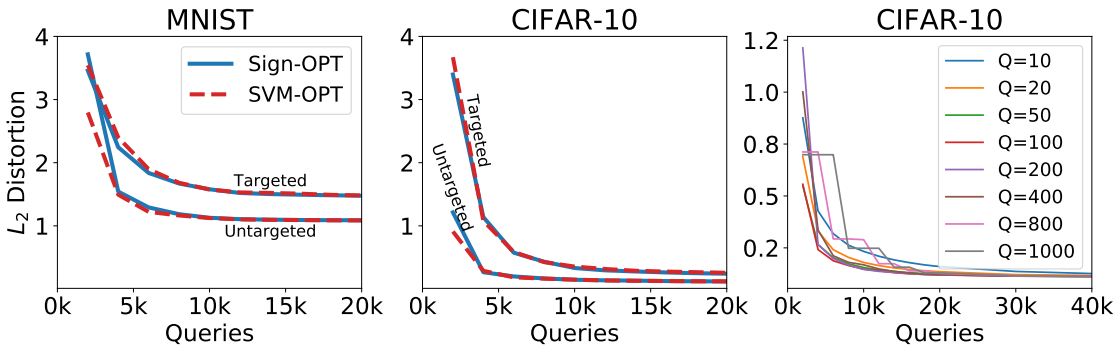


Figure 5.2: Median L_2 distortion vs Queries. First two: Comparison of Sign-OPT and SVM-OPT attack for MNIST and CIFAR-10. Third: Performance of Sign-OPT for different values of Q.

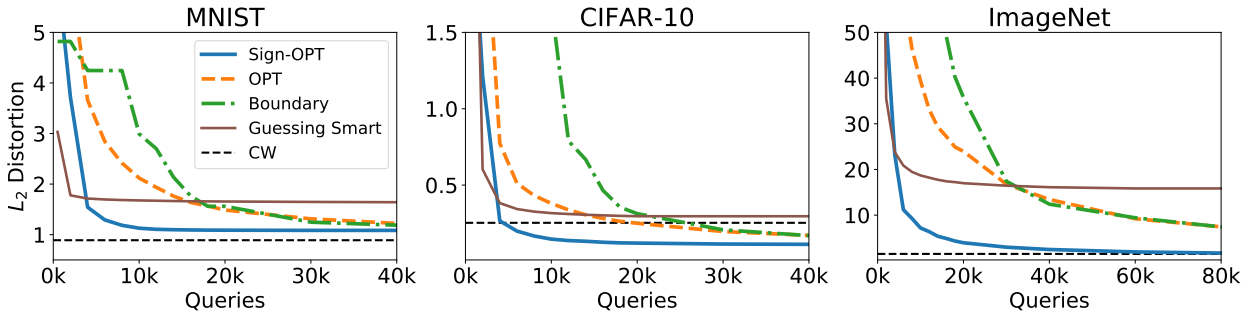


Figure 5.3: Untargeted attack: Median distortion vs Queries for different datasets.

We highlight some of the comparisons of Boundary attack, OPT-based attack and Sign-OPT attack (L_2 norm-based) in Table 5.1. Particularly for ImageNet dataset on ResNet-50 model, Sign-OPT attack reaches a median distortion below 3.0 in less than $30k$ queries while other attacks need more than $200k$ queries for the same.

5.3 Targeted attack

In targeted attack, the goal is to generate an adversarial perturbation for an image so that the prediction of resulting image is the same as a specified target. For each example, we randomly specify the target label, keeping it consistent across different attacks. We calculate the initial θ_0 in algorithm 1 using 100 samples in target label class from training dataset and

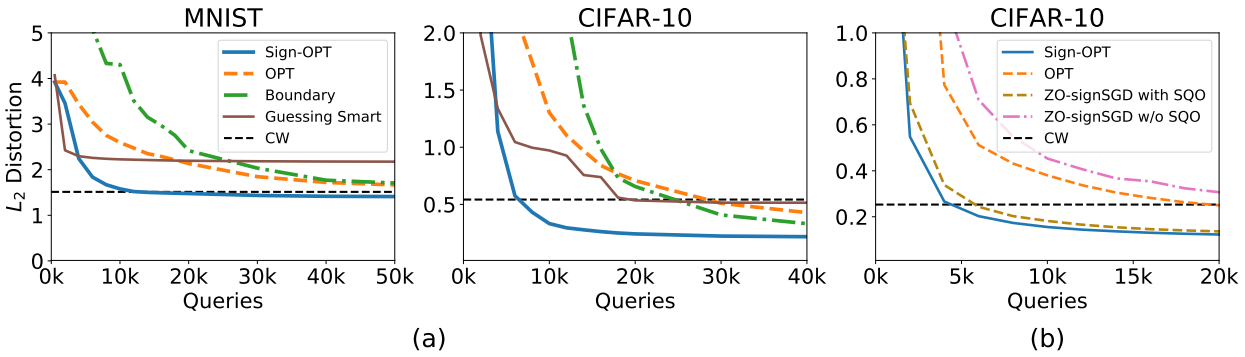


Figure 5.4: (a) Targeted Attack: Median distortion vs Queries of different attacks on MNIST and CIFAR-10. (b) Comparing Sign-OPT and ZO-SignSGD with and without single query oracle (SQO).

this θ_0 is the same across different attacks. Figure 5.1 shows some examples of adversarial examples generated by Sign-OPT attack and the Opt-based attack. The first two rows show comparison of Sign-OPT and Opt attack respectively on an example from MNIST dataset. The figures show adversarial examples generated at almost same number of queries for both attacks. Sign-OPT method generates an L_2 adversarial perturbation of 0.94 in $\sim 6k$ queries for this particular example while Opt-based attack requires $\sim 35k$ for the same. Figure 5.4 displays a comparison among different attacks in targeted setting. In our experiments, average distortion achieved by white box attack C&W for MNIST dataset is 1.51, for which Sign-OPT requires $\sim 12k$ queries while others need $> 120k$ queries. We present a comparison of success rate of different attacks for CIFAR-10 dataset in Figure 5.5 for both targeted and untargeted cases.

5.4 The power of single query oracle

In this section, we conduct several experiments to prove the effectiveness of our proposed single query oracle in hard-label adversarial attack setting. ZO-SignSGD algorithm [23] is proposed for soft-label black box attack and we extend it into hard-label setting. A straightforward way is simply applying ZO-SignSGD to solve the hard-label objective proposed in [11], estimate

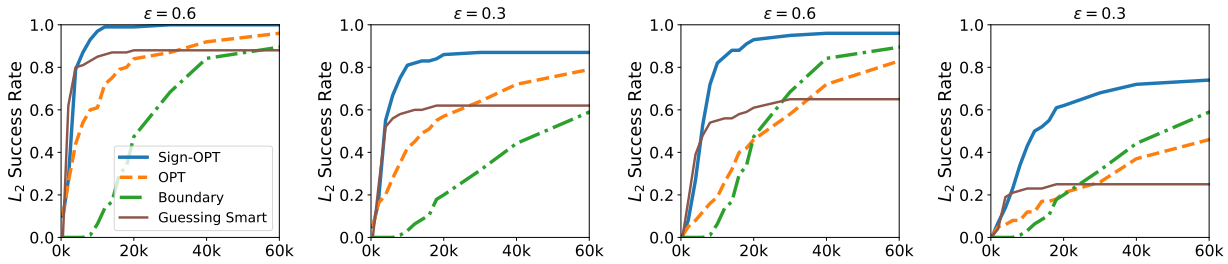


Figure 5.5: Success Rate vs Queries for CIFAR-10 (L_2 norm-based attack). First two and last two depict untargeted and targeted attacks respectively. Success rate threshold is at the top of each plot.

the gradient using binary search as [11] and take its sign. In Figure 5.4 (b), we clearly observe that simply combining ZO-SignSGD and [11] is not efficient. With the proposed single query sign oracle, we can also reduce the query count of this method, as demonstrated in Figure 5.4 (b). This verifies the effectiveness of single query oracle, which can universally improve many different optimization methods in the hard-label attack setting. To be noted, there is still improvement on Sign-OPT over ZO-SignSGD with single query oracle because instead of directly taking the sign of gradient estimation, our algorithm utilizes the scale of random direction u as well. In other words, signSGD's gradient norm is always 1 while our gradient norm takes into account the magnitude of u. Therefore, our signOPT optimization algorithm is fundamentally different [23] or any other proposed signSGD varieties. Our method can be viewed as a new zeroth order optimization algorithm that features fast convergence in signSGD.

5.5 Comparison with HopSkipJumpAttack

There is a recent paper [8] that applied the zeroth-order sign oracle to improve Boundary attack, and also demonstrated significant improvement. The major differences to our algorithm are that we propose a new zeroth-order gradient descent algorithm, provide its algorithmic convergence guarantees, and aim to improve the query complexity of the attack formulation proposed in [11]. To be noted, HopSkipJumpAttack only provides the bias and variance

analysis (Theorem 2 and 3) without convergence rate analysis.

Also, HopSkipJumpAttack uses one-point gradient estimate compared to the 2-point gradient estimate used by Sign-OPT. Therefore, although the estimation is unbiased, it has large variance, which achieves successful attack faster but generates a worse adversarial example with larger distortion than ours. For completeness, we also compare with this method (and mention the results) as follows.

Figure 5.6 shows a comparison of Sign-OPT and HopSkipJumpAttack for CIFAR-10 and MNIST datasets for the case of L_2 norm based attack. We find in our experiments that performance of both attacks is comparable in terms of queries consumed. In some cases, Sign-OPT converges to a better solution.

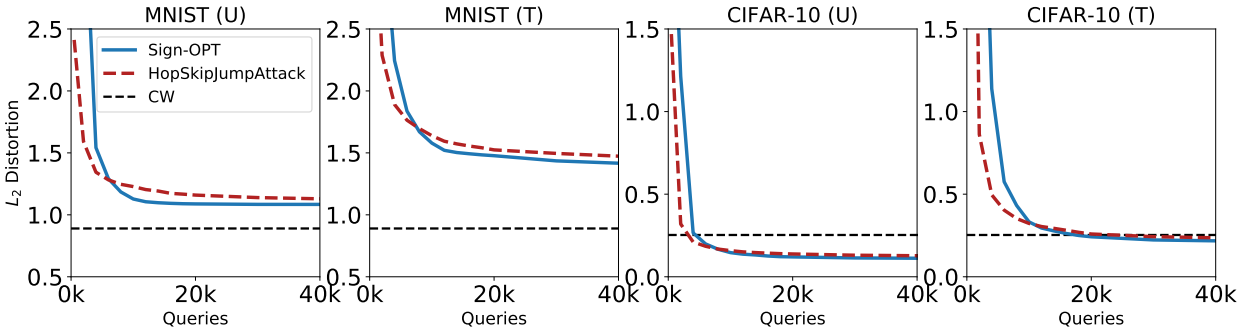


Figure 5.6: Comparison with HopSkipJumpAttack for CIFAR and MNIST: Median distortion vs Queries. (U) represents untargeted attack and (T) represents targeted attack.

5.6 Conclusion

We developed a new and ultra query-efficient algorithm for adversarial attack in the hard-label black-box setting. Using the same smooth reformulation in [11], we design a novel zeroth order oracle that can compute the sign of directional derivative of the attack objective using single query. Equipped with this single-query oracle, we design a new optimization algorithm that can dramatically reduce number of queries compared with [11]. We prove the convergence of the proposed algorithm and show our new algorithm is overwhelmingly better than current hard-label black-box attacks.

Table 5.1: L_2 Untargeted attack - Comparison of average L_2 distortion achieved using a given number of queries for different attacks. SR stands for success rate.

	MNIST			CIFAR10			ImageNet (ResNet-50)		
	#Queries	Avg L_2	SR($\epsilon = 1.5$)	#Queries	Avg L_2	SR($\epsilon = 0.5$)	#Queries	Avg L_2	SR($\epsilon = 3.0$)
Boundary attack	4,000	4.24	1.0%	4,000	3.12	2.3%	4,000	209.63	0%
	8,000	4.24	1.0%	8,000	2.84	7.6%	30,000	17.40	16.6%
	14,000	2.13	16.3%	12,000	0.78	29.2%	160,000	4.62	41.6%
OPT attack	4,000	3.65	3.0%	4,000	0.77	37.0%	4,000	83.85	2.0%
	8,000	2.41	18.0%	8,000	0.43	53.0%	30,000	16.77	14.0%
	14,000	1.76	36.0%	12,000	0.33	61.0%	160,000	4.27	34.0%
Guessing Smart	4,000	1.74	41.0%	4,000	0.29	75.0%	4,000	16.69	12.0%
	8,000	1.69	42.0%	8,000	0.25	80.0%	30,000	13.27	12.0%
	14,000	1.68	43.0%	12,000	0.24	80.0%	160,000	12.88	12.0%
Sign-OPT attack	4,000	1.54	46.0%	4,000	0.26	73.0%	4,000	23.19	8.0%
	8,000	1.18	84.0%	8,000	0.16	90.0%	30,000	2.99	50.0%
	14,000	1.09	94.0%	12,000	0.13	95.0%	160,000	1.21	90.0%
C&W (white-box)	-	0.88	99.0%	-	0.25	85.0%	-	1.51	80.0%

Table 5.2: L_∞ Untargeted attack - Comparison of average L_∞ distortion achieved using a given number of queries for different attacks. SR stands for success rate.

	MNIST			CIFAR10			ImageNet (ResNet-50)		
	Avg L_∞	# Queries	SR	Avg L_∞	# Queries	SR	Avg L_∞	# Queries	SR
OPT attack	0.4	13,414	72.5%	0.2	2,381	100.0%	2.0	3,202	94.0%
	0.15	17,650	2.1%	0.03	4,943	43.0%	0.5	10,712	54.0%
Sign-OPT attack	0.4	3,497	100.0%	0.2	1,080	100.0%	2.0	1,653	100.0%
	0.15	7,633	10.1%	0.03	5,379	70.0%	05	4,710	76.0%

BIBLIOGRAPHY

[1] Abdullah Al-Dujaili and Una-May O'Reilly. There are no bit parts for sign bits in black-box attacks. *arXiv preprint arXiv:1902.06894*, 2019.

[2] Moustafa Alzantot, Yash Sharma, Supriyo Chakraborty, Huan Zhang, Cho-Jui Hsieh, and Mani B Srivastava. Genattack: Practical black-box attacks with gradient-free optimization. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 1111–1119, 2019.

[3] Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *ICML*, 2018.

[4] Jeremy Bernstein, Yu-Xiang Wang, Kamyar Azizzadenesheli, and Animashree Anandkumar. signSGD: Compressed optimisation for non-convex problems. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 560–569, Stockholmsmässan, Stockholm Sweden, 10–15 Jul 2018. PMLR.

[5] Wieland Brendel, Jonas Rauber, and Matthias Bethge. Decision-based adversarial attacks: Reliable attacks against black-box machine learning models. *arXiv preprint arXiv:1712.04248*, 2017.

[6] Thomas Brunner, Frederik Diehl, Michael Truong Le, and Alois Knoll. Guessing smart: Biased sampling for efficient black-box adversarial attacks. *arXiv preprint arXiv:1812.09803*, 2018.

[7] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *Security and Privacy (SP), 2017 IEEE Symposium on*, pages 39–57. IEEE, 2017.

[8] Jianbo Chen, Michael I. Jordan, and Martin J. Wainwright. Hopskipjumpattack: A query-efficient decision-based attack. *arXiv preprint arXiv:1904.02144*, 2019.

[9] Pin-Yu Chen, Yash Sharma, Huan Zhang, Jinfeng Yi, and Cho-Jui Hsieh. Ead: elastic-net attacks to deep neural networks via adversarial examples. In *Thirty-second AAAI conference on artificial intelligence*, 2018.

[10] Pin-Yu Chen, Huan Zhang, Yash Sharma, Jinfeng Yi, and Cho-Jui 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*, pages 15–26. ACM, 2017.

[11] Minhao Cheng, Thong Le, Pin-Yu Chen, Huan Zhang, JinFeng Yi, and Cho-Jui Hsieh. Query-efficient hard-label black-box attack: An optimization-based approach. In *International Conference on Learning Representations*, 2019.

[12] Minhao Cheng, Simranjit Singh, Patrick Chen, Pin-Yu Chen, Sijia Liu, and Cho-Jui Hsieh. Sign-opt: A query-efficient hard-label adversarial attack. *arXiv preprint arXiv:1909.10773*, 2019.

[13] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. IEEE, 2009.

[14] John C Duchi, Peter L Bartlett, and Martin J Wainwright. Randomized smoothing for stochastic optimization. *SIAM Journal on Optimization*, 22(2):674–701, 2012.

[15] Ian Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*, 2015.

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[17] Andrew Ilyas, Logan Engstrom, Anish Athalye, and Jessy Lin. Black-box adversarial attacks with limited queries and information. In *International Conference on Machine Learning*, pages 2142–2151, 2018.

[18] Andrew Ilyas, Logan Engstrom, and Aleksander Madry. Prior convictions: Black-box adversarial attacks with bandits and priors. *arXiv preprint arXiv:1807.07978*, 2018.

[19] Kwang-Sung Jun, Lihong Li, Yuzhe Ma, and Jerry Zhu. Adversarial attacks on stochastic bandits. In *Advances in Neural Information Processing Systems*, pages 3640–3649, 2018.

[20] Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. Cifar-10 (canadian institute for advanced research).

[21] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial machine learning at scale. *arXiv preprint arXiv:1611.01236*, 2016.

[22] Yann LeCun, Léon Bottou, Yoshua Bengio, Patrick Haffner, et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[23] Sijia Liu, Pin-Yu Chen, Xiangyi Chen, and Mingyi Hong. signSGD via zeroth-order oracle. In *International Conference on Learning Representations*, 2019.

[24] Sijia Liu, Bhavya Kailkhura, Pin-Yu Chen, Paishun Ting, Shiyu Chang, and Lisa Amini. Zeroth-order stochastic variance reduction for nonconvex optimization. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 3727–3737. Curran Associates, Inc., 2018.

[25] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

[26] Sébastien Marcel and Yann Rodriguez. Torchvision the machine-vision package of torch. In *Proceedings of the 18th ACM International Conference on Multimedia*, MM '10, pages 1485–1488, New York, NY, USA, 2010. ACM.

[27] Yurii Nesterov and Vladimir Spokoiny. Random gradient-free minimization of convex functions. *Foundations of Computational Mathematics*, 17(2):527–566, 2017.

[28] Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z Berkay Celik, and Ananthram Swami. Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security*, pages 506–519. ACM, 2017.

[29] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.

[30] Chun-Chen Tu, Paishun Ting, Pin-Yu Chen, Sijia Liu, Huan Zhang, Jinfeng Yi, Cho-Jui Hsieh, and Shin-Ming Cheng. Autozoom: Autoencoder-based zeroth order optimization method for attacking black-box neural networks. *AAAI*, 2019.

[31] Huan Zhang, Hongge Chen, Zhao Song, Duane Boning, Inderjit S Dhillon, and Cho-Jui Hsieh. The limitations of adversarial training and the blind-spot attack. *arXiv preprint arXiv:1901.04684*, 2019.