Towards Query-Efficient Black-Box Adversary with Zeroth-Order Natural Gradient Descent

Pu Zhao,[1] **Pin-Yu Chen,**[2] **Siyue Wang,**[1] **Xue Lin**[1]
[1]Northeastern University, Boston, MA 02115
[2]IBM Research, Yorktown Heights, NY 10598
{zhao.pu, wang.siy}@husky.neu.edu, pin-yu.chen@ibm.com, xue.lin@northeastern.edu

Abstract

Despite the great achievements of the modern deep neural networks (DNNs), the vulnerability/robustness of state-of-the-art DNNs raises security concerns in many application domains requiring high reliability. Various adversarial attacks are proposed to sabotage the learning performance of DNN models. Among those, the black-box adversarial attack methods have received special attentions owing to their practicality and simplicity. Black-box attacks usually prefer less queries in order to maintain stealthy and low costs. However, most of the current black-box attack methods adopt the first-order gradient descent method, which may come with certain deficiencies such as relatively slow convergence and high sensitivity to hyper-parameter settings. In this paper, we propose a zeroth-order natural gradient descent (ZO-NGD) method to design the adversarial attacks, which incorporates the zeroth-order gradient estimation technique catering to the black-box attack scenario and the second-order natural gradient descent to achieve higher query efficiency. The empirical evaluations on image classification datasets demonstrate that ZO-NGD can obtain significantly lower model query complexities compared with state-of-the-art attack methods.

Introduction

Modern technologies based on machine learning (ML) and specifically deep learning (DL), have achieved significant breakthroughs (LeCun, Bengio, and Hinton 2015) in various applications. Deep neural network (DNN) serves as a fundamental component in artificial intelligence. However, despite the outstanding performance, many recent studies demonstrate that state-of-the-art DNNs in computer vision (Xie, Wang, and et. al. 2018), speech recognition (Alzantot, Balaji, and Srivastava 2018) and deep reinforcement learning (Lin, Hong, and et. al. 2017) are vulnerable to adversarial examples (Goodfellow, Shlens, and Szegedy 2015), which add carefully designed imperceptible distortions to legitimate inputs aiming to mislead the DNNs at test time. This raises concerns of the DNN robustness in many applications with high reliability and dependability requirements.

With the recent exploration of adversarial attacks in image classification and objection detection, the vulnerability/robustness of DNNs has attracted ever-increasing attentions and efforts in the research field known as *adversarial machine learning*. A large amount of efforts have been devoted to: 1) designing adversarial perturbations in various ML applications (Goodfellow, Shlens, and Szegedy 2015; Carlini and Wagner 2017; Chen et al. 2017a; Zhao et al. 2019c; Xu et al. 2018); 2) security evaluation methodologies to systematically estimate the DNN robustness (Biggio, Fumera, and Roli 2014; Zhang, Weng, and et. al. 2018); and 3) defense mechanisms against adversarial attacks (Bulò, Biggio, and et. al. 2017; Demontis, Melis, and et. al. 2018; Madry et al. 2017; Wang et al. 2019; 2018a; Xu et al. 2019). This work mainly investigates the first category to build the groundwork towards developing potential defensive measures in reliable ML.

However, most of preliminary studies on this topic focus on the white-box setting where the target DNN model is completely available to the attacker (Goodfellow, Shlens, and Szegedy 2015; Carlini and Wagner 2017; Zhao et al. 2018). More specifically, the adversary can compute the gradients of the output with respect to the input to identify the effect of perturbing certain input pixels, with complete knowledge about the DNN model's internal structure, parameters and configurations. Despite the theoretical interest, it is unrealistic to adopt the white-box adversarial methods to attack practical black-box threat models (Zhao et al. 2019b), where the internal model states/configurations are not revealed to the attacker (e.g., Google Cloud Vision API). Instead, the adversary can only query the model by submitting inputs and obtain the corresponding model outputs of prediction probabilities when generating adversarial examples.

In the black-box adversarial setting, it is often the case that the less queries, the more efficient an attack becomes. Large amount of queries may be at the risk of exposing the adversary or high financial cost in the case where the query is charged per query. Notably, to date most of the white-box (Goodfellow, Shlens, and Szegedy 2015; Carlini and Wagner 2017) and black-box attacks (Chen et al. 2017b; Ilyas et al. 2018; Zhao et al. 2019a) are based on first-order gradient descent methods. Different from the widely utilized first-order optimization, the application of second-order optimization (Martens 2016) is less explored due to the large

computation overhead, although it may achieve faster convergence rate. The work (Osawa et al. 2018) adopts natural gradient descent (NGD) to train ResNet-50 on ImageNet in 35 epochs, demonstrating its great potentiality.

In this work, inspired by the superb convergence performance of NGD, we propose zeroth-order natural gradient descent (ZO-NGD), which incorporates the zeroth-order (ZO) method and the second-order NGD, to generate black-box adversarial examples in a query-efficient manner. The contributions of this work are summarized as follows:

+ ***Design of adversary attacks with NGD:*** To the best of our knowledge, we are the first to derive the Fisher information matrix (FIM) and adopt the second-order NGD method for adversarial attacks, which is different from other first-order-based white-box and black-box attack methods.

+ ***Co-optimization of zeroth-order and second-order methods:*** In the black-box setting, we incorporate the zeroth-order random gradient estimation to estimate the gradients which is not directly available, and leverage the second-order NGD to achieve high query-efficiency.

+ ***No additional queries to obtain the FIM:*** During the queries to estimate the gradients of the loss, with our design the Fisher information is a byproduct that are extracted and evaluated without requiring additional query complexity.

+ ***Scalability to high dimensional datasets:*** In NGD, it is computationally infeasible to compute and invert the FIM with billions of elements on large scale datasets like ImageNet. To address this problem, we propose a method to avoid the computation and inverse of the FIM and thus the computation complexity is at most the same as the input images, rather than its square (the dimension of the FIM).

Related Work

In adversarial ML, the black-box setting is more practical where the attacker can only query the target model by providing input images and receive the probability density output for the input.

Black-box Attack

Attack with gradient estimation In the black-box setting, as the gradients are not directly available, gradient estimation methods via zeroth-order optimization (Wang et al. 2018b; Tu et al. 2018; Duchi et al. 2015) are proposed to estimate the gradients. The ZOO method (Chen et al. 2017b) performs pixel-level gradient estimation first and then perform white-box C&W attack (Carlini and Wagner 2017) with the estimated gradients. Despite its high success rate, it suffers from intensive computation and huge queries due to element-wise gradient estimation.

The more practical threat models are investigated in (Ilyas et al. 2018). New attack methods based on Natural Evolutionary Strategies (NES) and Monte Carlo approximation to estimate the gradients are developed to mislead ImageNet classifiers under more restrictive threat models. The work (Ilyas, Engstrom, and Madry 2018) further proposes to use the prior information including the time-dependent priors and data-dependent priors to enhance the query efficiency.

Different from the previous first-order-based methods, the work (Ye et al. 2018) exploits the second-order optimization to improve query efficiency. In general, they explore Hessian information in the parameter space while our work explores the Hessian information in the distribution space (aka information matrix). Particularly, our method obtains the Fisher information during the first-order information (gradients) estimation *for free* while the mentioned paper needs additional queries for Hessian-based second-order optimization.

Heuristic black-box attacks In the transfer attack (Papernot, McDaniel, and Goodfellow 2016), the attacker first trains a surrogate model with data labeled by the target model. White-box attacks are applied to attack the surrogate model and the generated examples are transferred to attack the target model. However, it may suffer from low attack success rate due to the low similarity between the surrogate model and the target model.

The boundary method (Brendel, Rauber, and Bethge 2017) utilizes a conceptually simple idea to decrease the distortion through random walks and find successful adversarial perturbations while staying on the misclassification boundary. However, it suffers from high computational complexity and lacks algorithmic convergence guarantees.

Second-order optimization

First-order gradient descent methods have been extensively used in various ML tasks. They are easy to implement and suitable for large-scale DL. But these methods come with well known deficiencies such as relatively-slow convergence and sensitivity to hyper-parameter settings. On the other hand, second-order optimization methods provide a elegant solution by selectively re-scaling the gradient with the curvature information (Martens 2016). As a kind of second-order method, NGD proves to be Fisher efficient by using the FIM instead of the Hessian matrix (Amari 1998; Martens 2014). But the large overhead to compute, store and invert the FIM may limit its application. To address this, Kronecker Factored Approximate Curvature (K-FAC) is proposed to train DNNs (Grosse and Martens 2016).

Problem Formulation

Threat Model: In this paper, we mainly investigate black-box adversarial attacks for image classification with DNNs. Different from the white-box setting which has fully access to the DNN model and its internal structures/parameters, the black-box setting constrains the information available to the adversary. The attacker can only query the model by providing an input image and obtain the DNN output score/probability of the input. The black-box setting is more consistent with the scenario of "machine-learning-deployed-as-a-service" like Google Cloud Vision API.

In the following, we first provide a general problem formulation for adversarial attack which can be adopted to either white-box or black-box settings. Then, an efficient solution is proposed for the black-box setting. We highlight that this method can be easily adopted to the white-box setting by using the exact gradients to achieve higher query efficiency.

Attack Model: Given a legitimate image $\mathbf{x} \in \mathbb{R}^d$ with its correct class label t, the objective is to design an optimal adversarial perturbation $\boldsymbol{\delta} \in \mathbb{R}^d$ so that the perturbed example

$(\mathbf{x}+\boldsymbol{\delta})$ can lead to a misclassification by the DNN model trained on legitimate images. The DNN model would misclassify the adversarial example to another class $t' \neq t$. $\boldsymbol{\delta}$ can be obtained by solving the following problem,

$$\underset{\boldsymbol{\delta}\in\mathbb{S}}{\text{minimize}}\, f(\mathbf{x}+\boldsymbol{\delta}, t), \tag{1}$$

where $\mathbb{S} = \{\boldsymbol{\delta}|(\mathbf{x}+\boldsymbol{\delta}) \in [0,1]^d,\ \|\boldsymbol{\delta}\|_\infty \leq \epsilon\}$ and $f(\mathbf{x}+\boldsymbol{\delta}, t)$ denotes an attack loss incurred by misclassifying $(\mathbf{x}+\boldsymbol{\delta})$ to another class t'. $\|\cdot\|_\infty$ denotes the ℓ_∞ norm. In problem (1), the constraints on $\mathbb{S}$ ensure that the perturbed noise $\boldsymbol{\delta}$ at each pixel (normalized to $[0,1]$) is imperceptible up to a predefined ϵ-tolerant threshold.

Motivated by (Carlini and Wagner 2017), the loss function $f(\boldsymbol{x}, t)$ is expressed as

$$\begin{aligned} f(\mathbf{x}+\boldsymbol{\delta}, t) \;=\; \max\{&\log p(t|\mathbf{x}+\boldsymbol{\delta}) \\ &- \max_{i\neq t}\{\log p(i|\mathbf{x}+\boldsymbol{\delta})\}, -\kappa\}, \end{aligned} \tag{2}$$

where $p(i|\mathbf{x})$ denotes the model's prediction score or probability of the i-th class for the input $\boldsymbol{x}$, and κ is a confidence parameter usually set to zero. Basically, $f(\mathbf{x}+\boldsymbol{\delta}, t)$ achieves its minimum value 0 if $p(t|\mathbf{x}+\boldsymbol{\delta})$ is smaller than $\max_{i\neq t} \log p(i|\mathbf{x}+\boldsymbol{\delta})$, indicating there is a label with higher probability than the correct label t and thus a misclassification is achieved by adding the perturbation $\boldsymbol{\delta}$ to $\boldsymbol{x}$. In this paper, we mainly investigate the untargeted attack which does not specify the target misclassified label. The targeted attack can be easily implemented following nearly the same problem formulation and loss function with slight modifications (Carlini and Wagner 2017; Ilyas, Engstrom, and Madry 2018). We focus on the general formulation here and omit the targeted attack formulation.

Note that in Eq. (2), we use the log probability $\log p(i|\mathbf{x})$ instead of $p(i|\mathbf{x})$ because the output probability distribution tends to have one dominating class. The log operator is used to reduce the effect of the dominating class while it still preserves the probability order of all classes.

As most of the white-box attack methods rely on gradient descent methods, the unavailability of the gradients in black-box settings will limit their application. Gradient estimation methods (known as zeroth-order optimization) are applied to perform the normal projected first-order gradient descent process (Ilyas et al. 2018; Liu et al. 2019) as follows,

$$\boldsymbol{\delta}_{k+1} = \prod_{\mathbb{S}} \left(\boldsymbol{\delta}_k - \lambda \hat{\nabla} f(\boldsymbol{\delta}_k)\right), \tag{3}$$

where λ is the learning rate and the $\prod_{\mathbb{S}}(\cdot)$ performs the projection onto the feasible set $\mathbb{S}$.

In the black-box setting, it is often the case that the query number is limited or high query efficiency is required by the adversary. The zeroth-order method tries to extract gradient information of the objective function and the first-order method is applied to minimize the loss due to its wide application in ML. However, the second-order information of the queries is not fully exploited. In this paper, we aim to take advantages of the model's second-order information and propose a novel method named ZO-NGD optimization.

Zeroth-order Nature Gradient Descent

The proposed method is based on NGD (Martens 2014) and ZO optimization (Duchi et al. 2015). In the applications of optimizing probabilistic models, NGD uses the natural gradient by multiplying the gradient with the FIM to update the parameters. NGD seems to be a potentially attractive alternative method as it requires fewer total iterations than gradient descent (Ollivier 2015; Martens and Grosse 2015; Grosse and Salakhudinov 2015).

Motivated from the perspective of information geometry, NGD defines the steepest descent/direction in the realizable distribution space instead of the parameter space. The distance in the distribution space is measured with a special "Riemannian metric" (Amari and Nagaoka 2007), which is different from the standard Euclidean distance metric in the parameter space. This Riemannian metric does not rely on the parameters like the Euclidean metric, but depends on the distributions themselves. Thus it is invariant to any smooth or invertible reparameterization of the model. More details are discussed in the Geometric Interpretation Section.

Next we will introduce the FIM and the implementation details to perform NGD. Basically, the proposed framework first queries the model to estimate the gradients and Fisher information. Then after the damping and inverting processes, natural gradient is obtained to update the perturbation. Algorithm 1 shows the pseudo code of the ZO-NGD.

Algorithm 1 Framework of ZO-NGD.

Require:
The legitimate image $\boldsymbol{x}$; the correct label t; the model to be queried; the learning rate λ; the sampling step size μ;
Ensure:
Adversarial perturbation $\boldsymbol{\delta}$;
initialize $\boldsymbol{\delta}_0$ with all zeros;
for $k = 0, ..., K$ **do**
  Query the model with $\boldsymbol{\delta}_k$ and obtain the probability $p(t|\boldsymbol{x}, \boldsymbol{\delta}_k) := p(t|\boldsymbol{x}+\boldsymbol{\delta}_k)$;
  for $j = 1, ..., R$ **do**
    Generate a random direction vector $\mathbf{u}_j$ drawn from a uniform distribution over the surface of a unit sphere;
    Query the model with $\boldsymbol{x}+\boldsymbol{\delta}_k+\mu\mathbf{u}_j$ and obtain $p(t|\boldsymbol{x}, \boldsymbol{\delta}_k+\mu\mathbf{u}_j)$;
  end for
  Estimate the gradients of the loss function $\hat{\nabla} f(\boldsymbol{\delta}_k)$ according to Eq. (16);
  Estimate the gradients of the log-likelihood function $\hat{\nabla} \log p(t\,|\mathbf{x}, \delta_{\mathbf{k}})$ according to Eq. (17);
  Compute the FIM $\mathbf{F}$ according to Eq. (18) and perform the nature gradient update as shown in Eq. (19).
end for

Fisher Information Matrix and Natural Gradient

We introduce and derive the FIM in this section. In general, finding an adversarial example can be formulated as a training problem. In the idealized setting, input vectors $\boldsymbol{x}$ are drawn independently from a distribution $Q_{\boldsymbol{x}}$ with density function $q(\boldsymbol{x})$, and the corresponding output t is drawn from a conditional target distribution $Q_{t|\mathbf{x}}$ with density function $q(t\,|\boldsymbol{x}\,)$. The target joint distribution is $Q_{t,\mathbf{x}}$ with the density

of $q(t, \boldsymbol{x}) = q(t|\boldsymbol{x})q(\boldsymbol{x})$. By finding an adversarial perturbation $\boldsymbol{\delta}$, we obtain the learned distribution $P_{t,\mathbf{x}}(\boldsymbol{\delta})$, whose density is $p(t, \boldsymbol{x}|\boldsymbol{\delta}) = p(t|\boldsymbol{x} + \boldsymbol{\delta})q(\boldsymbol{x}) := p(t|\boldsymbol{x}, \boldsymbol{\delta})q(\boldsymbol{x})$.

In statistics, the score function (Cox and Hinkley 1979) indicates how sensitive a likelihood function $p(t, \boldsymbol{x}|\boldsymbol{\delta})$ is to its parameters $\boldsymbol{\delta}$. Explicitly, the score function for $\boldsymbol{\delta}$ is the gradient of the log-likelihood with respect to $\boldsymbol{\delta}$ asbelow,

$$s(\boldsymbol{\delta}) = \nabla \log p(t, \boldsymbol{x}|\boldsymbol{\delta}). \tag{4}$$

Lemma 1 *The expected value of the score function with respect to $\boldsymbol{\delta}$ is zero.*

The proof is shown in the appendix. We can define an uncertainty measure around the expected value (i.e., the covariance of the score function) as follows,

$$\mathbb{E}\left[\left(s(\boldsymbol{\delta}) - \mathbb{E}[s(\boldsymbol{\delta})]\right)\left(s(\boldsymbol{\delta}) - \mathbb{E}[s(\boldsymbol{\delta})]\right)^{\mathrm{T}}\right]. \tag{5}$$

The covariance of the score function above is the definition of the Fisher information. It is in the form of a matrix and the FIM can be written as

$$\mathbf{F} = \mathbb{E}_{\boldsymbol{x}\in Q_x}\left[\mathbb{E}_{\tilde{t}\sim p(\cdot|\boldsymbol{x},\boldsymbol{\delta})}\left[\nabla \log p\left(\tilde{t}, \boldsymbol{x}\,|\boldsymbol{\delta}\right) \nabla \log p\left(\tilde{t}, \boldsymbol{x}\,|\boldsymbol{\delta}\right)^{T}\right]\right]. \tag{6}$$

Note that this expression involves the losses on all possible values of the classes $\tilde{t}$, not only the actual label for each data sample. As $\boldsymbol{\delta}$ only corresponds to a single input $\boldsymbol{x}$, the training set only contains one data sample. Besides, since $p(\tilde{t}, \boldsymbol{x}|\boldsymbol{\delta}) = p(\tilde{t}|\boldsymbol{x} + \boldsymbol{\delta})q(x) = p(\tilde{t}|\boldsymbol{x}, \boldsymbol{\delta})q(\boldsymbol{x})$ and $q(\boldsymbol{x})$ does not depend on $\boldsymbol{\delta}$, we have

$$\begin{aligned}\nabla \log p(\tilde{t}, \boldsymbol{x}|\boldsymbol{\delta}) &= \nabla \log p(\tilde{t}|\boldsymbol{x}, \boldsymbol{\delta}) + \nabla \log q(\boldsymbol{x}) \\ &= \nabla \log p(\tilde{t}|\boldsymbol{x}, \boldsymbol{\delta}).\end{aligned} \tag{7}$$

Then the FIM can be transformed to

$$\mathbf{F} = \mathbb{E}_{\tilde{t}\sim p(\cdot|\boldsymbol{x},\boldsymbol{\delta})}\left[\nabla \log p(\tilde{t}|\boldsymbol{x}, \boldsymbol{\delta}) \nabla \log p(\tilde{t}|\boldsymbol{x}, \boldsymbol{\delta})^{\mathrm{T}}\right]. \tag{8}$$

The exact expectation with T categories is expressed as,

$$\mathbf{F} = \sum_{\tilde{t}=1}^{T} p\left(\tilde{t}\,|\boldsymbol{x}, \boldsymbol{\delta}\right) \nabla \log p\left(\tilde{t}\,|\boldsymbol{x}, \boldsymbol{\delta}\right) \nabla \log p\left(\tilde{t}\,|\boldsymbol{x}, \boldsymbol{\delta}\right)^{T} \tag{9}$$

The usual definition of the natural gradient is

$$\tilde{\nabla} f(\boldsymbol{\delta}) = \mathbf{F}^{-1} \nabla f(\boldsymbol{\delta}), \tag{10}$$

and the NGD minimizes the loss function through

$$\boldsymbol{\delta}_{k+1} = \boldsymbol{\delta}_k - \lambda \tilde{\nabla} f(\boldsymbol{\delta}_k). \tag{11}$$

Outer Product and Monte Carlo Approximation

The FIM involves an expectation over all possible classes $\tilde{t} \sim p\left(\cdot\,|x, \delta\right)$ drawn from the probability distribution output. In the case with large number of classes, it is impractical to compute the exact FIM due to the intensive computation. To address the high computation overhead, in general there are two methods to approximate the FIM, the outer product approximation and the Monte Carlo approximation.

Outer Product Approximation The outer product approximation of the FIM (Pascanu and Bengio 2013a; Ollivier 2015) only uses the actual label t to avoid the expectation over all possible labels $\tilde{t} \sim p\left(\cdot\,|x, \delta\right)$, as below,

$$\mathbf{F}_{OP} = \nabla \log p\left(t\,|\boldsymbol{x}, \boldsymbol{\delta}\right) \nabla \log p\left(t\,|\boldsymbol{x}, \boldsymbol{\delta}\right)^{T}. \tag{12}$$

Thus a rank-one matrix can be obtained directly.

Monte Carlo Approximation Monte Carlo (MC) approximation (Ollivier 2015) replaces the expectation over $\tilde{t}$ with n_{MC} samples,

$$\mathbf{F}_{MC} = \frac{1}{n_{MC}} \sum_{i=1}^{n_{MC}} \nabla \log p\left(\tilde{t}_i\,|\boldsymbol{x}, \boldsymbol{\delta}\right) \nabla \log p\left(\tilde{t}_i\,|\boldsymbol{x}, \boldsymbol{\delta}\right)^{T}. \tag{13}$$

where each $\tilde{t}_i$ is drawn from the distribution $p\left(\cdot\,|x, \delta\right)$. The MC natural gradient works well in practice with $n_{MC} = 1$.

For higher query efficiency, we adopt the outer product approximation as it does not require additional queries .

Gaussian Smoothing and Gradient Estimation

To compute the FIM and perform NGD, we need to obtain the gradients of the loss function $\nabla f(\boldsymbol{\delta})$ and the gradients of the log-likelihood $\nabla p(t|\boldsymbol{x}, \boldsymbol{\delta})$, which are not directly available in the black-box setting.

To address this difficulty, we first introduce the Gaussian approximation of $f(\boldsymbol{x})$ (Nesterov and Spokoiny 2017),

$$f_\mu\left(\boldsymbol{x}\right) = \frac{1}{(2\pi)^{d/2}} \int_{R^d} f\left(\boldsymbol{x} + \mu\boldsymbol{u}\right) \exp\left(-\frac{1}{2}\|\boldsymbol{u}\|^2\right) d\boldsymbol{u}, \tag{14}$$

where $\|\cdot\|$ is the Frobenius norm, $\mu > 0$ is a smoothing parameter and $\boldsymbol{u}$ is a random vector distributed uniformly over the surface of a unit sphere, i.e., $\boldsymbol{u} \sim N\left(0, \mathbf{I}_d\right)$. Its gradient can be written as

$$\begin{aligned}\nabla f_\mu\left(\boldsymbol{x}\right) &= \frac{1}{M} \int_{R^d} \frac{f\left(\boldsymbol{x} + \mu\boldsymbol{u}\right) - f\left(\boldsymbol{x}\right)}{\mu} \boldsymbol{u} \exp\left(-\frac{1}{2}\|\boldsymbol{u}\|^2\right) d\boldsymbol{u} \\ &= E_{\boldsymbol{u}}\left(\frac{f\left(\boldsymbol{x} + \mu\boldsymbol{u}\right) - f\left(\boldsymbol{x}\right)}{\mu} \boldsymbol{u}\right),\end{aligned} \tag{15}$$

where $E_{\boldsymbol{u}}$ is the Gaussian smoothing function. Thus, based on Eq. (15), we apply the zeroth-order random gradient estimation to estimate the gradients by

$$\hat{\nabla} f(\boldsymbol{\delta}) = \frac{1}{R} \sum_{j=1}^{R} \frac{f(\boldsymbol{\delta} + \mu\mathbf{u}_j, t) - f(\boldsymbol{\delta}, t)}{\mu} \mathbf{u}_j, \tag{16}$$

and

$$\begin{aligned}\hat{\nabla} \log p(t\,|\mathbf{x}, \boldsymbol{\delta}) = \frac{1}{R\mu} \sum_{j=1}^{R} [&\log p(t\,|\mathbf{x}, \boldsymbol{\delta} + \mu\mathbf{u}_j) \\ &- \log p(t\,|\mathbf{x}, \boldsymbol{\delta})]\mathbf{u}_j,\end{aligned} \tag{17}$$

where R is the number of random direction vectors and $\{\mathbf{u}_j\}$ denote independent and identically distributed (i.i.d.) random direction vectors following Gaussian distribution.

We note that in each gradient estimation step, by querying the model $R + 1$ times, we can *simultaneously* obtain both the $\hat{\nabla} f(\boldsymbol{\delta})$ and $\hat{\nabla} \log p(t\,|\mathbf{x}, \boldsymbol{\delta})$ as demonstrated in Algorithm 1. Different from the zeroth-order gradient descent which only estimates the gradients of the loss function $\hat{\nabla} f(\boldsymbol{\delta})$ (such as Chen et al. and Ilyas et al.), ZO-NGD obtains $\hat{\nabla} \log p(t\,|\mathbf{x}, \boldsymbol{\delta})$ and computes the FIM from the same query outputs without incurring additional query complexity. This is one major difference between ZO-NGD and other zeroth-order methods. Thus, higher query-efficiency can be achieved by leveraging the FIM and second-order optimization.

Damping for Fisher Information Matrix

The inverse of the FIM is required for natural gradient. However, the eigenvalue distribution of the FIM is known to have an extremely long tail (Karakida, Akaho, and Amari 2018), where most of the eigenvalues are close to zero. This in turn causes the eigenvalues of the inverse FIM to be extremely large, leading to the unstable training. To mitigate this problem, damping technique is used to add a positive value to the diagonal of the FIM to stabilize the training as shown below,

$$\hat{\mathbf{F}} = \hat{\nabla} \log p\left(t \,|\boldsymbol{x}, \boldsymbol{\delta}\right) \hat{\nabla} \log p\left(t \,|\boldsymbol{x}, \boldsymbol{\delta}\right)^T + \gamma \mathbf{I}, \tag{18}$$

where γ is a constant. As the damping limits the maximum eigenvalue of the inverse FIM, we can restrict the norm of the gradients. This prevents ZO-NGD from moving too far in flat directions.

With the obtained FIM, the perturbation update is

$$\boldsymbol{\delta}_{k+1} = \prod_{\mathbb{S}} \left(\boldsymbol{\delta}_k - \lambda \hat{\mathbf{F}}^{-1} \hat{\nabla} f(\boldsymbol{\delta}_k) \right). \tag{19}$$

ZO-NGD tries to extract the Fisher information to perform second-order optimization for faster convergence rate and better query efficiency.

Scalability to High Dimensional Datasets

Note that the FIM has a dimension of d^2 where d is the dimension of the input image. On ImageNet dataset which typically contains images with about $270,000$ pixels ($\boldsymbol{x} \in \mathbb{R}^{299\times299\times3}$), the FIM would have billions of elements and thus it is quite difficult to compute or store the FIM, not to mention its inverse. In the application of training DNN models, the Kronecker Factored Approximate Curvature (K-FAC) method (Martens and Grosse 2015; Osawa et al. 2018) is adopted to deal with the difficulty of high dimensions of the DNN model. However, K-FAC methods may not be suitable in the application of finding adversarial examples as the assumption of uncorrelated channels is not valid and thus we can not apply the block diagonalization method for the FIM. Instead, we propose another method to compute $\hat{\mathbf{F}}^{-1}$ of high dimensions as follows. First we have

$$\hat{\nabla} \log p\left(t \,|\boldsymbol{x}, \boldsymbol{\delta}\right) = c \frac{\hat{\nabla} \log p\left(t \,|\boldsymbol{x}, \boldsymbol{\delta}\right)}{\|\hat{\nabla} \log p\left(t \,|\boldsymbol{x}, \boldsymbol{\delta}\right)\|} = c \frac{\hat{s}(\boldsymbol{\delta})}{\|\hat{s}(\boldsymbol{\delta})\|}, \tag{20}$$

where $c = \|\hat{s}(\boldsymbol{\delta})\|$. The inverse matrix $\hat{\mathbf{F}}^{-1}$ can be represented as,

$$\hat{\mathbf{F}}^{-1} = \frac{\left(\left(c^2+\gamma\right)^{-1} - \gamma^{-1}\right)}{c^2} \hat{s}\left(\boldsymbol{\delta}\right) \hat{s}\left(\boldsymbol{\delta}\right)^T + \gamma^{-1}\mathbf{I}. \tag{21}$$

This can be verified simply by checking their multiplication and we omit the proof here. Then the gradient update $\Delta\boldsymbol{\delta} = \lambda \hat{\mathbf{F}}^{-1} \nabla f(\boldsymbol{\delta})$ in Eq. (19) is

$$\begin{aligned} \Delta\boldsymbol{\delta} &= \lambda \left[\frac{\left(\left(c^2+\gamma\right)^{-1} - \gamma^{-1}\right)}{c^2} \hat{s}\left(\boldsymbol{\delta}\right) \hat{s}\left(\boldsymbol{\delta}\right)^T + \gamma^{-1}\mathbf{I} \right] \hat{\nabla} f\left(\boldsymbol{\delta}\right) \\ &= \lambda \frac{\left(\left(c^2+\gamma\right)^{-1} - \gamma^{-1}\right)}{c^2} \hat{s}\left(\boldsymbol{\delta}\right) \left[\hat{s}\left(\boldsymbol{\delta}\right)^T \hat{\nabla} f\left(\boldsymbol{\delta}\right) \right] \\ &\quad + \lambda\gamma^{-1} \hat{\nabla} f\left(\boldsymbol{\delta}\right). \end{aligned} \tag{22}$$

During the computation of $\Delta\boldsymbol{\delta} = \lambda \hat{\mathbf{F}}^{-1} \nabla f(\boldsymbol{\delta})$, we compute $\hat{s}\left(\boldsymbol{\delta}\right)^T \hat{\nabla} f\left(\boldsymbol{\delta}\right)$ first in Eq. (22) and obtain a scalar, then $\Delta\boldsymbol{\delta}$ is simply the sum of two vectors. Although $\hat{\mathbf{F}}$ and its inverse might have billions of elements, we avoid directly computing them and the dimension of the internal computation is at most the same level as the dimension d of the images, rather than its square d^2. Thus, the ZO-NGD method can be applied on datasets with high dimensional images.

Geometric Interpretation

We provide a geometric interpretation for the natural gradient here. The negative gradient $-\nabla f(\boldsymbol{\delta})$ can be interpreted as the steepest descent direction in the sense that it yields the most reduction in f per unit of change of $\boldsymbol{\delta}$, where the change is measured by the standard Euclidean norm $\|\cdot\|$ (Martens 2014), as shown below,

$$\frac{-\nabla f(\boldsymbol{\delta})}{\|\nabla f(\boldsymbol{\delta})\|} = \lim_{\epsilon \to 0} \frac{1}{\epsilon} \arg\min_{\|\boldsymbol{\alpha}\| \leq \epsilon} f(\boldsymbol{\delta} + \boldsymbol{\alpha}). \tag{23}$$

By following the $-\nabla f(\boldsymbol{\delta})$ direction, we can obtain the change of $\boldsymbol{\delta}$ within a certain ϵ-neighbourhood to minimize the loss function.

Lemma 2 *The negative natural gradient is the steepest descent direction in the distribution space.*

We provide the proof of Lemma 2 in the appendix[1]. In the parameter space, the negative gradient is the steepest descent direction to minimize the loss function. By contrast, in the distribution space where the distance is measured by KL divergence, the steepest descent direction is the negative natural gradient. Thus, the direction in distribution space defined by the natural gradient will be invariant to the choice of parameterization (Pascanu and Bengio 2013b), i.e., it will not be affected by how the model is parametrized, but only depends on the distribution induced by the parameters.

Experimental Results

In this section, we present the experimental results of the ZO-NGD method. We compare ZO-NGD with various attack methods on three image classification datasets, MNIST (Lecun et al. 1998), CIFAR-10 (Krizhevsky and Hinton 2009) and ImageNet (Deng et al. 2009).

We train two networks for MNIST and CIFAR-10 datasets, respectively. The model for MNIST achieves 99.6% accuracy with four convolutional layers, two max pooling layers, two fully connected layers and a softmax layer. For CIFAR-10, we adopt the same model architecture as MNIST, achieving 80% accuracy. For ImageNet, a pre-trained Inception v3 network (Szegedy et al. 2016) is utilized instead of training our own model, attaining 96% top-5 accuracy. All experiments are performed on machines with NVIDIA GTX 1080 TI GPUs.

Evaluation of White-box Attack

We first check the white-box setting, where we compare the proposed NGD with PGD from adversarial training. PGD is a typical first-order method while NGD utilizes the second-order FIM. The query here is defined as one forward pass and one subsequent backpropagation as we need to obtain

Table 1: Performance evaluation of black-box adversarial attacks on MNIST and CIFAR-10.

dataset	Attack method	success rate	average queries	reduction rate
MNIST	Transfer attack	82%	-	-
	ZOO attack	100%	8,300	0%
	NES-PGD	98.2%	1,243	85%
	ZO-NGD	98.7%	523	93.7%
CIFAR	Transfer attack	85%	-	-
	ZOO attack	99.8 %	6,500	0%
	NES-PGD	98.9%	417	93.6%
	ZO-NGD	99.2 %	131	98%

the gradients through backpropagation. We report the average number of queries over 500 images for successful adversaries on each dataset. On MNIST, NGD requires 2.12 queries while PGD needs 4.88 queries with $\epsilon = 0.2$. On CIFAR-10, NGD requires 2.06 queries while PGD needs 4.21 queries with $\epsilon = 0.1$. On ImageNet, NGD requires 2.20 queries while PGD needs 5.62 queries with $\epsilon = 0.05$. We can see that NGD achieves higher query efficiency by incorporating FIM.

Evaluation on MNIST and CIFAR-10

In the evaluation on MNIST and CIFAR-10, we select 2000 correctly classified images from MNIST and CIFAR-10 test datasets, respectively, and perform black-box attacks for these images. We compare the ZO-NGD method with the transfer attack (Papernot, McDaniel, and Goodfellow 2016), ZOO black-box attack (Chen et al. 2017b), and the natural-evolution-strategy-based projected gradient descent method (NES-PGD) (Ilyas et al. 2018). For the transfer attack (Papernot, McDaniel, and Goodfellow 2016), we apply C&W attack (Carlini and Wagner 2017) to the surrogate model. The implementations of ZOO and NES-PGD are based on the GitHub code released by the authors[1]. For the attack methods, the pixel values of all images are normalized to the range of $[0, 1]$. In the proposed ZO-NGD method, the sampling number R in the random gradient estimation as defined in Eq. (16) and (17) is set to 40. ϵ is set to 0.4 for MNIST and 0.2 for CIFAR-10 or ImageNet. In Eq. (16) and (17), we set $\mu = 1$ for three datasets. γ is set to 0.01.

The experimental results are summarized in Table 1. We show the success rate and the average queries over successful adversarial examples for the black-box attack methods on MNIST and CIFAR-10 datasets. As shown in Table 1, the transfer attack does not achieve high success rate due to the difference between the surrogate model and the original target model. The ZOO attack method can achieve high success rate at the cost of excessive query complexity since it performs gradient estimation for each pixel of the input image. We can observe that the ZO-NGD method requires significantly less queries than the NES-PGD method. NES-PGD uses natural evolutionary strategies for gradient estimation and then perform first-order gradient descent to obtain the adversarial perturbations. Compared with NES-PGD, the proposed ZO-NGD not only estimates the first-order gradients of the loss function, but also tries to obtain the second-order Fisher information from the queries without incurring additional query complexities, leading to higher query-efficiency. From Table 1, we can observe that the ZO-NGD method attains the smallest number of queries to successfully obtain the adversarial examples in the black-box setting. Benchmarking on the ZOO method, the query reduction ratio of ZO-NGD can be as high as 93.7% on MNIST and 98% on CIFAR-10.

[1]The code and appendix are available at https://github.com/LinLabNEU/ZO_NGD_blackbox.

Table 2: Performance evaluation of black-box adversarial attacks on ImageNet.

Attack method	success rate	average queries	reduction ratio
ZOO attack	98.9%	16,800	0%
NES-PGD	94.6%	1,325	92.1%
Bandits[TD]	96.1%	791	95.3%
ZO-NGD	97%	582	96.5%

Evaluation on ImageNet

We perform black-box adversarial attacks on ImageNet where 1000 correctly classified images are randomly selected. On ImageNet, we compare the proposed ZO-NGD with the ZOO attack, NES-PGD method and the bandit attack with time and data-dependent priors (named as Bandits[TD]) (Ilyas, Engstrom, and Madry 2018). The transfer attack is not performed since it is not easy to train a surrogate model on ImageNet. The Bandits[TD] method makes use of the prior information for the gradients estimation, including the time-dependent priors which explores the heavily correlated successive gradients, and the data-dependent priors which exploits the spatially local similarity exhibited in images. After gradient estimation with the priors or bandits information, first-order gradient descent method is applied.

We present the performance evaluation on ImageNet in Table 2. The success rate and the average queries over successful attacks for various black-box attack methods are reported. Table 2 shows the ZOO attack method can achieve high success rate with high query complexity due to its element-wise gradient estimation. We can have a similar observation that the ZO-NGD method only requires a much smaller number of queries than the NES-PGD method due to the faster convergence rate of second-order optimization by exploring the Fisher information. We also find that the ZO-NGD method also outperforms the Bandits[TD] method in terms of query efficiency. The Bandits[TD] method enhances the query efficiency of gradient estimations through the incorporation of priors information for the gradients, but its attack methodology is still based on the first-order gradient descent method. As observed from Table 2, the ZO-NGD method achieves the highest query-efficiency for successful adversarial attacks in the black-box setting. It can obtain 96.5% query reduction ratio on ImageNet when compared

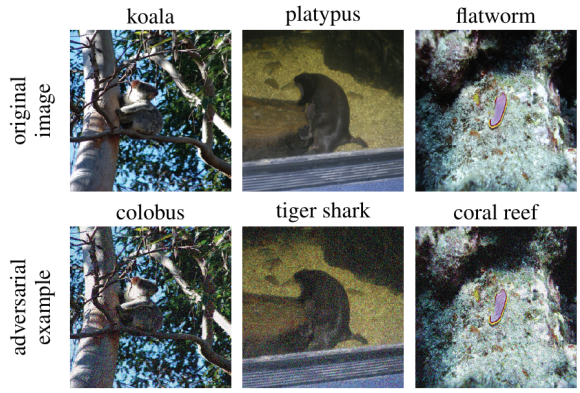

Figure 1: The legitimate images and their adversarial examples generated by ZO-NGD.

with the ZOO method. In Figure 1, we show some legitimate images on ImageNet and their corresponding adversarial examples obtained by ZO-NGD. We can observe that the adversarial perturbations are imperceptible. More examples on MNIST and CIFAR-10 are shown in the appendix.

Ablation study

In this ablation study, we perform sensitivity analysis on the proposed ZO-NGD method based variations in model architectures and different parameter settings. Below we summarize the conclusion and findings from this ablation study and report their details in the appendix. (1) Tested on VGG16 and ResNet and varying the parameters μ and ϵ in ZO-NGD, the results demonstrate the consistent superior performance of ZO-NGD by leveraging the second-order optimization. (2) We inspect the approximation techniques used in ZO-NGD including damping and outer product method. The results show that there is a wide range of proper γ values such that damping can work effectively to reduce the loss, and the outer product is a reasonable approximation based on the empirical evidence. We also note that the ASR of ZOO is higher than ZO-NGD. We provide a discussion about the ASR v.s. query number in the appendix.

Query Number Distribution Figure 2 shows the cumulative distribution (CDF) of the query number for 1000 images on three datasets, validating ZO-NGD's query efficiency.

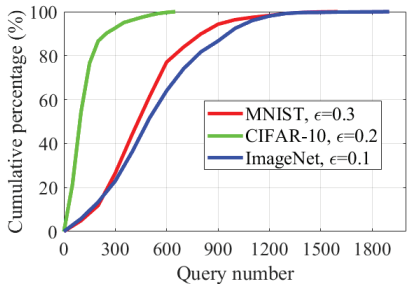

Figure 2: CDF of query number on three datasets using ZO-NGD.

Transferability

The transferability of adversarial examples is an interesting and valuable metric to measure the performance. To show the transferability, we use 500 targeted adversarial examples generated by ZO-NGD on ImageNet with $\epsilon = 0.1$ on Inception to attack ResNet and VGG16 model. It achieves 94.4% and 95.6% ASR, respectively, demonstrating high transferability of our method. Our transferred ASR is also higher than NES-PGD (92.1% and 92.9% ASR).

Conclusion

In this paper, we propose a novel ZO-NGD to achieve high query-efficiency in black-box adversarial attacks. It incorporates the ZO random gradient estimation and the second-order FIM for NGD. The performance evaluation on three image classification datasets demonstrate the effectiveness of the proposed method in terms of fast convergence and improved query efficiency over state-of-the-art methods.

Acknowledgments

This work is partly supported by the National Science Foundation CNS-1932351, and is also based upon work supported by the Department of Energy National Energy Technology Laboratory under Award Number DE-OE0000911.

References

Alzantot, M.; Balaji, B.; and Srivastava, M. B. 2018. Did you hear that? adversarial examples against automatic speech recognition. *CoRR* abs/1801.00554.

Amari, S.-i., and Nagaoka, H. 2007. *Methods of information geometry*, volume 191. American Mathematical Soc.

Amari, S.-I. 1998. Natural gradient works efficiently in learning. *Neural Comput.* 10(2):251–276.

Biggio, B.; Fumera, G.; and Roli, F. 2014. Security evaluation of pattern classifiers under attack. *IEEE TKDE* 26(4):984–996.

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

Bulò, S. R.; Biggio, B.; and et. al. 2017. Randomized prediction games for adversarial machine learning. *IEEE Transactions on Neural Networks and Learning Systems* 28(11):2466–2478.

Carlini, N., and Wagner, D. 2017. Towards evaluating the robustness of neural networks. In *Security and Privacy (SP), 2017 IEEE Symposium on*, 39–57. IEEE.

Chen, P.-Y.; Sharma, Y.; Zhang, H.; Yi, J.; and Hsieh, C.-J. 2017a. Ead: elastic-net attacks to deep neural networks via adversarial examples. *arXiv preprint arXiv:1709.04114*.

Chen, P.-Y.; Zhang, H.; Sharma, Y.; Yi, J.; and Hsieh, C.-J. 2017b. 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*, 15–26. ACM.

Cox, D. R., and Hinkley, D. V. 1979. *Theoretical statistics*. Chapman and Hall/CRC.

Demontis, A.; Melis, M.; and et. al. 2018. Yes, machine learning can be more secure! a case study on android malware detection. *IEEE TDSC* 1–1.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, 248–255. IEEE.

Duchi, J. C.; Jordan, M. I.; Wainwright, M. J.; and Wibisono, A. 2015. Optimal rates for zero-order convex optimization: The power of two function evaluations. *IEEE Transactions on Information Theory* 61(5):2788–2806.

Goodfellow, I.; Shlens, J.; and Szegedy, C. 2015. Explaining and harnessing adversarial examples. *2015 ICLR* arXiv preprint arXiv:1412.6572.

Grosse, R. B., and Martens, J. 2016. A kronecker-factored approximate fisher matrix for convolution layers. In *ICML*, volume 1, 2.

Grosse, R., and Salakhudinov, R. 2015. Scaling up natural gradient by sparsely factorizing the inverse fisher matrix. In Bach, F., and Blei, D., eds., *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, 2304–2313. Lille, France: PMLR.

Ilyas, A.; Engstrom, L.; Athalye, A.; and Lin, J. 2018. Black-box adversarial attacks with limited queries and information. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018*.

Ilyas, A.; Engstrom, L.; and Madry, A. 2018. Prior convictions: Black-box adversarial attacks with bandits and priors. *ICLR 2019*.

Karakida, R.; Akaho, S.; and Amari, S.-i. 2018. Universal statistics of fisher information in deep neural networks: Mean field approach. *arXiv preprint arXiv:1806.01316*.

Krizhevsky, A., and Hinton, G. 2009. Learning multiple layers of features from tiny images. *Master's thesis, Department of Computer Science, University of Toronto*.

LeCun, Y.; Bengio, Y.; and Hinton, G. 2015. Deep learning. 521:436–44.

Lecun, Y.; Bottou, L.; Bengio, Y.; and Haffner, P. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE* 86(11):2278–2324.

Lin, Y.-C.; Hong, Z.-W.; and et. al. 2017. Tactics of adversarial attack on deep reinforcement learning agents. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, 3756–3762. AAAI Press.

Liu, S.; Chen, P.-Y.; Chen, X.; and Hong, M. 2019. SignSGD via zeroth-order oracle. *International Conference on Learning Representations*.

Madry, A.; Makelov, A.; Schmidt, L.; Tsipras, D.; and Vladu, A. 2017. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*.

Martens, J., and Grosse, R. 2015. Optimizing neural networks with kronecker-factored approximate curvature. In *International conference on machine learning*, 2408–2417.

Martens, J. 2014. New perspectives on the natural gradient method. *CoRR* abs/1412.1193.

Martens, J. 2016. *Second-order optimization for neural networks*. University of Toronto (Canada).

Nesterov, Y., and Spokoiny, V. 2017. Random gradient-free minimization of convex functions. *Foundations of Computational Mathematics* 17(2):527–566.

Ollivier, Y. 2015. Riemannian metrics for neural networks i: feedforward networks. *Information and Inference: A Journal of the IMA* 4(2):108–153.

Osawa, K.; Tsuji, Y.; Ueno, Y.; Naruse, A.; Yokota, R.; and Matsuoka, S. 2018. Second-order optimization method for large mini-batch: Training resnet-50 on imagenet in 35 epochs. *CVPR 2019*.

Papernot, N.; McDaniel, P. D.; and Goodfellow, I. J. 2016. Transferability in machine learning: from phenomena to black-box attacks using adversarial samples. *CoRR* abs/1605.07277.

Pascanu, R., and Bengio, Y. 2013a. Natural gradient revisited. *CoRR* abs/1301.3584.

Pascanu, R., and Bengio, Y. 2013b. Revisiting natural gradient for deep networks. *arXiv preprint arXiv:1301.3584*.

Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; and Wojna, Z. 2016. Rethinking the inception architecture for computer vision. *CVPR 2016* 2818–2826.

Tu, C.-C.; Ting, P.; Chen, P.-Y.; Liu, S.; Zhang, H.; Yi, J.; Hsieh, C.-J.; and Cheng, S.-M. 2018. Autozoom: Autoencoder-based zeroth order optimization method for attacking black-box neural networks. *arXiv preprint arXiv:1805.11770*.

Wang, S.; Wang, X.; Zhao, P.; Wen, W.; Kaeli, D.; Chin, P.; and Lin, X. 2018a. Defensive dropout for hardening deep neural networks under adversarial attacks. In *ICCAD '18*.

Wang, Y.; Du, S.; Balakrishnan, S.; and Singh, A. 2018b. Stochastic zeroth-order optimization in high dimensions. In *AISTATS 2018*, volume 84 of *Proceedings of Machine Learning Research*. PMLR.

Wang, X.; Wang, S.; Chen, P.-Y.; Wang, Y.; Kulis, B.; Lin, X.; and Chin, S. 2019. Protecting neural networks with hierarchical random switching: Towards better robustness-accuracy trade-off for stochastic defenses. In *IJCAI 2019*.

Xie, C.; Wang, J.; and et. al. 2018. Adversarial examples for semantic segmentation and object detection. In *ICCV 2017*, 1378–1387.

Xu, K.; Liu, S.; Zhao, P.; Chen, P.-Y.; Zhang, H.; Erdogmus, D.; Wang, Y.; and Lin, X. 2018. Structured Adversarial Attack: Towards General Implementation and Better Interpretability. *ArXiv e-prints*.

Xu, K.; Chen, H.; Liu, S.; Chen, P.-Y.; Weng, T.-W.; Hong, M.; and Lin, X. 2019. Topology attack and defense for graph neural networks: An optimization perspective. *arXiv preprint arXiv:1906.04214*.

Ye, H.; Huang, Z.; Fang, C.; Li, C. J.; and Zhang, T. 2018. Hessian-aware zeroth-order optimization for black-box adversarial attack. *CoRR* abs/1812.11377.

Zhang, H.; Weng, T.-W.; and et. al. 2018. Efficient Neural Network Robustness Certification with General Activation Functions. In *NIPS 2018*.

Zhao, P.; Liu, S.; Wang, Y.; and Lin, X. 2018. An admm-based universal framework for adversarial attacks on deep neural networks. In *ACM Multimedia 2018*.

Zhao, P.; Liu, S.; Chen, P.-Y.; Hoang, N.; Xu, K.; Kailkhura, B.; and Lin, X. 2019a. On the design of black-box adversarial examples by leveraging gradient-free optimization and operator splitting method. In *ICCV 2019*.

Zhao, P.; Wang, S.; Gongye, C.; Wang, Y.; Fei, Y.; and Lin, X. 2019b. Fault sneaking attack: A stealthy framework for misleading deep neural networks. In *DAC 2019*.

Zhao, P.; Xu, K.; Liu, S.; Wang, Y.; and Lin, X. 2019c. Admm attack: An enhanced adversarial attack for deep neural networks with undetectable distortions. In *ASPDAC 2019*.