ZOO: ZEROTH ORDER OPTIMIZATION BASED BLACK-BOX ATTACKS TO DEEP NEURAL NETWORKS WITHOUT TRAINING SUBSTITUTE MODELS

Brinda Ashar (50361965)

University at Buffalo The State University of New York

INTRODUCTION

Introduction

- Deep neural networks (DNNs) are used to perform image classification, text mining, speech processing etc.
- There are rising concerns on the robustness of DNNs for tasks like traffic sign identification for autonomous driving
- These tasks require a high level of security, as it is easy for someone to include a set of images that might lead to misclassification

Adversarial attacks and transferability

- Methods of attacks –
  - Fast gradient sign method (FGSM):
    - Originated from L-INF constraint on maximal distortion
    - Considers the sign from the backpropogation of the target DNN to create admissible adversarial images
  - Jacobian-based Saliency Map Attack (JSMA):
    - Uses the Jacobian-based saliency map to analyse the input-output relation of the targeted DNN
    - In each iteration, the most significant pixel is modified a bit to fool the model
    - The change misleads the classification but is expected to not be distinguishable by the machine or the human eye

Adversarial attacks and transferability

- Methods of attacks –
  - DeepFool:
    - Untargeted attack algorithm, uses the knowledge that corresponding separating hyperplanes are infact decision boundaries for each class
    - Aims to find the least distortion (i.e Euclidean distance) by projecting a potential adversarial image onto the closest hyperplane
  - Carlini & Wagner (C&W) Attack
    - Take advantage of the internal configurations of the targeted DNN and calculate the difference between the original and adversarial image outputs using L2 norm
    - Targeted adversarial attack, uses representation in the logit layer as a measure of effectiveness
    - Successfully bypasses 10 different detections methods designed for detecting adversarial examples, considered to be the best attack

Adversarial attacks and transferability

- Methods of attacks –
  - Transferability:
    - Adversarial examples that are misclassified by one model are likely to be misclassified by another model as well
    - Transferability raises concerns for security as the above methods prove that it is easy to craft malicious images even if the internal configurations of DNN are unknown

Black box attack

- What is a black box attack?
  - Legit pictures and class labels given; no information on internal configurations of the target DNN
  - Attacker cannot query the classifier for its internal structure; however, the targeted DNN can be queried by giving inputs and observing outputs
  - This can be used by the attacker to create a substitute model which can be iteratively improved to behave like the targeted DNN
  - The adversarial images can be used to attack the targeted DNN using the **transferability property**
- What is a white box attack? – Also known as “open-box attack”, has full access to the knowledge of internal structure of a Neural network

Substitute Models

- What is a substitute model?
  - Black box attack allows a user to have access to only the input and output of the targeted DNN
  - A substitute model is built on top of the targeted model, to act like the targeted DNN
  - On this substitute model, the user has full access to the internal configurations, which he can change iteratively to behave like the targeted DNN
- Why is a substitute model needed?
  - Since the user has full access to the substitute model, he can observe the output for a certain adversarial input on the targeted DNN, and tune the substitute model to make it work the same way
  - This helps the attacker to create perfect adversarial examples that can be used to attack the targeted DNNs using the transferability property

Defense against adversarial attacks

- Defense methods –
  - Detection based method:
    - Assumes that the distribution of the legit examples and adversarial examples are fundamentally distinct
    - Aims to differentiate using statistical tests and out-of-sample analysis
    - Feature squeezing is used to detect adversarial examples by projecting the example to a confined space which decreases the effect of the attack
    - In the C&W attack, however, it is stated that the distribution of the examples is nearly indistinguishable, hence proving to be the strongest attack
  - Gradient and representation masking:
    - As gradient knowledge from backpropogation is largely used to create adversarial examples, gradient masking is used
    - Example is to retrain a model using defense distillation based on soft labels, and a new concept ‘temperature’ in the softmax step
    - Another example is representation masking
    - The last few layers (logit layer representation) is replaced with more robust representations like Gaussian processes of RBF kernels

Defense against adversarial attacks

- Defense methods –
  - Adversarial training:
    - The DNNs will be less sensitive to adversarial examples if the examples are used to stabilize the training (data augmentation method)
    - The DNN is made more robust by retraining and adding more model parameters, which increases the networking capacity of the model

Black-box attack using zeroth order optimization

- What are zeroth order methods?
  - Derivative-free optimization method, using zeroth order oracle (objective function value f(x) for any x)
  - Gradient can be calculated by evaluating the objective function value f(x) at two very close points f(x+hv) and f(x-hv) (h -> small number)
  - The proposed black-box attack in this paper leverages the characteristics of ZOO to avoid the creation and training of a substitute model
  - ZOO ***has*** to be used with the correct optimization method, like a stochastic gradient method, especially for large image inputs
  - ZOO, like SGD, iteratively optimizes and updates a coordinate or a batch of coordinates instead of computing full gradients for efficient updates
  - Techniques to speed up computational power and reduce the number of queries: attack-space dimension reduction, hierarchical attacks and importance sampling

University at Buffalo The State University of New York

ZOO ATTACK

Black box attack without training a substitute model

- Notations used in formulation of the attack:
  - *F(x)* -> Targeted DNN
  - $\mathbf{x} \in \mathbb{R}^p$ -> Image input (p-dimensional vector)
  - $F(\mathbf{x}) \in [0,1]^K$ -> Output vector of confidence scores for each class (*K* = no. of classes)
  - The *k*th entry in the above vector indicates the probability of classifying *x* in class *k*

Formulation of C&W attack

- Is the strongest white-box attack on DNNs
- Finds adversarial example **x** by solving the optimization problem:
- $$\begin{aligned} &\text{minimize}_{\mathbf{x}} \ \|\mathbf{x}-\mathbf{x}_0\|_2^2 + c \cdot f(\mathbf{x},t) \\ &\text{subject to } \mathbf{x} \in [0,1]^p, \end{aligned} \qquad \text{-----(1)}$$

where $\|\mathbf{v}\|_2 = \sqrt{\sum_{i=1}^{p} v_i^2}$ denotes L2 norm of a vector ***v*** and $c > 0$ is a regularization parameter

- $$f(\mathbf{x},t) = \max\{\max_{i \neq t}[Z(\mathbf{x})]_i - [Z(\mathbf{x})]_t, -\kappa\}, \qquad \text{------(2)}$$

was proposed by Carlini and Wagner, where $[Z(x)]_k$ is the predicted probability that *x* belongs to class *k*, and kappa (*K)* is the tuning parameter for attack transferability

- The output of a DNN is determined by the softmax function:
- $$[F(\mathbf{x})]_k = \frac{\exp([Z(\mathbf{x})]_k)}{\sum_{i=1}^{K} \exp([Z(\mathbf{x})]_i)}, \ \forall\, k \in \{1,\ldots,K\}. \qquad \text{------(3)}$$

- Based on the above function, $\max_{i \neq t}[Z(\mathbf{x})]_i - [Z(\mathbf{x})]_t \leq 0$ indicates that the attack was successful, and if the value is greater than 0, it indicates that the attack using *x* was unsuccessful

Proposed black-box attack via ZO Stochastic Coordinate Descent

- Methods proposed to amend the black box attack to not rely on assumptions that it's a white box attack:
  - Modifying loss function f(x,t) such that it only depends on the output of DNN (with desired class label t):
    - A new loss function is introduced which is defined as:

$$f(\mathbf{x}, t) = \max\{\max_{i \neq t} \log[F(\mathbf{x})]_i - \log[F(\mathbf{x})]_t, -\kappa\}, \qquad (4)$$

    - The log operator reduces the intensity of skew on the confidence scores of classes in the output
    - A similar loss function is devised for untargeted attacks where the attack is assumed to be successful when *x* is classified as a part of any class other than t_0

$$f(\mathbf{x}) = \max\{\log[F(\mathbf{x})]_{t_0} - \max_{i \neq t_0} \log[F(\mathbf{x})]_i, -\kappa\}, \qquad (5)$$

  - ZOO on the loss function by calculating approximate gradient:

$$\hat{g}_i := \frac{\partial f(\mathbf{x})}{\partial \mathbf{x}_i} \approx \frac{f(\mathbf{x} + h\mathbf{e}_i) - f(\mathbf{x} - h\mathbf{e}_i)}{2h}, \qquad (6)$$

    - A symmetric difference quotient is used to calculate the gradient

Proposed black-box attack via ZO Stochastic Coordinate Descent (..contd)

- h has a value as small as 0.0001, e_i is a standard basis vector with only the i^{th} component as 1
- We can estimate the coordinate-wise Hessian estimate using one more objective function evaluation

$$\hat{h}_i := \frac{\partial^2 f(\mathbf{x})}{\partial \mathbf{x}_{ii}^2} \approx \frac{f(\mathbf{x} + h\mathbf{e}_i) - 2f(\mathbf{x}) + f(\mathbf{x} - h\mathbf{e}_i)}{h^2}. \qquad (7)$$

- Since in a black-box attack setting, the network structure is unknown and backpropogation is not allowed, we cannot use gradient descent techniques for training
- One of the naïve solutions is to apply objective function evaluation similar to that in eqn (7), but that solution is too expensive as it requires thousands to tens of thousands of function evaluations in each step, and around hundreds of iterations to converge
- The proposed solution is to use coordinate-wise updates for each step, and each step only has 2 function evaluations

Proposed black-box attack via ZO Stochastic Coordinate Descent (..contd)

- Stochastic Coordinate descent:

Algorithm 1 Stochastic Coordinate Descent

while not converged **do**
  Randomly pick a coordinate $i \in \{1, \ldots, p\}$
  Compute an update δ^* by approximately minimizing

$$\arg\min_{\delta} f(\mathbf{x} + \delta \mathbf{e}_i)$$

  Update $\mathbf{x}_i \leftarrow \mathbf{x}_i + \delta^*$
end while

- At each iteration, a coordinate is select at random and updated by approximately minimizing the objective function along that coordinate
- The best δ is decided based on the estimated gradient and Hessian for the image using any first or second order method
- For this, ADAM and Newton are the two methods used
- Experimental results in the upcoming sections prove that ADAM is faster than Newton method

Proposed black-box attack via ZO Stochastic Coordinate Descent (..contd)

- ADAM method:

Algorithm 2 ZOO-ADAM: Zeroth Order Stochastic Coordinate Descent with Coordinate-wise ADAM

Require: Step size η, ADAM states $M \in \mathbb{R}^p, v \in \mathbb{R}^p, T \in \mathbb{Z}^p$, ADAM hyper-parameters $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$

$M \leftarrow \mathbf{0}, v \leftarrow \mathbf{0}, T \leftarrow \mathbf{0}$
while not converged **do**
  Randomly pick a coordinate $i \in \{1, \cdots, p\}$
  Estimate $\hat{g}_i$ using (6)
  $T_i \leftarrow T_i + 1$
  $M_i \leftarrow \beta_1 M_i + (1-\beta_1)\hat{g}_i, \quad v_i \leftarrow \beta_2 v_i + (1-\beta_2)\hat{g}_i^2$
  $\hat{M}_i = M_i/(1-\beta_1^{T_i}), \quad \hat{v}_i = v_i/(1-\beta_2^{T_i})$
  $\delta^* = -\eta \frac{\hat{M}_i}{\sqrt{\hat{v}_i}+\epsilon}$
  Update $\mathbf{x}_i \leftarrow \mathbf{x}_i + \delta^*$
end while

- Newton method:

Algorithm 3 ZOO-Newton: Zeroth Order Stochastic Coordinate Descent with Coordinate-wise Newton's Method

Require: Step size η

while not converged **do**
  Randomly pick a coordinate $i \in \{1, \cdots, p\}$
  Estimate $\hat{g}_i$ and $\hat{h}_i$ using (6) and (7)
  if $\hat{h}_i \leq 0$ **then**
    $\delta^* \leftarrow -\eta \hat{g}_i$
  else
    $\delta^* \leftarrow -\eta \frac{\hat{g}_i}{\hat{h}_i}$
  end if
  Update $\mathbf{x}_i \leftarrow \mathbf{x}_i + \delta^*$
end while

Attack space dimension reduction

- The term $\Delta \mathbf{x} = \mathbf{x} - \mathbf{x}_0$ is used to define the noise introduced in the original image. The optimization problem starts with $\Delta x = 0$
- For large input sizes($\Delta \mathbf{x} \in \mathbb{R}^p$), since it takes a large amount of time to go over the entire attack space, the author introduces dimension reduction transformation *D(y)*, where $\mathbf{y} \in \mathbb{R}^m$ and $m < p$
- Therefore eqn (1) becomes:

$$\begin{aligned} &\text{minimize}_{\mathbf{y}} \ \|D(\mathbf{y})\|_2^2 + c \cdot f(\mathbf{x}_0 + D(\mathbf{y}), t) \qquad (8) \\ &\text{subject to } \mathbf{x}_0 + D(\mathbf{y}) \in [0,1]^p. \end{aligned}$$

Hierarchical Attacks

- In dimensionality reduction, if the *m* is too small, a valid attack might not be possible due to small search space, and similarly, a larger *m* would be quite slow.
- Hence, a proposed solution is to use a series of transformations *D1, D2, D3….* with constantly increasing dimensions *m1, m2, m3..* during the optimization process

Optimizing important pixels

- In Stochastic coordinate descent, updating gradient and computing Hessian for each pixel is costly
- Hence only the important pixels are chosen for updation; this method is called importance sampling
- Eg: Pixels at the corners and edges are not as important as the ones near the main object of focus, possibly at the center of the image. Hence those pixels are updated with noise to create an adversarial image
- The image is divided into 8x8 regions, and for each region, sampling property is defined based on how large the change in pixel value in that region is
- Incorporating importance sampling as attack space is increased in hierarchical attacks is necessary

Experiment and performance analysis

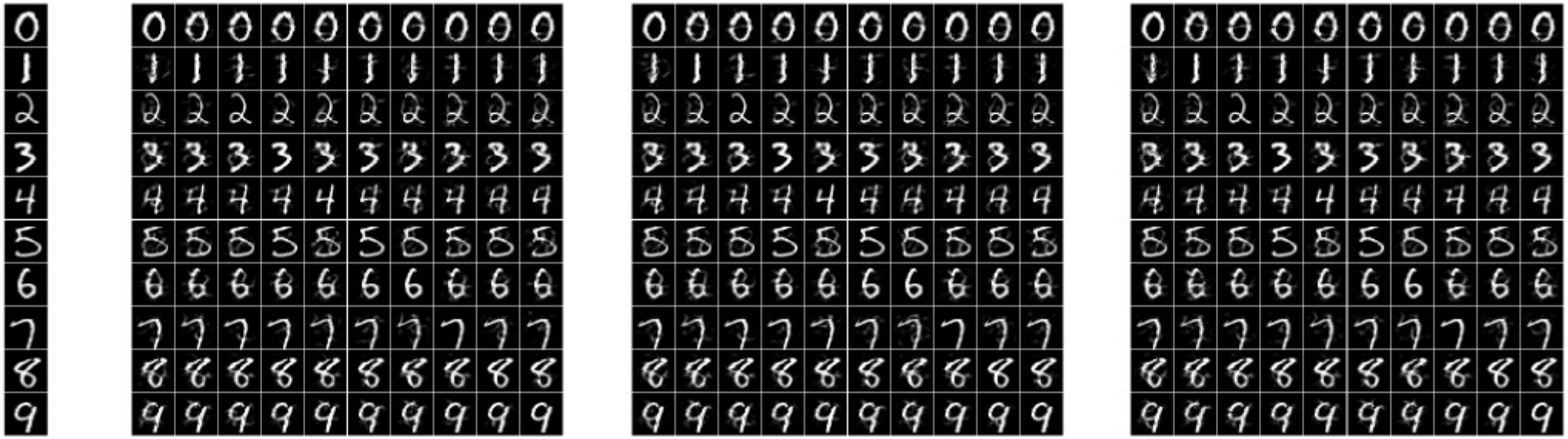
Using MNIST and CIFAR10 training/testing sets

	MNIST					
	Untargeted			Targeted		
	Success Rate	Avg. L_2	Avg. Time (per attack)	Success Rate	Avg. L_2	Avg. Time (per attack)
White-box (C&W)	100 %	1.48066	0.48 min	100 %	2.00661	0.53 min
Black-box (Substitute Model + FGSM)	40.6 %	-	0.002 sec (+ 6.16 min)	7.48 %	-	0.002 sec (+ 6.16 min)
Black-box (Substitute Model + C&W)	33.3 %	3.6111	0.76 min (+ 6.16 min)	26.74 %	5.272	0.80 min (+ 6.16 min)
Proposed black-box (ZOO-ADAM)	100 %	1.49550	1.38 min	98.9 %	1.987068	1.62 min
Proposed black-box (ZOO-Newton)	100 %	1.51502	2.75 min	98.9 %	2.057264	2.06 min
	CIFAR10					
	Untargeted			Targeted		
	Success Rate	Avg. L_2	Avg. Time (per attack)	Success Rate	Avg. L_2	Avg. Time (per attack)
White-box (C&W)	100 %	0.17980	0.20 min	100 %	0.37974	0.16 min
Black-box (Substitute Model + FGSM)	76.1 %	-	0.005 sec (+ 7.81 min)	11.48 %	-	0.005 sec (+ 7.81 min)
Black-box (Substitute Model + C&W)	25.3 %	2.9708	0.47 min (+ 7.81 min)	5.3 %	5.7439	0.49 min (+ 7.81 min)
Proposed Black-box (ZOO-ADAM)	100 %	0.19973	3.43 min	96.8 %	0.39879	3.95 min
Proposed Black-box (ZOO-Newton)	100 %	0.23554	4.41 min	97.0 %	0.54226	4.40 min

Results of experiment and performance analysis

Using MNIST and CIFAR10 training/testing sets

- ZOO achieves nearly 100% success rate
- The L2 distortion rates are close to the ones in C&W white box attack
- Success rate for ZOO is higher than substitute-based attack methods, especially for targeted attacks



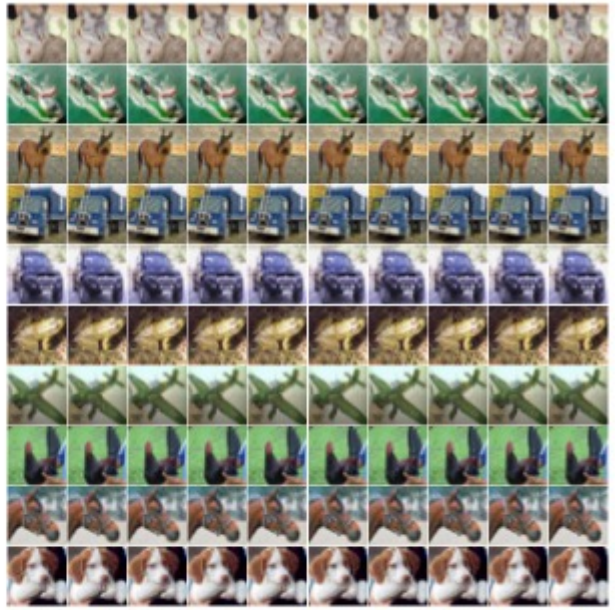
(a) (b) White-box C&W attack (c) ZOO-ADAM black-box attack (d) ZOO-Newton black-box attack

Figure 4: Visual comparison of successful adversarial examples in MNIST. Each row displays crafted adversarial examples from the sampled images in (a). Each column in (b) to (d) indexes the targeted class for attack (digits 0 to 9).

(a)

(b) White-box C&W attack

(c) ZOO-ADAM black-box attack



(d) ZOO-Newton black-box attack

Experiment and performance analysis

Using Inception-v3 black-box network (very large network) (Untargeted attack)

- In this experiment, hierarchical attack is not used
- Every attack has 1500 iterations and runs for about 20 mins
- Attack space used is (32x32x3), instead of the original (299x299x3)

Table 2: Untargeted ImageNet attacks comparison. Substitute model based attack cannot easily scale to ImageNet.

	Success Rate	Avg. L_2
White-box (C&W)	100 %	0.37310
Proposed black-box (ZOO-ADAM)	88.9 %	1.19916
Black-box (Substitute Model)	N.A.	N.A.

Results of experiment and performance analysis

Using Inception-v3 black-box network (very large network)

- Proposed black box attack gets a 90% success rate
- The L2 distortion in the proposed attack is 3 times larger than in the white box attack, but the adversarial images are still indistinguishable to the human eye
- The distortion can be improved if the number of iterations are increased

Experiment and performance analysis

Using Inception-v3 black-box network (very large network) (Targeted attack)

- In this experiment, hierarchical attack is used
- The ADAM algorithm is used with 20,000 iterations and runs for about 260 mins
- Attack space is started with (32x32x3)
- Importance sampling is used

Table 3: Comparison of different attack techniques. "First Valid" indicates the iteration number where the first successful attack was found during the optimization process.

Black-box (ZOO-ADAM)	Success?	First Valid	Final L_2	Final Loss
All techniques	Yes	15,227	3.425	11.735
No Hierarchical Attack	No	-	-	62.439
No importance sampling	Yes	17,403	3.63486	13.216
No ADAM state reset	Yes	15,227	3.47935	12.111

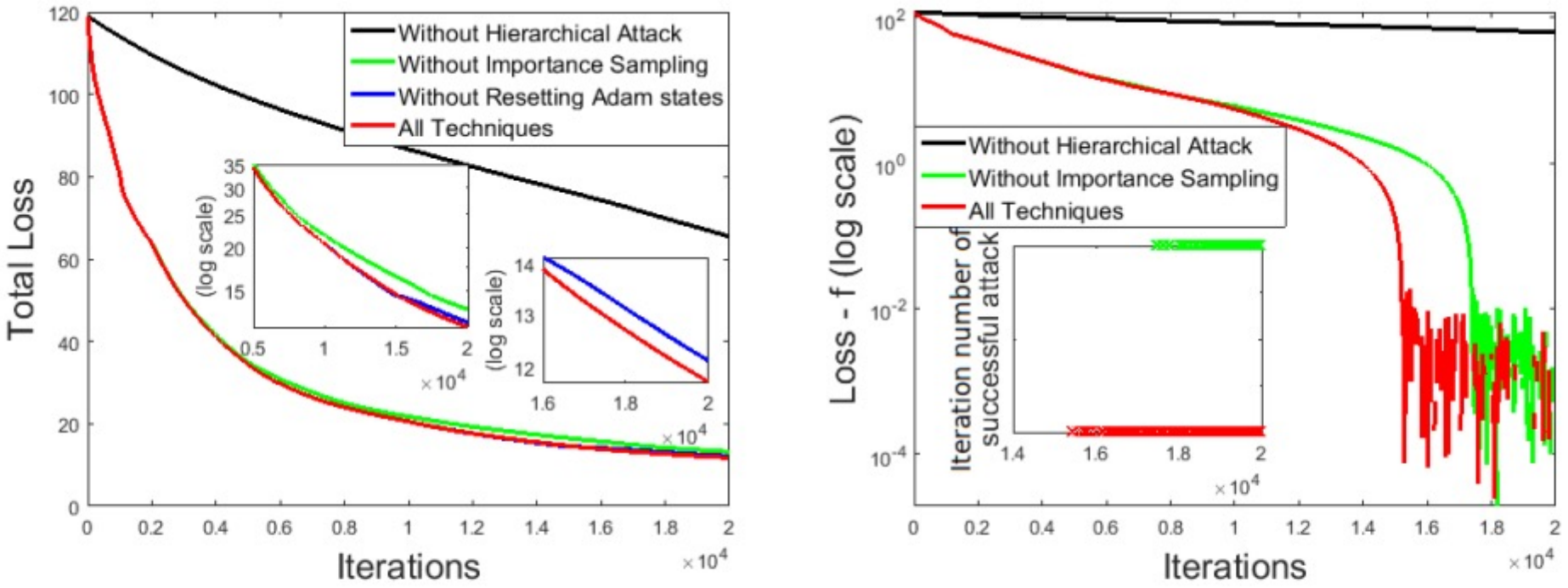


Figure 6: Left: total loss $\|\mathbf{x}-\mathbf{x}_0\|_2^2 + c \cdot f(\mathbf{x}, t)$ versus iterations. Right: $c \cdot f(\mathbf{x}, t)$ versus iterations (log y-scale). When $c \cdot f(\mathbf{x}, t)$ reaches 0, a valid attack is found. With all techniques applied, the first valid attack is found at iteration $15,227$. The optimizer then continues to minimize $\|\mathbf{x}-\mathbf{x}_0\|_2^2$ to reduce distortion. In the right figure we do not show the curve without resetting ADAM states because we reset ADAM states only when $c \cdot f(\mathbf{x}, t)$ reaches 0 for the first time.

Conclusion

- ZOO is a new type of attack to DNN without needing a substitute model
- Comparable results to C&W's white box attacks achieved
- The ZOO attack significantly outperforms the substitute-model based attacks

University at Buffalo The State University of New York

THANK YOU