DEPENDABLE AI PROJECT

Efficient Adversarial Attack Generation in Black-Box Settings

Kethireddy Harshith Reddy (B20AI018)
Gopathi Rahul Pramodh (B20AI009)
Tammireddi Sai Unnathi (B20AI045)
IIT JODHPUR

Introduction

The security and privacy of machine learning systems are greatly at risk from adversarial attacks. It is harder to think of strong adversarial instances in a black-box attack situation because the attacker is unaware of the target system. Efficient adversarial attack generation in black-box settings refers to the development of techniques for generating adversarial examples when the attacker has only limited knowledge of the target model. In this setting, the attacker has no direct access to the target model's architecture, parameters, or gradients but only to its input/output behavior. There is a need for a strategy that can effectively explore the input space and create high-quality adversarial examples in black-box attack scenarios because existing methods for creating adversarial examples in black-box scenarios may be ineffective or have low success rates.

In this project, we examined how well a trained Resnet-50 model can withstand the Boundary Biased Attack, and how well a defense mechanism known as "adversarial training" functions when this attack that we have implemented attacks it. We also compare the performances of various other attacks with this attack that we have implemented.

Implementation of Boundary Biased Adversarial Attack

We first investigate the robustness of the deep learning models against four different adversarial attacks. To start with, we have used ResNet-50 as the deep learning model and the CIFAR-10 dataset to fine-tune the ResNet-50 model. Boundary Bias is a recently proposed adversarial attack method that aims to generate adversarial examples that are closer to the decision boundary of the targeted model by adding a proximity term to the loss function. This proximity term encourages the attack to find adversarial examples that are closer to the decision boundary of the model, which can be useful in situations where query efficiency is a concern.

Mathematically, the loss function of the Boundary Biased Attack is defined as follows:

$$L(x, y, \theta) = c(x', y) + w * d(x', x)$$

where:

x is the original input image,

y is the true label of the input image,

θ is the target model,

c(x', y) is the cross-entropy loss between the predicted label of the perturbed image x' and the true label y,

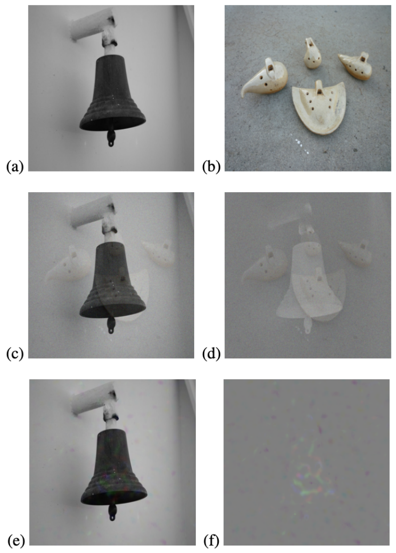
d(x', x) is the proximity term, which measures the distance between the perturbed image x' and the original image x,

w is a weighting factor that controls the importance of the proximity term.

The proximity term d(x', x) is defined as the L2 distance between the perturbed image x' and the original image x, normalized by the L2 norm of the original image x:

$$d(x', x) = \|x' - x\|_2 / \|x\|_2$$

During each iteration of the attack, the input image is perturbed by adding a small step in the direction of the gradient of the loss function with respect to the input image, subject to the constraint that the perturbed image remains within a certain L-infinity distance from the original image. The L-infinity distance is typically set to a small value, such as 0.03.



The images above show a targeted attack on ImageNet using 15000 queries, which are as follows:

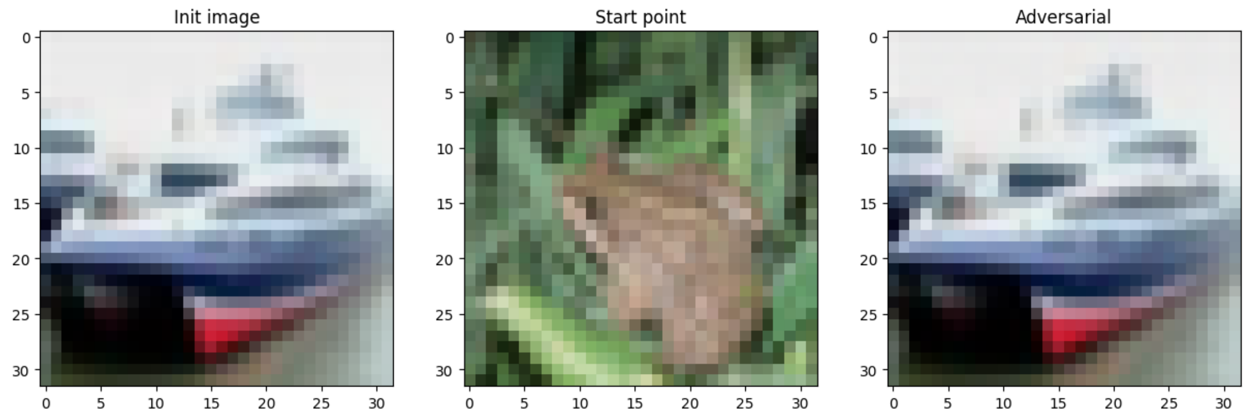
(a) the original image (bell) ,

(b) an image of the target class (ocarina),

(c) is an adversarial example generated by the original Boundary Attack (d` 2 = 18.52),

(d) showing the difference to the original image,

(e) and (f) show the biased Boundary Attack with all biases enabled. (d`2 = 4.78)



A small experiment was conducted on a single image where Initial Image the target class image that we want is given, as we can see in the above image, The initial image was perturbed to the target class that we wanted, and since we are also using regional masking and Perlin noise the frequency is very low for all the perturbations.

This approach can be useful for evaluating the robustness of machine learning models in realistic scenarios, where an attacker may only have access to a limited number of samples near the decision boundary.

Comparative performance analysis of different attacks

The following are the different adversarial attack mechanisms taken to attack the model and compare the results with those of the Boundary Biased attack that we have implemented:

White Box Attacks

PGD Attack (Projected Gradient Descent): PGD is an iterative adversarial attack method that tries to find the changes that make the model's loss function go up the most while staying within a certain number of changes. To make hostile examples, it does a lot of small updates along the direction of the gradient.

FGSM Attack (Fast Gradient Sign Method): FGSM is a one-step adversarial attack method that changes input cases by making small changes to the direction of the gradient of the loss function with respect to the input. Its goal is to do as much damage as possible and show bad behavior as quickly as possible.

Black Box Attacks

Boundary Attack: A boundary attack is an adversarial attack that starts at a random place in the input space and moves along the decision boundary, looking for the smallest change that will cause a misclassification. It looks at the shape of the decision line to find situations where people don't agree.

Hop skip jump Attack: A hop-skip-jump attack is a quick and agile way to attack that includes hopping, skipping, and jumping towards a target in quick succession. This method uses speed, coordination, and being able to get away from your opponent to beat them.

The following are the accuracies of the model after it has been attacked by these white box and black box attacks respectively:

White Box Attacks: (Attacks' names in column and epsilon values in row)

	0.001	**0.01**	**0.1**	**1**
PGD Attack	84.4%	9.4%	88.3%	0.0%
FGSM Attack	84.4%	18.8%	5.5%	4.7%

Black Box Attacks: (Attacks' names in column and epsilon values in row)

	0.001	**0.01**	**0.1**	**1**
Hop skip jump attack	79.7%	79.7%	79.7%	**28.9%**
Boundary Attack	95.3%	95.3%	95.3%	88.3%
Boundary Biased Attack (ours)	**41.0%**	**39.6%**	**37.1%**	34.2%

The above results of the attacks that have been performed on the model shows us that even with a very small magnitude of the sigma the White Box Attacks can easily degrade the models accuracy as all the models considered have access to the gradients of the model and they are designed in order to perform gradient ascent in a sense which have very efficient implementation that lead to degradation of the performance, where as the black box attacks don't have access to the gradients or outputs of the model and work on their respective principles which makes it much more difficult for the algorithms to actually cause the degradation of performance for lower epsilon values, and hence require higher values of epsilons to cause significant degradation of performance.

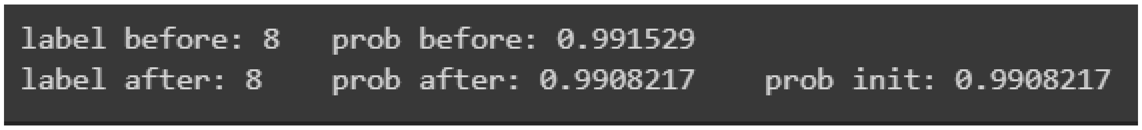
Defense mechanism against adversarial attacks

In the context of efficient adversarial attack generation in black-box settings, defense mechanisms refer to techniques that can be used to protect models against adversarial attacks. These mechanisms aim to make the model more robust to adversarial examples and prevent the model from being easily fooled by them. Some of the commonly used defense mechanisms include:

1. Adversarial training: This involves training the model on both clean and adversarial examples. By doing so, the model can learn to be more robust to adversarial attacks.
2. Input transformation: This involves transforming the input to the model in some way to make it more difficult for an attacker to generate adversarial examples. For example, adding noise to the input image can make it harder for the attacker to generate an adversarial example.
3. Detection-based defenses: These defenses involve detecting whether an input is adversarial or not. If the input is detected as adversarial, it can be discarded or handled in some way to prevent the model from making incorrect predictions.
4. Gradient masking: This involves modifying the model's architecture to prevent an attacker from easily computing the gradients required to generate adversarial examples.
5. Model ensemble: This involves training multiple models and using their predictions to make a final decision. This can make it harder for an attacker to generate an adversarial example that fools all of the models.

We have implemented Adversarial Training, and we tried to defend the model from the adversarial attacks that we have implemented above. We have observed that adversarial training was able to defend all attacks, including the Boundary Biased Attack, which we have implemented.
During adversarial training, the model is exposed to a variety of adversarial examples, which can make the decision boundary less well-defined and more irregular. This can make it difficult for BBA to generate adversarial examples because the boundary is no longer a clear separation between regions of the input space but rather a complex and potentially fragmented surface.

During adversarial training, the model is usually trained with a regularization term, such as adversarial loss or robust loss, which encourages the model to learn a smoother decision boundary that is less affected by adversarial changes. This can further complicate the task of BBA, as the perturbations that lie on the decision boundary may be less effective in generating adversarial examples.



```
label before: 8   prob before: 0.991529
label after: 8    prob after: 0.9908217    prob init: 0.9908217
```

The probability of labeling before and after is almost the same, which means Adversarial training was able to defend the Boundary Biased attack.

To conclude, The biased boundary attack has performed very well w.r.t the state-of-art Black box setting attacks, but this can be further taken care of and can make it undefendable by the adversarial training.

References:

1. https://github.com/matveymor/substitute_boundary_attack/blob/master/attacks/boundary_attack.py
2. https://arxiv.org/pdf/1812.09803.pdf

——————————————THE END——————————————