KLE Technological University

Creating Value
Leveraging Knowledge

School of
Computer Science and Engineering

Machine Learning Course Project Report
on
NIPS 2017: Adversarial Attack

By:

Dinesh Dathod	USN: 01FE17BCS069
Jyotiba Mane	USN: 01FE17BCS072
Gaurav Bhushan	USN: 01FE17BCS088
K Kruthika	USN: 01FE17BCS090

Under the Guidance of

Asst.Prof.Uday Kulkarni

Asst.Prof.Sunil V G

K.L.E SOCIETY'S
KLE Technological University,
HUBBALLI-580031

2017-2018

ACKNOWLEDGEMENT

We would like to express our sincere gratitude to all those people who have assisted us in the completion of this project, without whose help it could not have been possible. All their contributions are deeply appreciated and acknowledged.

We would like to take this opportunity to thank Dr. Ashok Shettar, Vice-Chancellor, KLE Technological University, Hubli.

We would also like to thank Dr. Meena S M, Professor, and Head of Department, School of Computer Science and Engineering for having provided the opportunity to extend our skills in the direction of this project.

We extend our heartfelt gratitude to both our guides, Mr. Uday Kulkarni and Mr. Sunil V G, School of Computer Science, whose valuable insights proved to be vital in contributing to the success of our project.

ABSTRACT

The purpose of this project is to perform an adversarial attack on a classication model, so that the classier fails in its classication. There are two type of attacks that can be performed on an classier.

Targeted Attack means, The images generated to fool the classier is carefully designed, that the classier will identify it as the attacker wants. That mean, The attack is targeted toward a particular class.

A Non Targeted Attack is when the image generated can be classied as any of the classes in the classier. There is no such particular class that the attacker is trying to target. The predicted class can be any class except that for its original class, basically the Non Targeted Attack is only about fooling the Classier.

In this paper, Non Targeted Adversarial Attack is discussed. And an attempt ot attack VGG16 has been made.

Contents

Chapter 1

INTRODUCTION

1.1 Preamble

This is a research based problem introduced by NIPS. The goal of the non-targeted attack is to slightly modify source image in a way that image will be classified incorrectly by generally unknown machine learning classifier. Deep learning systems are broadly vulnerable to adversarial examples, carefully chosen inputs that cause the network to change output without a visible change to a human. These adversarial examples most commonly modify each pixel by only a small amount and can be found using a number of optimization strategies. Other attack methods seek to modify only a small number of pixels in the image (Jacobian-based saliency map), or a small patch at fixed location of the image. Adversarial examples are inputs to machine learning models that an attacker has intentionally designed to cause the model to make a mistake; they are like optical illusion for machines.

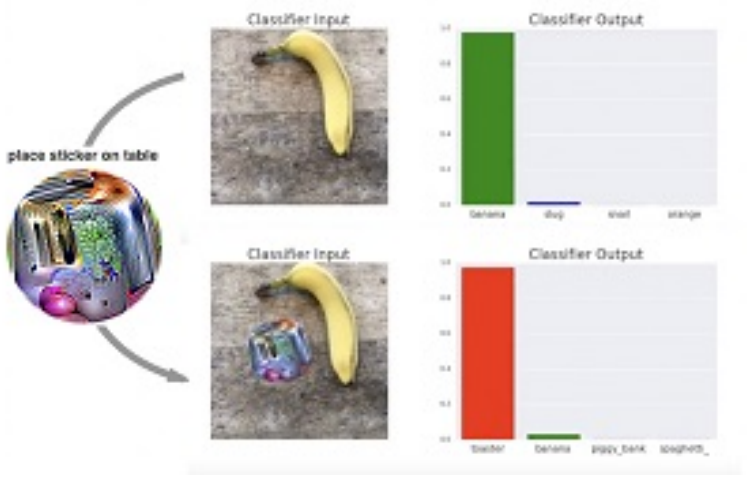


Figure 1.1: Adversarial example

1.2 Motivation

Adversarial examples are a good aspect of security to work on because they represent a concrete problem in AI safety that can be addressed in the short term, and because xing

them is dicult enough that it requires a serious research eort.

1.3 Objectives

1.Generate images that can perform perform non targeted adversarial attack
2.Fool the classifier

1.4 Literature Survey

In 2012, with gain in computing power and improved tooling, rapid growth in Machine learning and AI applications was observed. Adversarial Attack: Adversarial examples are inputs to machine learning models that an attacker has intentionally designed to cause the model to make a mistake; they are like optical illusion for machines. In 2014, a group of researchers at Google and NYU found that it was far too easy to fool ConvNets with an imperceivable, but carefully constructed nudge in the input.

Papers referred:
1. Boosting Adversarial Attacks with Momentum
Authors:Yinpeng Dong1, Fangzhou Liao1, Tianyu Pang1, Hang Su1, Jun Zhu1, Xiaolin Hu1, Jianguo Li2 1 Department of Computer Science and Technology, Tsinghua Lab of Brain and Intelligence 1 Beijing National Research Center for Information Science and Technology, BNRist Lab 1 Tsinghua University, 100084 China 2 Intel Labs China
Insight:This paper gave information about the methods used to attack a network. [1]

2. Explaining and harnessing adversarial examples
Authors:Ian J. Goodfellow, Jonathon Shlens, Christian Szegedy
Insight:This paper gave information about dfast method of generating adversarial examples. [2]

3.One Pixel Attack for Fooling Deep Neural Networks
Author:Jiawei Su*, Danilo Vasconcellos Vargas* and Kouichi Sakurai
Insight:This paper deals with the minimum amount of change made in an image that can fool the classifier.[3]

4.AdversarialPatch
Author:TomB.Brown,DandelionMané,AurkoRoy,MartínAbadi,JustinGilmer
Insight:In this paper we learnt about Adversarial Patches, that how adding some noise or tweaking pixel values can create adversarial patches to the image that can fool the classifier. [4]

1.5 Problem Definition

To slightly modify source image in a way that image will be classified incorrectly by generally unknown machine learning classifier.

Chapter 2

PROPOSED SYSTEM

2.1 Proposed System

Step1: Feeding the test images to the Adversarial Image Generator.
Step2: Attack the trained classifier with the generated images.

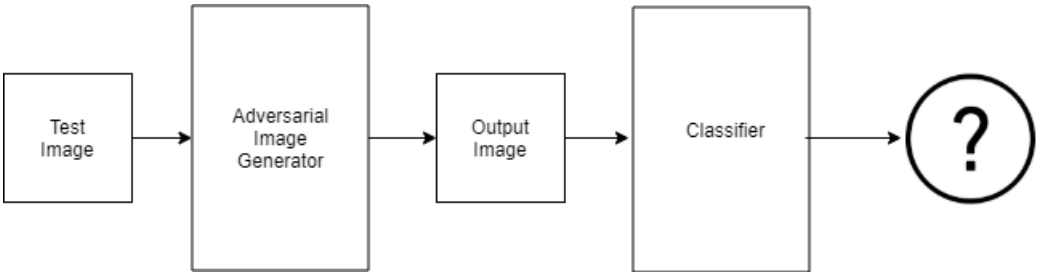


Figure 2.1: Block diagram

2.2 Advantages

If one knows what images the classifier classifies, one can take such images and generate adversarial designed images, that can be given to the classifier and it may be able to fool the classifier.

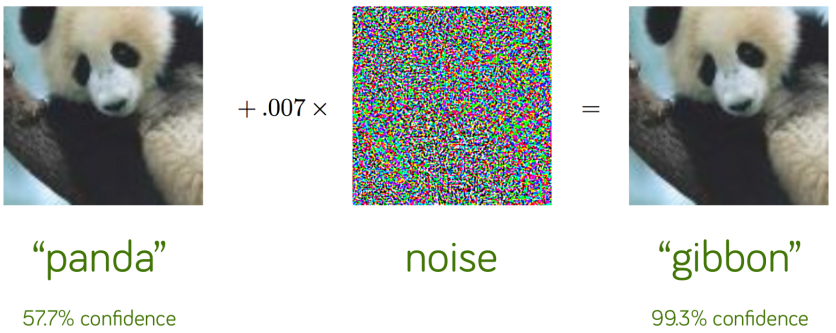
2.3 Scope

These attacks shows the vulnerability of the classifier present today and they will inspire the neural network designers and classifiers designers to come up with such solutions that makes our network and classifiers more robust towards such attacks and make correct predictions each and every time.

Chapter 3

IMPLEMENTATION

The data-set given had 1000 random images without labels. On feeding those images to the Adversarial Image Generator model that designs those images using the following equation:



Suppose we have a machine learning model

$$y = ax + b \tag{3.1}$$

The model parameters are a and b. The loss function would be

$$L(x, y, a, b) = (\mathbf{y} - (\mathbf{ax} + \mathbf{b}))^2 \tag{3.2}$$

Here the loss is the squared difference of real y and and (ax+b). We compute the derivative of our function L according to the parameters of model a and b.

$$(dL/da)=2x(ax+b-y) \tag{3.3}$$

$$(dL/db)=2x(ax+b-y) \tag{3.4}$$

Then the values of the parameters are updated in order to minimize the loss.

This is how a general gradient descent works. But if the model is fixed, we cant change the parameters and we need to increase the loss. One thing that can be done is that modify the data points, changing Y doesn't make sense so we change the input that is xs. Yes we have to make changes in the inputs, but that should not be observable that is the change should be very slight and effective. We can compute the derivative of the loss function according to x.

$$(dL/dx) = 2a(ax + b - y) \tag{3.5}$$

Now, we update the value of x by a very small amount accordingly. The loss will increase and as the modification is very small, the perturbation is hard to detect.
When we enter the neural network to classify images, here there is not much difference than what we had discussed, the loss function here is cross entropy, the model parameters are the weights of the neural network and the inputs are the pixel values of the image.

Let x be the original image and y is the class of x, θ the weights and J(θ,x,y) the loss to train the network.

$$\delta \mathrm{J}(\theta,\mathrm{x},\mathrm{y})$$

We calculate the gradient loss function according to the input pixels.

$$X' = x + \epsilon sign(xJ(x,y)) \tag{3.6}$$

Here ϵ is the small value so that we dont go too far from the loss function surface and that the perturbation will be imperceptible.
X' is x but with added perturbation.
Whatever we have seen till now is just one iteration we need to do the same for multiple iterations.

Chapter 4

RESULTS AND DISCUSSIONS

The results are shown below.
On the left, original images are placed and on the right are designed images by the Image Generator. When both images were given to the classifier, the original image is classified correctly but classification for the image on right is wrong.

Figure 4.1: Example 1

Figure 4.2: Example 2

Figure 4.3: Example 3

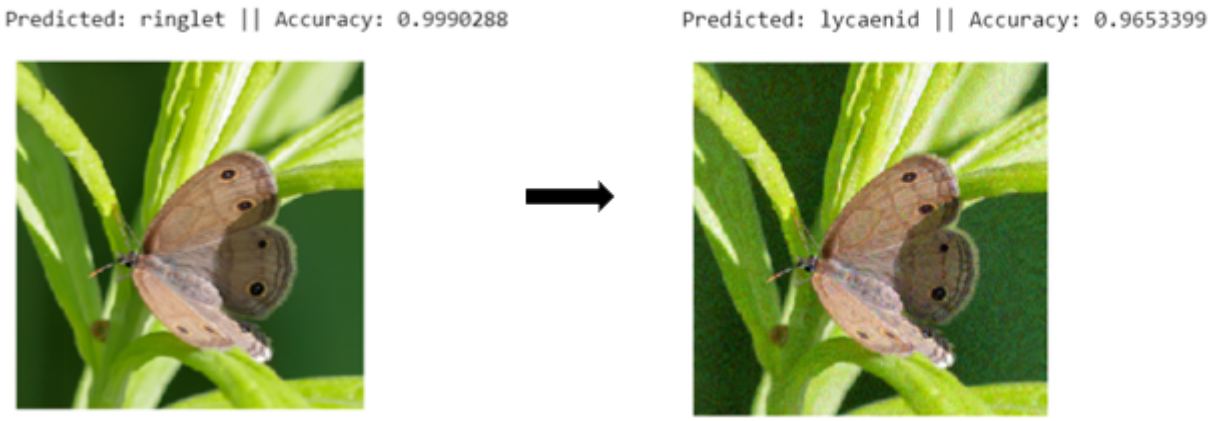


Figure 4.4: Example 4

The given graphs shows that out of 1000 generated images, How many images were classifies correctly represented at 0, that is the attack failed and How many images with noise were wrongly predicted represented at 1, that is the attack was successful.

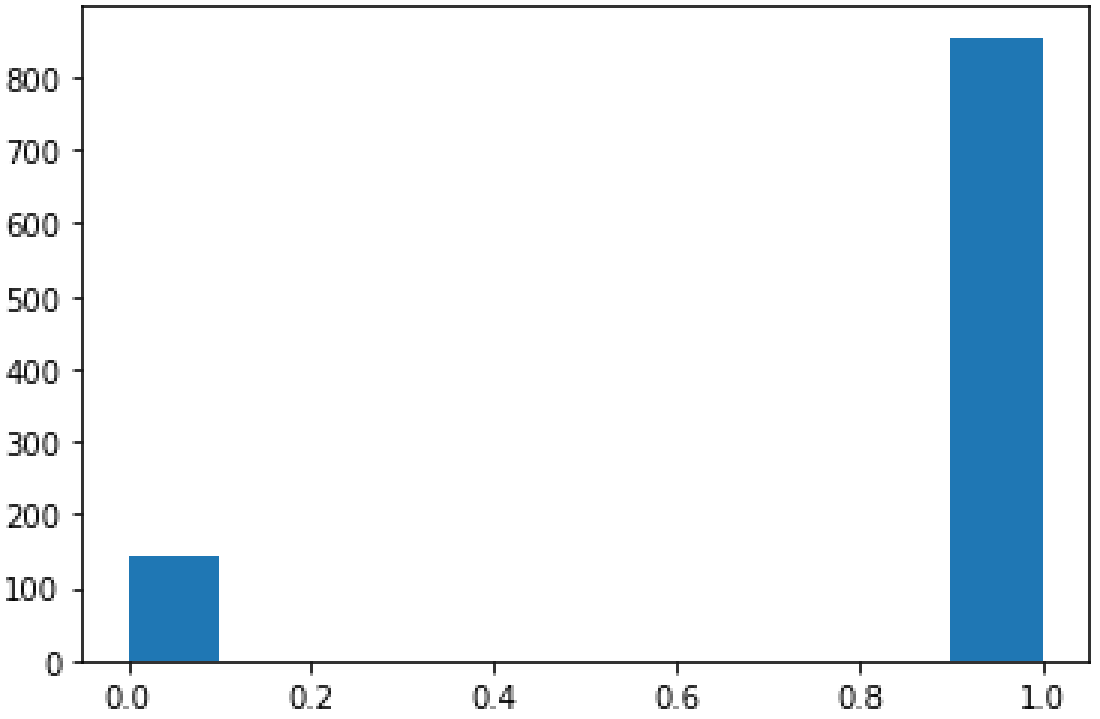


Figure 4.5: Prediction 1

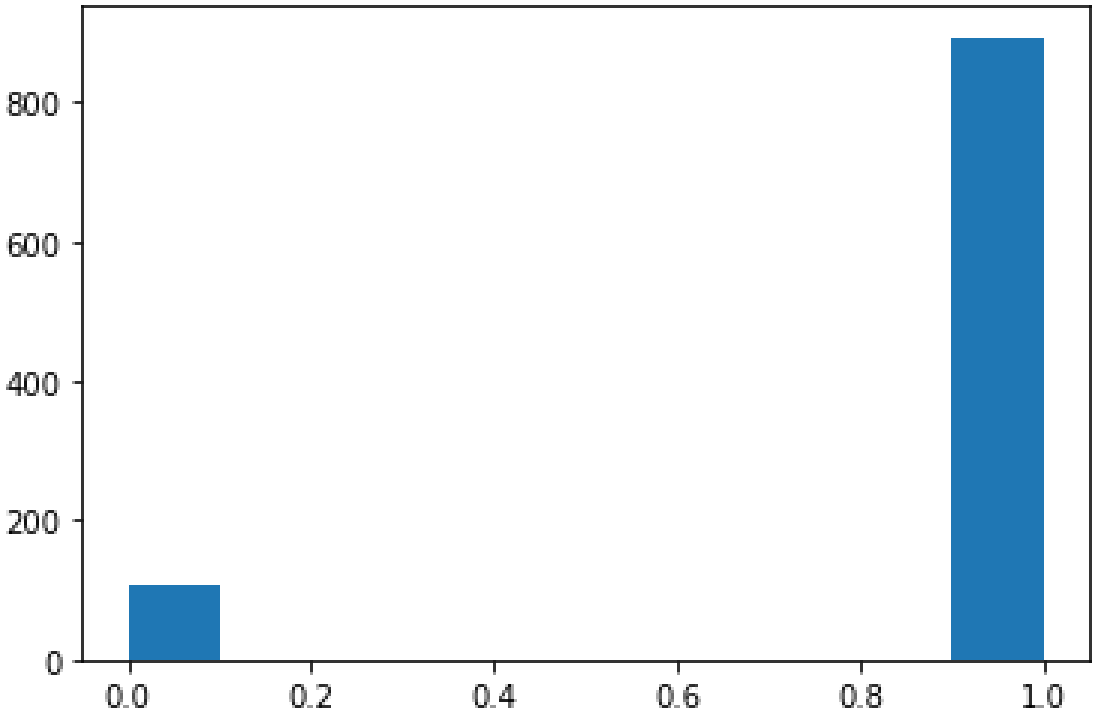


Figure 4.6: Prediction 2

Chapter 5

CONCLUSION AND FUTURE SCOPE

We are successfully able to attack the VGG16 model that has been trained on the imageNet data set.
Scope for these models are that, If these models get better at attacking networks. Our networks will also have to be more robust towards such attacks. This will help to test the limits of a classification model. And as the attacks are getting stronger, the defence system will also improve. This will be very helpful in medical field where there is no scope for error.

Bibliography

[1] Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su1, Jun Zhu, Xiaolin Hu, and Jianguo Li. Boosting adversarial attacks with momentum.

[2] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples.

[3] Jiawei Su, Danilo Vasconcellos Vargas, and Kouichi Sakura. One pixel attack for fooling deep neural network.

[4] TomB.Brown, DandelionMané, AurkoRoy, MartínAbadi, and JustinGilmer. Adversarialpatch.