Lab A - Adversarial Attack on Malware Classifier Student Document

Table of Contents

Lab Descriptors

Objectives

The student will learn how to implement an adversarial attack using the Fast Gradient Sign Method (FGSM) against a convolutional neural network that has been trained to classify malware samples as images.

Assumptions

The student has basic knowledge of virtual machines, Debian Linux, the Python language, and some knowledge of machine learning.

Outcomes

The student will be able to create adversarial samples that disrupt a CNN model given a sample non-adversarial input image.

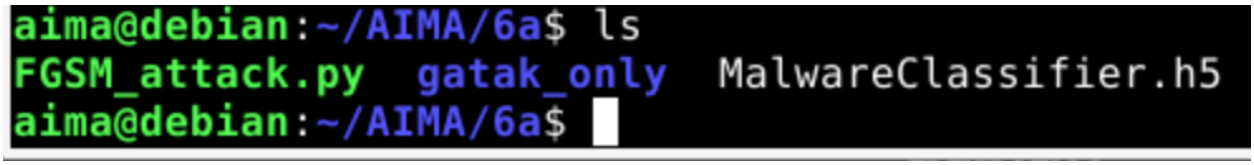
Introduction

In this lab, students will implement an adversarial attack against a pre-trained malware classifier using the Fast Gradient Sign Method (FGSM). This classifier is a convolutional neural network (CNN) that classifies inert images of malware samples into eight families of malware. **There is no live malware in this lab.** The samples used in this lab are sterilized malware samples from the Microsoft Big 2015 Dataset linked in the resources section at the bottom.Students will construct an adversarial sample of its expected input in order to attack it. This will be done in a Debian Linux virtual machine in Python, making use of the Keras, Tensorflow, and scikit-learn libraries.

Lab Details

Environment

1. The username for this lab is **aima** and the password is **toor**. The working directory for the lab is **/home/aima/AIMA/6a**.



Materials

The lab materials are all contained in the **AIMA/6a** working directory discussed in the Environment section above. Within that, you will find the files and directories discussed in this section. Do your work in place as these files depend on one another.

1. The **gatak_only** directory tree contains the image of a malware sample in the format that the model uses to verify labels. This is an image of a sterilized malware sample's bytecode read in as a grayscale .PNG image. This sample is inert and safe to work with and view inside the VM. The model is trained to classify samples from eight families: Gatak, Kelihos_ver1, Kelihos_ver3, Lollipop, Obfuscator_ACY, Ramnit, Tracur, and Vundo. You are only provided with one Gatak sample for this lab.

2. The **MalwareClassifier.h5** file is the pre-trained export of the convolutional neural network model you will be working with. Do not edit this file as you will not be able to complete the lab without it.

3. The **FGSM_attack.py** script is the python script in which you will do all of your work. It may be a good idea to make a backup copy of this before you start making changes to it.

Adversarial Attack

1. Navigate to the working directory and run the FGSM_attack.py script. If you are doing this from the command line, run **python3 FGSM_attack.py** from the working directory. This will feed the model an unedited 64x64 sample of the gatak malware's texture. It will open a window to display the image and its prediction as a plot.

    1.1. Knowing that the input was an unedited Gatak sample, did the model correctly classify it? Record your answer to your answer document.

    1.2. Either save the image from the plot or take a screenshot of it and embed this in your answer document. It may be easier to take screenshots from your host computer than it will be to save the plots on the VM.

2. Close any plot windows to allow the script to finish running. Now open the FGSM_attack.py script in a text editor. If you are doing this from the command line, nano or vim will work fine.

    2.1. Uncomment line 91 for "plt.show()", save the changes and run the script again. This will observe the loss from the model's classification of the Gatak image sample and create an adversarial pattern with respect to that loss.

    2.2. The adversarial pattern will be plotted so you can see it. Take a screenshot and embed it in your answer document. Close any plot windows to allow the script to finish running.

3. Now that you have the base image and an adversarial pattern, you are ready to make an adversarial sample.

    3.1. Open the script again and go to line 98. This epsilon value will determine how much of the adversarial pattern that you generated in step 2 will change the base image. For now, leave it at 0.15.

    3.2. Go to line 129 and uncomment plt.show(). Save the changes and run the script again to see the completed adversarial image and the model's prediction on it.

        3.2.1. Knowing that the base image was a Gatak sample, did the model classify the adversarial sample correctly?

        3.2.2. If not, what family did the model misclassify it as? What was its confidence for that prediction?

        3.2.3. If you run this test more than once, does the model always produce the same prediction with the same confidence on this adversarial sample?

3.3. Take a screenshot of the model's prediction on the adversarial sample and embed it in your answer document. Close all plot windows to allow the script to finish executing.

Fine-tuning the Adversarial Attack

Now you are ready to try your own values for epsilon to fine-tune this adversarial attack. Remember, the goal of an adversarial image is to maximize disruption to the model while minimizing visible changes to the image.

1. Open the **FGSM_attack.py** script in a text editor and change the value of epsilon on line 98 to an epsilon value that you would like to test. These values should be between 0 and 1, with larger numbers representing a heavier weight for the adversarial pattern's effect on the base image. Two decimals of precision is sufficient for this lab. Save your changes and run the script between changes of this epsilon value to test. A range of values will be accepted here.

2. Make sure to look at the visual representation of your adversarial image between trials. While it is not hard to choose a large value for epsilon that will trick the model, it is more difficult to choose a value that is small enough to not cause overt visual changes to the image while still fooling the model.

3. What is your best value for epsilon?

    3.1. Do you see visual differences between your adversarial sample and the original?

4. Did the model misclassify your adversarial sample?

    4.1. If so, what malware family did it predict and with what confidence?

5. Insert a screenshot of your best adversarial image in your answer document.

Deliverables

You will turn in a PDF with your name at the top and your answers to the lab questions and embedded screenshots from each step of the lab.

Resources

Virtualbox Documentation

https://www.virtualbox.org/wiki/Documentation

Microsoft Big 2015 Dataset

https://www.kaggle.com/c/malware-classification/data

Python Documentation

https://www.python.org/doc/

Linux manual pages

https://linux.die.net/man/

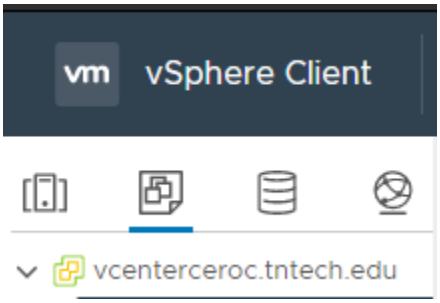
Tensorflow FGSM Tutorial

https://www.tensorflow.org/tutorials/generative/adversarial_fgsm

Accessing Vsphere

To access the Vsphere from Tennessee Tech's network, follow the steps below. If you are accessing the Vsphere from off-campus, install the school VPN from the ITS help site here: https://its.tntech.edu/display/MON/Off-Campus+VPN+Installation, then follow the same steps.

1. Go to https://vcenterceroc.tntech.edu/.
2. Your browser may display a message that the connection is not secure. This is fine. Click the advanced option and proceed to the site anyway.
3. Click the button that says "Launch Vsphere Client (HTML5)".
4. Login with your tech username and password. This username will be the first part of your university email address before the '@' symbol.
5. From here, select the second icon on the upper left menu.



6. Expand the tree under CEROC-Datacenter to view virtual machines that you have access to.
7. Select the virtual machine for this lab and click "Launch Web Console". This will open a new tab from which you can interact with the VM.
8. For this set of labs, the username is "**sstudent**" and the password is "**toor**".