ADVANCE Labs - Adversarial Attack on Cyberbully Detection Models Lab

1 Lab Overview

In this lab, you will manipulate an artificial intelligence/machine learning (AI/ML) algorithm trained to detect cyberbullying on images to output the wrong prediction. The process of fooling the trained models is known as an adversarial attack. Cyberbullying is bullying performed via electronic means such as mobile/-cell phones or the Internet. The objective of this lab is for students to gain insights into adversarial attacks and how they cause a cyberbullying detection algorithm to produce incorrect output.

In this lab, students will follow the instructions provided in a Jupyter notebook. This notebook contains an algorithm that detects cyberbullying, an algorithm that can fool cyberbullying detection algorithms, and open-ended questions.

Content Warning: This lab contains some inappropriate images. We minimized showing such content samples in this lab. They do not represent the views of the authors.

2 Lab Environment

This ADVANCE lab has been designed as a Jupyter notebook. ADVANCE labs have been tested on the Google Colab platform. It would be best if you used Google Colab since it has nearly all software packages preinstalled, is free to use, and provides free graphical processing units (GPUs). You can access this lab in here.

3 Lab Tasks

3.1 Getting Familiar with Jupyter Notebook

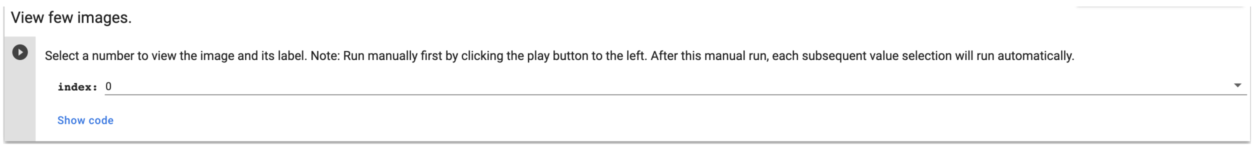
The main objective of this lab is to learn how to perform adversarial attack on models trained to detect cyberbullying. Before proceeding to that, let us get familiar with the Jupyter notebook environment.

Jupyter notebooks have a Text area and a Code cell. The Text area is where you'll find instructions and notes about the lab tasks and the Code cell is where code written. You won't be interacting with code in this lab, however, you will be executing code.

In this lab, you will be executing code cells and selecting values from a drop-down menu in code cells. There are two tasks to be completed indicated. Two tasks are to be completed in the notebook provided, indicated by ”Task #” where # is the task number. You must complete these two tasks, plus the discussion at the end of this document, to complete this lab successfully. To execute a code cell, click the play button on the left of the cell. An example of a code cell and a play button is shown in Figure 1. If you mistakenly click on the “Show code” button, don't worry about it. It only expands the code cell and makes the code visible.

Figure 1: Example of a code cell and play button.

There are code cells that contain a drop-down menu for selecting different values. To execute such cells, click the play to the left to run the cell with the pre-selected value. Subsequent value selection will be executed automatically for you. Figure 2 shows an example of such a code cell.


View few images.

Select a number to view the image and its label. Note: Run manually first by clicking the play button to the left. After this manual run, each subsequent value selection will run automatically.

index: 0

Figure 2: Example of a code cell with a drop-down menu.

Get started. After going through this manual, go through the notebook, follow the instructions and run the code cells.

3.2 Adversarial Attack on Cyberbullying Detection Model

You will attack a cyberbullying detection model already trained on not-safe-for-work (NSFW) images in this lab. This model is based on the ResNet50 architecture. Your task is to make this model predict NSFW images as safe for work, i.e., make the model predict that the image is non-NSFW. You will use a small NSFW dataset we collected to test your attack. You must observe how an adversarial attack and a control value known as epsilon affect the model prediction accuracy and document your observation. You can access the lab by clicking here.

3.2.1 Dataset

We provide an NSFW dataset having five categories, drawings, neutral, hentai, porn, and sexy. Drawings and neutral images are safe for work. Hentai, porn, and sexy (explicit content that is not pornography) are images not safe for work. You can learn more about these labels in the official NSFW data page.

3.3 Adversarial Attack

Deep neural network algorithms are vulnerable to adversarial attacks despite their high accuracy in various tasks. This vulnerability usually results in a little change to the input that causes the network to predict an incorrect output. This small change is often imperceptible to human vision. Figure 3 shows an example of a perturbed cyberbully image on the right and the original image on the left. Observe how close the perturbed example is to the original and the high prediction probability (86.5%) assigned by the model. We will generate such perturbations in this lab using the fast gradient sign method (FGSM). FGSM is briefly described below.

An adversarial attack can be categorized into white-box and black-box attacks. In a white-box attack, the adversary knows information about the target model, like model parameters, architecture, and training data. The adversary does not have information about the target model in a black-box attack. We will perform a white-box attack in this lab since we know the model architecture, its parameters, and training

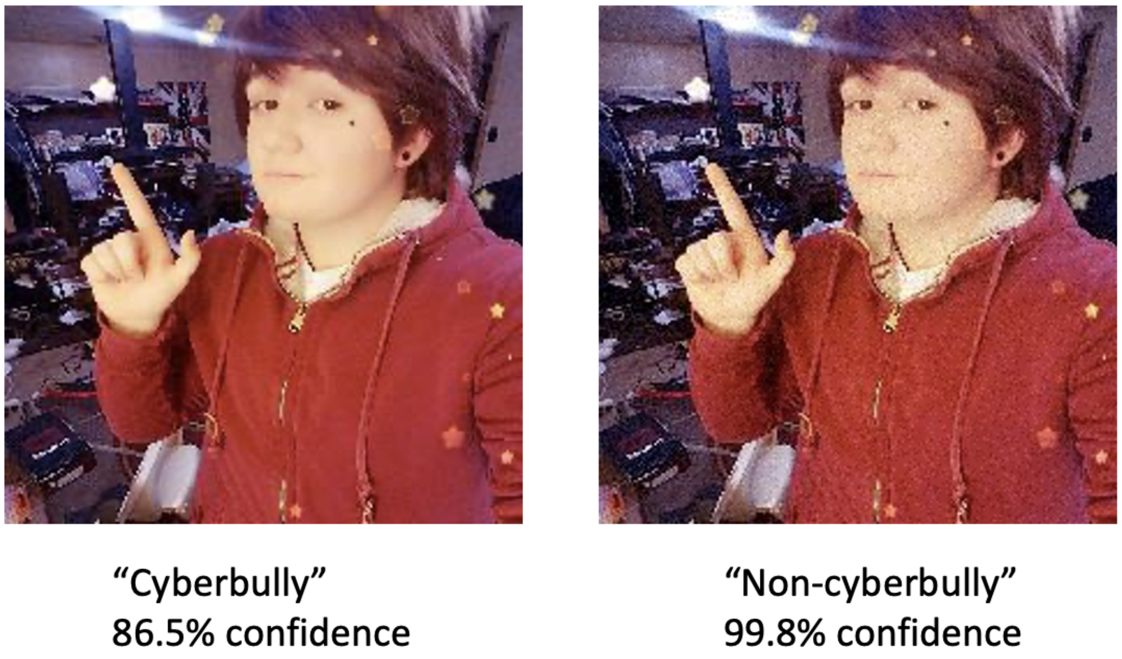


Figure 3: Example of perturbed input used for adversarial attack.

data. Adversarial attacks can be targeted or untargeted. In a targeted attack, the model is fooled not to predict a specific label. In contrast, in an untargeted attack, we do not care about a specific label as long as the prediction is incorrect.

3.3.1 Fast Gradient Sign Method (FGSM) Attack

The fast gradient sign method is a fast method for generating adversarial examples. Given a clean image with no perturbation, FGSM efficiently finds an adversarial perturbation (noise) that, when added to the clean image, maximizes the algorithm's prediction error loss leading to misclassification. Consider an adversarial example (perturbed image) $x^{'}$ defined as: $x' = x + \delta$ where x is a vector of a clean image, δ is a small noise (perturbation) added to the clean image to cause a misclassification. FGSM finds the small noise δ capable of causing misclassification.

3.3.2 Task 1: Cyberbullying detection without an attack

Run the code cell under *Execute Without FGSM Attack* by clicking the play button to the left of the cell. In order to get the correct output you must have executed all code cells preceding this cell. Observe the output graph and report your observation in terms of accuracy

3.3.3 Task 2: Cyberbullying detection without an attack

Run the code cell under *Execute FGSM Attack* by clicking the play button to the left of the cells. You must execute the attack using different control (epsilon) values for this task. Note that in the code cell with a drop-down menu, you must first click the button to the left and wait for the execution to finish. Then choose a new value from the drop-down, which will cause the cell to be executed automatically. You will have to choose each value in the drop-down before executing the visualization and sample example code cells. Answer the question in Task 2 of the notebook.

3.3.4 Discussion

- How does the FGSM attack work?.
- What potential problems do you envision if machine learning systems like this are used in real-world situations?
- Given that the epsilon values used in this lab have been cheery picked, which if the values would you choose and why?

4 Submission Instructions

You need to submit a detailed lab report, with screenshots, to describe what you have done and what you have observed. You also need to provide explanation to the observations that are interesting or surprising.