Deep Learning Security **Experimental Assignment #2** **Mar. 4, 2023**

Adversarial Examples

Overview

- This assignment has two major parts: i) generating adversarial examples to fool a convolutional neural network (CNN), and ii) adversarial training to defend against adversarial examples.
- **Assignment goals:**
  - Become familiar with adversarial examples
  - Understand how adversarial examples can be generated
  - Understand one technique for how to defend against adversarial examples
  - Continue developing skills in report writing
- **Deliverables** (due Apr. 5, 2023, at 5 pm)**:**
  - Report (approx. 4-6 pages)
  - Code (Drive links in the report or separately submitted file)
- **Expected Work Load:** 20 hours to do great work, 15 hours to get by. *Note: this does not include down time while experiments run. Please start early to account for this!*

What to do

1. Get the code from this Google Drive (link also below). Install the dependencies (even if you are using Google Colab). You can find both Python script and a Jupyter Notebook version of the code in the drive. Run the code and check whether everything is working properly. This code will train a CNN model with MNIST using the PyTorch deep learning framework. The test accuracy of this model should be 99%. Feel free to tune the model if you want to push the accuracy higher.
2. Generate adversarial examples (AE) of the MNIST test samples using the Fast Gradient Signed Method (FGSM: https://arxiv.org/abs/1412.6572) with different perturbation values ε.
   a. You may follow a tutorial such as this one to get your attack working: https://pytorch.org/tutorials/beginner/fgsm_tutorial.html
   b. Use ε = [0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9]. Randomly pick 10 test samples from each class and generate adversarial examples for those samples, i.e., generate 100 samples for each value of ε.
   c. Plot the accuracy vs. ε values for the trained model on your adversarial samples.
   d. Pick one class and use it to show one randomly selected adversarial sample for each value of ε.
3. Perform adversarial training to defend against your attack.
   a. Do not use any tutorial for this part of the assignment. Write your own code.
   b. MNIST training set has 60K training samples. Generate adversarial examples for each of the training samples using ε = [0.05,0.1,0.2,0.25,0.3]. *Note that 0.15 is omitted on purpose*.
   c. Train both a new (untrained) model **AND** the original (already trained) model with all 60K benign + 300K adversarial samples. Plot your training loss and your models' performance on benign test samples.
      i. Note: Do not change any of the hyperparameters of the model (explain why in your report), but you can train for more epochs to reach to a reasonable level of performance.
   d. Generate adversarial test samples using ε = [0.05,0.1,0.15,0.2,0.25,0.3,0.35,0.4] from 10 randomly selected test samples for each class. Plot the accuracy vs. ε values for the two adversarially trained models on benign samples and, separately, your adversarial test samples.
4. Repeat Steps 2 and 3 with another attack:
   a. Carlini-Wagner, using any of the three Lp-norms:
      https://www.nicholas.carlini.com/papers/2017_sp_nnrobustattacks.pdf
      https://fairyonice.github.io/Learn-the-Carlini-and-Wagners-adversarial-attack-MNIST.html
   b. PGD: https://arxiv.org/pdf/1706.06083.pdf
      https://adversarial-ml-tutorial.org/adversarial_examples/
   c. IGSM: https://arxiv.org/pdf/1607.02533.pdf – just the iterative version of FGSM
      https://deepnotes.io/adversarial-attack
   d. JSMA: https://arxiv.org/pdf/1511.07528.pdf

https://deepnotes.io/adversarial-attack

e. Note: you may need to substitute something for changing ε, and note that the amount of perturbation may be lower and still get effective results, so be intelligent here.
f. Assuming that you follow a tutorial or other code, cite it clearly both in the code and in your report.
g. In Step 3 on FGSM, you will find that only one model (new or original) will work. For this attack, you only need to train in the way that will work, not both.

5. Let's see how well the adversarial training generalizes to another attack. Evaluate both the robust FGSM-trained model on the new attack's adversarial examples and the new-attack-trained model on the FGSM adversarial examples.

https://drive.google.com/drive/folders/1d43ZWPFDHcy8qo4fvMmmJI-B275cjE39?usp=sharing

If you have trouble getting the MNIST data, you may try: https://github.com/cvdfoundation/mnist

Instructions on mounting your Google Drive: https://colab.research.google.com/notebooks/io.ipynb

Expected Report Elements

- The report should emulate the Experiments and Results sections of a high-quality research paper.
- Provide a short introduction to the work, but do not provide details on the base models or a long description of the research problem – just refer to other resources.
- Results and analysis of the experiments you ran
    - Have answers to important questions such as:
        - *How did epsilon effect accuracy?*
        - *How did the accuracy on both benign and adversarial samples change with the adversarially trained models?*
        - *How did the models compare? Why does this happen?*
        - *Were your results as expected? Why or why not?*
    - Reference additional papers that support your findings.
- Provide the graphs and tables as described above
- Include any citations in an appropriate and consistent format
- Include a python or Jupyter Notebook file of your code
    - Clearly mark any code obtained elsewhere (with a citation), and any code that you wrote yourself.
    - Document and comment your code clearly (and/or use very easy-to-read code)

Code Submission

- Submit your LSTM code, including any data processing code.
- Submit your CNN code, highlighting areas that you have changed.
- Ensure that your code is commented, and provide instructions for how to run it.

Grading Checklist

Code Requirements

- ☐ Clear instructions for how to run it are provided
- ☐ It runs as expected
- ☐ Code is reasonably easy to read and follow
- ☐ Variable names are generally meaningful
- ☐ Code includes comments to help the reader understand more complex parts

Report Requirements

Introduction section

- ☐ Is between 100-150 words

- ☐ Briefly explains AEs and adversarial training
- ☐ Includes at least four citations to the literature on AEs and adversarial training
- ☐ Briefly explains the point of the report
- ☐ Briefly mentions two or three key findings from the results

Adversarial Examples section
- ☐ Briefly describes the baseline model
- ☐ Briefly describes FGSM and the other attack you select
- ☐ Describes any attack parameters that were evaluated, and the values/ranges tried
- ☐ Includes a graph for accuracy vs. ε values for both attacks, starting with $\varepsilon = 0$ for a baseline
- ☐ Shows 10 samples from the same class for both attacks
- ☐ Describes the findings clearly
- ☐ Almost all the statements seem to be correct

Adversarial Training section
- ☐ Describes the adversarial training procedure in enough detail that another researcher could reproduce it
- ☐ Includes a graph for accuracy vs. ε values for both FGSM-trained models and the model trained on the other attack, starting with $\varepsilon = 0$ for a baseline
- ☐ Includes the same graphs for cross-attack testing (just one for attacking the FGSM model that actually works, not the one that doesn't)
- ☐ Describes the key findings clearly
- ☐ Almost all the statements seem to be correct

Analysis section
- ☐ Addresses all questions described in **Expected Report Elements**
- ☐ Includes at least three different relevant citations
- ☐ Almost all the statements seem to be correct

Writing & Graphs Quality
- ☐ Graphs are well designed, with sufficiently large fonts, good use of color, well-labeled axes, and a clear key
- ☐ Report consists of well-formed sentences that are usually grammatically correct
- ☐ Sections and paragraphs are organized to form a coherent structure of thought
- ☐ Each point flows smoothly to the next one for easy reading and clear structure
- ☐ The document is nicely formatted for easy reading and understanding, including clear section headings
- ☐ The writing is mostly free of spelling and other minor mistakes, such that they don't significantly distract from the arguments being made
- ☐ Citations are well formed

A Note on Plagiarism

When writing, you must include in-line citations whenever you are reporting on information that is not "general knowledge," i.e., anything you learned for this project and didn't know in advance. This is **NOT** just for quoted information. Failure to do this is plagiarism.

This article on plagiarism is good and covers the line between common knowledge and other material https://writingcenter.unc.edu/tips-and-tools/plagiarism/

Also: https://www.plagiarism.org/ has a ton of additional information.

On Paraphrasing: It is not good enough to just change some words around from the original document. For example: "It is insufficient to merely alter some words from the source document." ← That is plagiarism of the first sentence, **even if you cite the source**. To avoid this, you should take the notes you need from the original without copying, and later write in your paper what you learned from your notes. Repeat as necessary to understand the source material well enough to write about it.

Tip: are you sure you need that information in your paper? Many cases of plagiarism like this are from people who are trying to add an explanation that simply isn't required for their paper.

On Diagrams: You must not copy a diagram or image from another paper, even if you cite it. There is no way to properly "quote" a diagram. If you are sure that the diagram will help an important explanation in your paper, such that just directing the reader to the source is not enough (*Tip: in many such cases, that **is** enough*), then you must make your own version of the diagram AND cite the paper like this: "Fig. 1. Diagram of …. (reproduced from Foo et al. [cite])."