Assignment 1: Adversarial Example and Backdoor Attack

Deadline: April 22nd at 11:59 pm

Overview

The goal of this assignment is to introduce students to the concepts of AI security with a focus on the attacks.

This assignment is broken into **4** parts. (1) Part 1 involves familiarizing and training neural networks in PyTorch. (2) Part 2 involves implementing a **white-box adversarial example attack** using the Projected Gradient Descent Method (PGD). (3) Part 3 involves implementing a **simple backdoor attack**. (5) Part 4 involves some questions about large language models.

We will provide notebooks for students to work on the assignment with solution submission to Gradescope. We ask that students **not attempt to perform out of scope attacks that attack the grading infrastructure** such as DDoS attacks or exfiltrating non-publicly released assignment components. It is **recommended that students develop and test exploits locally** first before submitting to the autograder. However, **we encourage students to find bugs and issues with the assignment application**! If students identify any additional vulnerabilities in the application, please privately message the course instructors. 😀

Assignment Code: See below.

Installation

Google Collab

For this assignment, we have provided machine learning notebooks that can be accessed through Google Collab. Google Collab is a service which allows training and interfacing with data science notebooks. It has all required machine learning libraries needed for this assignment in addition to access to cloud computing with specialized computing resources (e.g. GPUs) needed for this assignment. All of the assignment notebooks can be accessed through notebooks if students are interested in downloading dependencies to work on their own local machines, these can be followed below and communicated with the course instructors.

PyTorch

PyTorch is a Python library used for developing and training machine learning models, particularly neural networks. It is a powerful library that allows fine-grained control of the training process through APIs that allow the manipulation of a dynamic computational graph and useful utility functions which abstract many parts of the machine learning process such as gradient descent, optimization, and backpropagation. For this assignment, PyTorch and additional relevant machine learning libraries are installed on Google Collab for students already. However, if you are interested in installing PyTorch, it is recommended that you use a solution like Conda or virtualenv when dealing with Python dependencies. The instructions to install PyTorch can be found on the project GitHub.

PyTorch GitHub: https://github.com/pytorch/pytorch

Assignment Prompts.

The following are parts 1-4 of the assignment. Each part contains instructions and guiding questions to complete each part. Finally, you should **submit a report in pdf** showing your solutions. Try to take notes and document each part of your exploration process while completing the assignment. These notes and the responses to the questions should be included in **your report** submitted on GradeScope.

Part 1. CIFAR10 Classification with PyTorch

The first part of this assignment involves introducing yourself to the PyTorch framework and the fundamentals of machine learning. Your goal is to train a neural network on the popular machine learning dataset, CIFAR10, which contains images of ten categories. To save you some time on this assessment, we have implemented a majority of the code for you but have asked you to focus on implementing the **key functions (e.g., training and evaluation)** and finding the **optimal hyperparameters** in order for your model to perform with 80+% accuracy. For this part of the assignment, it is recommended and encouraged that you reference the lab solutions. Detailed instructions can be found in provided notebook!

Notebook: [HERE]

Your Task: Complete **Question 1-3** in notebook Part_1_2_3.ipynb.

Part 2. White-Box Adversarial Example with PGD

The second part of this assignment involves implementing the most representative white-box adversarial exampl algorithm, PGD, to attack your trained CIFAR10 classifier. You are supposed to explain each hyperparameter in PGD attack. Besides, you are supposed to understand the adversarial

training, a commonly used defense to thwart adversarial examples. Detailed instructions can be found in provided notebook!

Reference:

[1] PGD Attack (https://arxiv.org/pdf/1706.06083.pdf)

[2] Tutorial (https://adversarial-ml-tutorial.org/adversarial_examples/)

Notebook: [HERE]

Your Task: Complete **Question 4-8** in notebook Part_1_2_3.ipynb.

Part 3. Simple Implementation of Backdoor Attack

The third part of this assignment involves implementing the most representative backdoor attack method against CIFAR10 classification, called BadNets. You are supposed to build a backdoored classifier that achieves both effectiveness goal and utility goal (explained in the notebook). Besides, you are supposed to understand the differences between adversarial example and backdoor attack. Detailed instructions can be found in provided notebook!

Reference:

[1] BadNets (https://arxiv.org/pdf/1708.06733.pdf)

Notebook: [HERE]

Your Task: Complete **Question 9-13** in notebook Part_1_2_3.ipynb.

Part 4. Quick questions about LLMs

Please answer the following three questions in your report:

Q14: What is a jailbreak attack against LLM (List at least three different purposes)? Why does LLM suffer from jailbreak attacks?

Q15: How can we effectively mitigate the jailbreak risks of LLM (List at least two potential countermeasures)?

Q16: List two approaches to detect LLM generated contents and compare their pros and cons. (Optional) What other methods do you think might be useful for such detection?

Submission

Submission will be handled through Gradescope. You must submit a report **report.pdf** containing your solutions for parts **1-4** of the assignment. For code implementation, you should take a screenshot of your code and the corresponding outputs for grading. You should also submit your completed notebooks (i.e., part_1_2_3.ipynb) to canvas for reference. This will be used for partial credit when grading if necessary.

Grading

This assignment will be graded out of **100 points** and will involve some autograding and some manual grading. Points for each question will be available in the GradeScope. For each question, we focus on your understanding of the problem and your efforts for completion. Therefore, try your best and Good Luck! 😀

If there's any problem, send an email to the TA:
jinghuai 1998@g.ucla.edu.