Homework 2

Security and Privacy of Machine Learning (NWI-IMC069)

Deadline: 02/05/25 23:59

For this homework assignment, you are asked to implement and execute backdoor attacks and defenses. The exercises described below will be graded. However, to complete these exercises, you will need to implement, train, and test your own model using the CIFAR-10 dataset.

In other words, you will need to load the CIFAR-10 training and test sets. You may then either choose an existing (pre-trained) model or design and train one yourself.

You will need to choose a loss function and an optimizer, along with a set of hyperparameters. Then, you must train your neural network on the clean training set to obtain a clean model (or use a pre-trained one). Test your clean model on the CIFAR-10 test set to compute the clean test accuracy. Aim to achieve the accuracy of >70%. You are free to decide on the training settings as long as you remain consistent throughout your work and document everything.

For the exercises below, you will then execute the backdoor attacks and defenses on the CIFAR-10 training and test sets using your model.

You are expected to present your code and explanations within a Jupyter Notebook, which will serve as your report. The notebook should describe what is done, why it is done that way, what the results are, and what they mean. You should include a thorough discussion of the results—not only what is observed, but also why it occurs. We strongly encourage you to include tables and/or figures to present the results.

Please submit notebooks that have been executed and saved with their results, and include descriptive yet concise comments in your code. You must provide clear instructions on how to run the code, as any code lacking instructions or that does not run out of the box will be considered incorrect.

Finally, **copy the (sub)questions you are answering into your notebook for clarity. Please copy each question and write your answer below it.**

1. **BadNet Attack (3 points)**

   (a) *(2 points)* Execute a source-specific BadNet[1] attack on the CIFAR-10 dataset. Create a backdoored dataset using this attack with the following settings:

   - poisoning rate = 8%
   - source label = ship (index 8)
   - target label = cat (index 3)
   - Trigger size = pick one yourself
   - Trigger position = pick one yourself

   Plot a couple of randomly selected images with the trigger. Then train a new model on this poisoned dataset to create a backdoored model. Please follow the same training settings as you used for your clean model. Finally, compute and report the Attack Succes Rate (ASR) and Clean Accuracy Drop (CAD). Save the dataset and model for later use. Evaluate the performance of the attack and share your conclusions.

   (b) *(1 point)* With the source specific attack, the attacker's goal is to let the input be *missclassified* to the target label only when the input has a specific source label. However, we also poison other images. Why do we do this? For every other label (so not source or target), report the percentage of samples in the test set that are now, because of the attack, also missclassified with the target label.

[1] BadNets: Identifying Vulnerabilities in the Machine Learning Model Supply Chain

2. **WaNet Attack (4 points)**

   (a) *(2 points)* Execute a source-agnostic WaNet[2] attack on the CIFAR-10 dataset. Create a backdoored dataset using this attack with the following parameters:

   - k = 8
   - s = 1
   - poisoning rate = 8%
   - target label = cat (index 3)
   - mode = attack. Use the attack mode and not the noise mode as described in the WaNet paper. More specific, use a cross ratio of 0.
   - attack mode = all to one. So one specific target class.
   - grid rescale = 1

   Then train a new model on this poisoned dataset to create a backdoored model. Please follow the same training settings as you used for your clean model. Finally, Compute and report the Attack Succes Rate (ASR) and Clean Accuracy Drop (CAD). Evaluate the performance of the attack and share your conclusions.

   (b) *(1 point)* Apply the WaNet attack using the settings above to generate just one or a few poisoned images. Plot/display them. What can an attacker do to make this attack stealthier? In your answer, explain what the parameters k and s stand for and what they are used for.

   (c) *(1 point)* In the original paper, the authors also mention a third mode, the noise mode. Why did they add this mode for their attack? Explain how this mode improves their attack.

[2] WaNeT - Imperceptible Warping-Based Backdoor Attack

3. STRIP Defense (3 points)

(a) *(2 points)* Implement the STRIP[3] defense to detect backdoored (triggered) inputs at test-time.

Use your BadNet backdoored model from part 1.

You are required to:

- Implement the STRIP defense.
- Create x clean perturbation images and calculate their entropy values.
  - Draw x clean images from the CIFAR-10 test set.
  - For every of the x images, replicate the drawn clean image y times ($y < x$). Creating y identical background images each time.
  - Then also draw new clean images from the test set for every of the y created background images. So you end up with y overlay images.
  - Create y perturbated images by superimposing the background and overlay images.
  - Calculate the entropy for the y images.
  - You should end up with x benign entropy values.
- Create x poisoned perturbation images and calculate their entropy values.
  - Draw x clean images from the CIFAR-10 test set.
  - First, poison these x images using your BadNet attack.
  - For every of the x images, replicate the poisoned image y times ($y < x$). Creating y identical poisoned background images each time.
  - Then also draw new clean images from the test set for every of the y created background images. So you end up with y overlay images.
  - Create y perturbated images by superimposing the background and overlay images.
  - Calculate the entropy for the y images.
  - You should end up with x poisoned entropy values.
- Plot the entropy distributions of clean and poisoned inputs on the same graph for comparison.
- Determine a FRR value you want to work with.
- Using your FRR value, determine a suitable threshold value to distinguish between clean and backdoored inputs.[4]
- Apply your threshold to a sample of mixed poisoned and benign images and classify them as clean or backdoored based on their STRIP entropy values.
- Display a sample of images alongside their STRIP-based predictions (i.e., clean or backdoored).

You may pick your own x and y values for this exercise. Please share your conclusions based on the entropy plot and sample predictions.

[3] STRIP: a defense against Trojan attacks on deep neural networks

[4] For example, you can follow the method described in the STRIP-ViTA paper or use `scipy.stats.ppf()` as explained during the tutorial.

(b) *(1 point)* Why does Strip use entropy of model predictions to detect backdoor triggers? Please include in your answer an explanation on how the entropy is computed, what it means and how it can be used to detect backdoored images.

Note: For all questions, you can use the existing libraries for the attacks. However, you must provide a reference.