Trustworthy Machine Learning, SS 2025
Prof. Dr. Adam Dziedzic and Prof. Dr. Franziska Boenisch
Assignment Sheet #3: Robustness

Teaching Assistant: Jyotika Mahapatra

Deadline: 9th of July, 2025

Task: Training a robust classifier with the highest possible accuracy for clean as well as adversarial data points.

Adversarial examples are inputs perturbed in subtle ways that are imperceptible to human eyes but can mislead machine learning models into making incorrect predictions. The goal of this assignment is to train a classifier which is robust against such adversarial examples that are generated using **Fast Gradient Sign Method (FGSM)** and **Projected Gradient Descent (PGD)**. Thus, your model should make correct predictions for clean data as well as not get misled by adversarial inputs. Your model is expected to perform well and get the highest accuracy for clean data and adversarial examples. Your model will act as the victim model which will be tested against different inputs.

For better understanding of the concepts such as adversarial examples and FGSM and PGD attacks, refer to the following lecture - **Lecture Video**, **Notes**

You are provided with:
Training set for your model: **Training Set**

- Structure of the dataset is identical to the ones provided for the previous tasks.
- To avoid any confusion, the labels of the images are encoded by integers (not by random strings).

Deliverable from your side :
A PyTorch *.pt file containing your model's state dict, as well as the model class name.

***Note**: The samples on which your model will be evaluated come from the same distribution as the provided dataset.*

What's your task in this assignment?

- Your aim is to build a robust classifier model.
- You need to adversarially train your selected class model against adversarial inputs generated using PGD and FGSM.
- There exists a trade-off between robustness and performance for clean data. Take both into consideration while training.

How to submit your robust model for evaluation?

- The evaluation endpoint (for this task, http://34.122.51.94:9090/robustness) expects you to provide a **PyTorch *.pt file** consisting of the **model's state dict**, as well as the **model's class name** (string, in the "model-name" field) and your respective **token number** provided to you earlier via email(same as previous assignments).
- We require you to provide one of the following model classes (from torchvision.models), for us to run evaluations:
  - resnet18
  - resnet34
  - resnet50
- You can refer to **this code** as an example submission.
- **The code** also has the assertions that are run on the side of the evaluation server regarding your submitted file. We encourage you to first run the assertions, and only then submit your model for evaluation.

Things that can go wrong from your side :

- Your model expects incorrect input dimensionality (it has to be 3x32x32, same as before)
- Your model output is of incorrect dimensionality (it has to be 10, one for each class)
- Your file structure is incorrect, this might include:
  - Use of custom model class.
  - Saving the whole model instance, instead of the model dict.
  - Some other kind of mismatch.
- During submission, the wrong model is mentioned in the model_name field or left empty (it has to be one of the listed models mentioned above).

  Thus, it is recommended to run the provided assertions before making a submission.

Evaluating your submission

The evaluation process is as follows:

1. We first load your submitted model, and run assertions on it. The assertions must pass for the evaluation to continue.
2. The clean accuracy for your model is calculated on our private samples.
   Note: We require the clean accuracy to be above 50% for your results to be accepted and then checked against adversarial examples.
3. We run adversarial attacks on the samples and compute accuracies of the perturbed data.

Important Note: Your results will be overwritten with each submission. This is because there exists a trade-off between robustness and performance, and that needs to be measured.

Remember: You're limited to only **one submission per hour**, and the immediate result you get back is evaluated on 30% of the data, and your final model (after the deadline) will be evaluated on 70% of the private data.

Scoreboard

- You can access the scoreboard for this task here http://34.122.51.94:9090/score_board in the Robustness(Clean), Robustness(PGD) and Robustness(FGSM). This will help you to compare your solutions with other teams and see where you stand.
- This scoreboard only shows the results on 30% of the samples from your submission, so it is an intermediate scoreboard.
- The evaluation is split into two sets: immediate and final (30:70). The score you will see before the deadline is for the immediate set; the final (now hidden) one will be revealed once the deadline for the assignment passes.

Read the following instructions carefully

- Create a GitHub repository TML25_A3_YourTeamNumber. For example, if you are in Team 13, your repository should be named TML25_A3_13. ***The team number here should be the one assigned to you via email and not the one you see in CMS.***
- Upload your code files to the repository.
- The documentation for the assignment is split into a **README and a report, and both should be submitted**. The final grade will heavily depend on the documentation.
- Add a README.md to your GitHub repository that points us to the most important files and pieces of code.
- Add a report (PDF format) to your GitHub repository that describes your solution.
- **Submit a zip file (from your GitHub repository) with your readme, report, and the whole repository (only the important files, please avoid including the large models and artifacts in your repository).**
- You are only supposed to make *one submission* per group.