Assignment 3 (robustness)

Train a robust classifier against adversarial examples. Your goal is to achieve the highest accuracy for clean, as well as adversarial examples created using FGSM and PGD.

- You are provided with a training set for your model. Its structure is identical to the ones provided for the previous tasks. This time, to avoid any confusion, the labels of the images are encoded by integers, not by random strings.
- Your submission will be an PyTorch *.pt file containing your model's state dict, as well as the model class name.
- The samples, on which your model will be evaluated come from the same distribution as the provided dataset.

Task artifacts

1. Training set

Submission

The evaluation endpoint (for this task, http://34.71.138.79:9090/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 order for us to be able to run the evaluation, we require you to provide one of the following classes (from torchvision.models):

- resnet18
- resnet34
- resnet50

A couple of things can go wrong on your side:

- Incorrect input dimensionality expected by your model (has to be 3x32x32, as before)
- Incorrect output dimensionality (has to be 10, each one corresponding to one class)
- Incorrect file structure
    - Custom model class
    - Whole model instance, instead of the state dict
    - Some other mismatch is also possible
- Wrong/empty model_name field (has to match the above bullet point list)

We provide you an example submission, under *example_assignment_3.py*, as well as the assertions 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.

Evaluation

The evaluation procedure is as follows:

1. We load your model, and run assertions on it.
2. We calculate clean accuracy on our private samples.
    a. **We require the clean accuracy to be above 50% for your results to be accepted.**

3. We run adversarial attacks on the samples and compute accuracies of the perturbed data.

Very important note: your results will be overwritten with each submission. It's because robustness and clean performance are a trade-off game, and we want to consider that.
Same as in the first Assignment, you're limited to only one submission per hour, and the immediate result you get back is evaluated on 30% of the data, and the final (after the deadline) will be on 70% of the private data.

README

Please, write the README.md in your GitHub repository for this assignment. Describe how you solved the assignment and point to the most important files and pieces of code. The final grade will depend heavily on how you describe your implementation.