Winter 2023 CS291A: Special Topics on Adversarial Machine Learning – Homework 1

Due: **Sunday, Feb 19, 2022, 11:59 pm PST**

Note: Please upload your PDF report and implementations (python files) to Gauchospace before the deadline.

In this homework, you will implement typical attack generation methods to evaluate the adversarial robustness of deep learning models. You will conduct attacks on the CIFAR-10 dataset against both **standard trained model** and **robustly trained model**. The model architecture you need to use is ResNet-18. Specifically, in your report you need to report the following results and your analysis:

1. The accuracy of the provided models (both standard and robust trained models) on CIFAR-10 test set.

2. The accuracy of the provided models under **untargeted FGSM** (l_∞) attack with the following parameters: $\epsilon = 8/255$ (*i.e.,* $\|\delta\|_\infty \leq 8/255$). Please use the untargeted CE loss as the attack loss.

3. The accuracy of the provided models under **untargeted PGD** (l_∞) attack with the following parameters: $\epsilon = 8/255$, CE loss, $\alpha = 2/255$ (attack step size in each PGD iteration), and $T = 10$ (number of PGD interations).

4. The same setting as the third question, but change the CE loss to the **untargeted C&W** loss. Please set the confident threshold τ as zero.

5. The same setting as the third question, but change the CE loss to the **targeted C&W** loss ($\tau = 0$). Please use "class **1**" as the target class (note class index starts at zero) and **only evaluate** on examples that are not class one.

6. Study the influence of different attack settings and plot and report the accuracy changes for both models. Specifically, consider the **untargeted PGD** attack using the CE loss with the following settings:

   (a) Fix $\alpha = 2/255$ and $\epsilon = 8/255$, vary $T \in \{1, 5, 10, 20, 50\}$.

   (b) Fix $\alpha = 2/255$ and $T = 10$, vary $\epsilon = \{1/255, 2/255, 4/255, 6/255, 8/255\}$.

To help with your implementation, we provide a pre-trained standard ResNet-18 model (`resnet_cifar10.pth`) as well as a robust one (`pgd10_eps8.pth`) for you to download. We also provide a data loader for the CIFAR-10 dataset and a checkpoint loader for the two ResNet-18 models. All you need to implement is the attack algorithm and two attack losses. Specifically, your implementations should support changing different hyperparameters for your attacks easily, which includes

1. The number of PGD iterations T.

2. The perturbation budget ϵ.

3. The attack step size α (for PGD only).

4. Choosing between targeted v.s. untargeted C&W loss.

5. Choosing different victim models.

To have a better training dynamic, please use **sign-based** gradient descent for your PGD implementations. In other words, the descent function should be implemented as

$$\delta_{t+1} = \delta_t - \alpha * \text{sign}(\nabla_\delta \ell_{atk}).$$

This is very important to ensure consistent results with existing literature.

Your program should be named as `evaluate.py` and support the command line arguments (as shown below). We also provide a template that specifies the pipeline of the robust evaluation process. You are recommended to use the template and fill in the missing parts. Of course, you are also welcome to write your own one. Note that all evaluations are conducted on the test set.

```
import argparse

parser = argparse.ArgumentParser()
parser.add_argument(
    "--eps", type=int, default=8, help="Attack budget: epsilon / 255"
)
parser.add_argument(
    "--alpha", type=float, default=2, help="PGD attack step size: alpha / 255"
)
parser.add_argument(
    "--attack_step", type=int, default=10, help="Number of PGD iterations"
)
parser.add_argument(
    "--loss_type", type=str, default="ce", choices=["ce","cw"], help="Loss type
                                            for attack"
)
parser.add_argument(
    "--data_dir", default="./data/", type=str, help="Folder to store downloaded
                                            dataset"
)
parser.add_argument(
    "--model_path", default="resnet_cifar10.pth", help="Filepath to the trained
                                            model"
)
parser.add_argument("--targeted", action="store_true")
parser.add_argument("--device", type=str, default="cuda:0", help="Device to use")

args = parser.parse_args()
```

Sanity check:

1. When attacking the standard model using untargeted PGD attack (CE loss) with the following parameters $\epsilon = 8/255, \alpha = 2/255, T = 10$, the robust accuracy should be smaller than 1%.

2. When attacking the robust model using untargeted PGD attack (CE loss) with the following parameters $\epsilon = 8/255, \alpha = 2/255, T = 10$, the robust accuracy should be smaller than 55%.