European Championship in Trustworthy AI
Prof. Dr. Adam Dziedzic and Prof. Dr. Franziska Boenisch
TASK #1: Adversarial Examples

CISPA HELMHOLTZ CENTER FOR INFORMATION SECURITY

Task: Find Adversarial Examples, while applying as little change to the original images as possible.

The goal of Adversarial Example generation is to create altered images that look identical to the original image to the human eye, but are classified differently by an image classifier. In this task, you are in the *attacker's shoes*, and your goal is to create these exact adversarial examples, based on a provided dataset and an image classifier with black-box access.

As an attacker, you get access to:

- An image classifier that returns logits for an input image via an API
- A dataset containing 100 natural images (also available at HuggingFace or on juelich login nodes at p/project1/training2557/common/adversarial-examples)

What you don't have access to:

- The training dataset used to train the classifier
- The architecture and parameters of the classifier

What's your task:

Your goal is to create one adversarial example per natural image from the provided dataset. These examples should be misclassified by the classifier, while being as similar as possible to the original images.

Your Starting Point

To start you off, we give you a coding template that shows you how to load the dataset, and how to query the image classifier with input images, as well as instructions explaining the result submission process.

Querying the Classifier

In order to merely give you black-box access to the image classifier, you do not have direct access to it. However, you can input images and get back the corresponding logits by interacting with the API. Remember to replace PATH/TO/YOUR/QUERY_FILE.npz with the actual images you want to check (keep the double quotes), and set GET_LOGITS to True. The code for this is given in the coding template.

You can query batches of 100 images at a time. **You can only query the classifier for logits once per hour.**

How to submit your results?

- We will provide a unique API-key to every participating team. You are supposed to only use this token to submit your results to our server. If you did not receive this team-specific token, please contact us.

- Your submissions should consist of a .pt file containing all of your adversarial examples, as well as IDs indicating what original images they are based on.

- You are supposed to submit your scores using the code provided in the coding template. Remember to replace YOUR_API_KEY_HERE with your actual API-key (keep the double quotes), and replace PATH/TO/YOUR/SUBMISSION.npz with your real results. Also set SUBMIT to True.

- To avoid the possibility of brute-forcing, only one submission can be made every 5 minutes. If the submission was unsuccessful due to an error, this cooldown period will not apply.

Evaluation:

After submitting your results, they will be evaluated.The evaluation is conducted taking into account the two essential factors for adversarial examples:

- They yield a different result (incorrect label in this case) from the natural version on a given classifier
- They are (visually) as close to the natural version as possible.

The evaluation metric is the average normalized L2-distance between the natural images and your adversarial examples. All submitted examples that yield the same result as the natural version (with the correct label) on the classifier are assigned distance 1. The distance is normalized in the range [0,1]. Your goal is to provide the modified 1000 images so that they are mislabeled and as close as possible (the smallest distance as close to 0) to the initial images.

Leaderboard

After evaluation, your results can be found in the leaderboard:

- You can access the leaderboard for this task at http://34.122.51.94:80/leaderboard_page. This will help you to compare your solutions with other teams and see where you stand.

- The leaderboard shows the best result per team only. As output to your request, you will get back the score for your current submission. If it is lower than the score saved in the leaderboard, the score will not be updated.

References

- "Towards Deep Learning Models Resistant to Adversarial Attacks" Aleksander Mądry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, Adrian Vladu. ICLR 2018. https://openreview.net/pdf?id=rJzIBfZAb (PGD attack)

- “Towards Evaluating the Robustness of Neural Networks” Nicholas Carlini, David Wagner. IEEE Symposium on Security and Privacy (S&P) 2017. https://arxiv.org/pdf/1608.04644 (CW attack)

- “Explaining and Harnessing Adversarial Examples” Ian J. Goodfellow, Jonathon Shlens, Christian Szegedy. ICLR 2015. https://arxiv.org/pdf/1412.6572 (FGSM attack)