NovHack
Technical Description

April 2021

The following document provides all information relative to the technical aspects of the hackathon. It starts by a presentation of the subject as well as a description of the tasks to be completed. Then, the practical information is detailed. More specifically, those are the setup, the infrastructure, the submission, the evaluation and the metric.

Table of Contents

Subject **2**
- Description 2
- Setups 3

Practical Information **3**
- In Practice 3
  - White-Box 3
  - Black-Box 3
- Infrastructure 4
- Submission 4
- Evaluation 4
- Metric 4

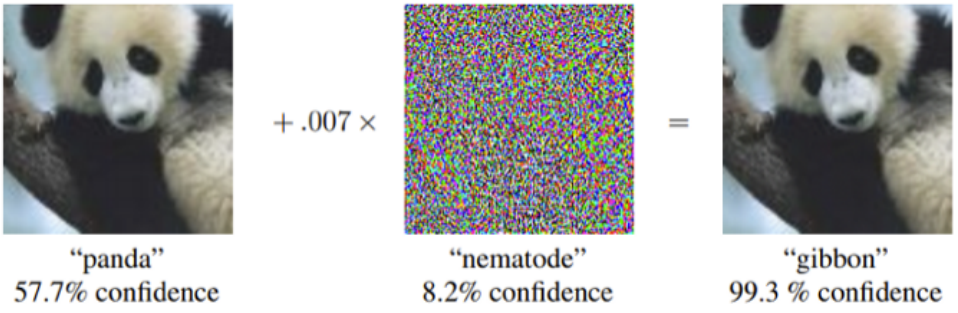
Subject

During this hackathon, you will have to realize two adversarial attacks: the first one in a white-box setup and the second one in a black-box setup. In the following, the concept of adversarial attack is presented before explaining the white-box and the black-box setups.

Description

In classification problems, machine learning models aim at separating classes by producing a function that maps input data to a given class. To do so, the model is trained on a set of data that allows it to learn the corresponding class related to the feature of a given data point. In the context of computer vision and more specifically in image classification, such a trained model will infer the class of a given image, i.e. with a confidence score/the probability that the image belongs to that class.

In image classification, an adversarial attack consists in slightly modifying images that were correctly classified by the model so that they will be misclassified. Indeed, the introduction of a perturbation, that is not visible to the human eye, on an image may induce the model to misclassify it. Such a modified image is called an adversarial example. The figure below illustrates such behavior.



The leftmost image is correctly classified as a panda by the classifier with a 57,7% confidence. A certain amount of noise is added to the original image after being weighted. The result is an image identical to the original one for the human eye. However, the classifier is not able to correctly classify the image anymore since it is confident at 99.3% that the image contains a gibbon.

Mathematically, for a given image i

$$i' = i + min(\varepsilon \mid y \neq y')$$

with:

i' *the modified image*

ε *the noise added to the original image in order to obtain an adversarial example*

y *the original predicted class corresponding the image* i

y' *the predicted class corresponding the image* i'

Setups

There exist two setups of interest in the context of the hackathon: the white-box and the black-box setups.

- In a white-box setup, the attacker (you) has access to the model. Nothing is hidden from him and all information can be retrieved from the model such as the algorithm, its hyper-parameters, its architecture, etc.
- In a black-box setup, the attacker can only query the targeted classification model. In other words, one can only ask for the classification result on a given image. In this hackathon, the black-box model provides a vector of probabilities whose elements are the probabilities to belong to the considered classes. You are allowed to query the model as many times as you wish.

Practical Information

In Practice

In practice, each team will have a Jupyter Server on Google Cloud Platform that is accessible from the web browser at a unique IP address-port provided to the team coach. A password that is common for all teams will be published on the Discord main chat.

Note: You must keep the IP address-port secret and must not share it with members of other teams!

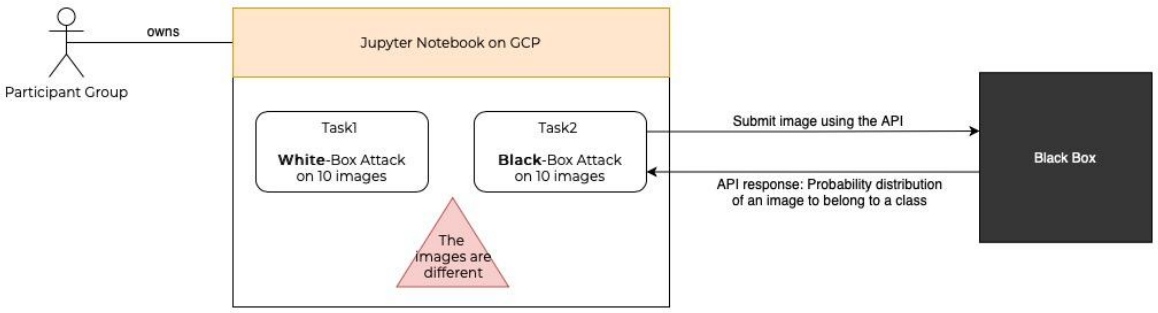
White-Box

In the Jupyter server, there is a notebook that contains an example of how to use the model in inference. The model takes an image as input and it outputs a confidence score for each label as a probability distribution. The image is classified according to the highest confidence score.

The figure below illustrates the process of the attack.

Black-Box

In the Jupyter notebook, the team will find an example of how to query images against the black-box. The response of the query is a vector of probabilities whose elements are the probabilities that the image belongs to the considered classes, as illustrated in the figure below.



Additionally, the figure shows that there are 2 tasks, each having its own test set of images, i.e. 10 images for the white-box attack and 10 images for the black-box attack.

Infrastructure

Each jupyter notebook is hosted on an instance of Google Cloud Platform. Each instance has 3.5 CPUs, 22 GB of RAM and 5 GB of Disk space

Note : Any request made from an external computer to the cluster will be rejected.

Submission

An example of submission can be found in the Jupyter notebook. The team should submit a folder of a specific structure containing the adversarial examples of each task using an API call.

This submission must be made before **5PM** and only the last submission will be taken into account. The scores (described below) will then be automatically computed on a server and the images will be saved so that the jury can access both the scores and the corresponding images.

Evaluation

The evaluation is made in 2 stages. The first one allows a pre-selection based on the weighted sum of the metric values, described below, obtained on the adversarial examples for both tasks.

$$score_{final} = \frac{0.45}{|I_{white-box}|} \sum_{i \in I_{white-box}} score(i) + \frac{0.55}{|I_{black-box}|} \sum_{i \in I_{black-box}} score(i)$$

$with:$

$I_{white-box}$ $the\ test\ set\ of\ \ image\ for\ the\ white-box\ attack$

$I_{black-box}$ $the\ test\ set\ of\ \ image\ for\ the\ black-box\ attack$

Then, the top 10 best teams will be invited to present their work to the jury that is composed of technical and non-technical members. The winners will be selected based on multiple criteria that are listed below with their weighting.

- Comprehension of the subject[1] - 11%
- Quality of the oral presentation[1] - 9%
- Quality of the supports of the presentation (visual)[1] - 7%
- Innovation/Originality[1] - 8%
- Scientific quality of the argumentation during the Q&A[1] - 11%
- Real word usability[1] - 10%
- Score white-box[2] - 10%
- Score black-box[2] - 11%
- Organisation skills[3] - 7%
- Team working[3] - 7%
- Methodology[3] - 9%

Metric

The metric used to evaluate the adversarial examples accounts for diverse aspects. For a given image these aspects are:

- Whether the image is wrongly classified by the target model or not. In the last case, we account for the decrease in the confidence score of the original label.
- The quantity of added noise with respect to original image.

Mathematically, this expressed as follows:

$$score = g(i, i')\left(1 - \frac{\| i - i' \|_2}{\| white\ image \|_2}\right)$$

$with$:

i' *the adversarial example corresponding to the image* i

$$g(i, i') = \begin{cases} 1 \ if \ argmax(F(i')) \neq y \\ F_{\Delta y} \ otherwise \end{cases}$$

F *the target classifier model that outputs a vector of probabilities to belong to each classes*

y *the original predicted class corresponding to the image* i

$$F_{\Delta y} = \begin{cases} the\ y^{th}\ component\ of\ the\ vector\ F(i) - F(i')\ if\ it\ is > 0 \\ 0\ otherwise \end{cases}$$

[1] Filled by the jury members
[2] Filled automatically based on your submission
[3] Filled by your mentor