Adversarial Attack on LeNet5

1. Introduction

Given a pre-trained LeNet5 model, your job is to provide jpeg images to fool the network. There are 2 attack scenarios in this assignment, correspond to 2 questions:

Scenario 1: (50%) Make the network thinks a random image belong to a class with high confidence. Submit a random image, as random as possible (characterized by having high Shannon entropy) that make the network believe it belong to one class (any class of your choice) with high confidence (characterized by having a high predicted max probability).

Scenario 2: (50%) Make the network think an image belongs to one class when it clearly belongs to another class.
For this scenario, submit a modified version of the first image in the test set of the MNIST dataset (see the Trick LeNet notebook on e-class) that make the network think it belong to the 0 (zero) class. The modified image must have a high PSNR score (correspond to imperceptible change with respect to the original version).

2. Submission:

You need to submit **2 jpeg** files: **image1.jpg** and **image2.jpg** correspond to the two scenarios above to e-class.
You need to generate the 2 images yourself by any mean neccesary. The resource folder in this assignment only contain a notebook to do the evaluation, the pretrained model and 2 random jpeg images as an example. There's no skeleton code to generates the images for the questions but you can use the Trick The LeNet notebook on e-class as a place to start. A good hint is just treat the requirements in the two scenarios as part of the optimization objectives to optimize your images, similar to what happens in the Trick The LeNet notebook.
To do the evaluation by yourself, just replace the 2 random jpeg images with your own images by uploading them to google colab.

3. Marking:

Scenario 1 (50%):
- Score depends on the highest probability predicted by the LeNet for your image. Scale from 0.8 to 0.9.
- Your image must satisfy the "randomness" requirement by having a Shannon entropy of at least 6.5

Scenario 2 (50%):
- Score depends on the PSNR score of your image. Scale from 0.2 to 0.3 PSNR.
- Your image must fool the network correctly by making it thinks that the image belong to the 0 class.

You can see your score by running the provided notebook using your images. The score you got from there will be your final score.