Blackbox explainer

Context

In this lab, we are interested in blackbox methods. In particular, we focus on two methods of the state-of-the-art. First, you will be asked to compute LIME [LIME] (provided by pip) and to study the impact of the various parameters of the method. Second, you will be asked to implement another well-known method, RISE [RISE].

[LIME] Ribeiro, M. T., Singh, S., & Guestrin, C. (2016, August). " Why should i trust you?" Explaining the predictions of any classifier. In Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining (pp. 1135-1144)

[RISE] Petsiuk V., Das A. and Saenko K., Rise: Randomized input sampling for explanation of black-box models.arXiv preprint arXiv:1806.07421, 2018

Blackbox explainers

Such explainers do no need to access the inner-working of the model but only its input(s) and output(s). While other blackbox methods exist, we are interested in this lab in Feature Attribution methods producing saliency maps.

In this context, most methods fall in the perturbation based category. Such an method uses perturbations of the initial input sample (here, an image) and the predictions of the model on these perturbed samples to compute the importance of each feature of the input sample. The intuition is that the more relevant a combination of features is, the more the prediction should be perturbed when perturbing these features.

In this lab, we are interested in LIME and RISE, two methods that produce saliency maps using masking as perturbation. As all combinations of features cannot be considered, the two methods propose different strategies to reduce the necessary computation and therefore alleviate this issue.

Work

You can find on moodle all the resources of this lab.

LIME

This exercice is described in `LIME.ipynb` and the requirements to execute the notebook are in `LIME_requirements.txt`.

RISE

RISE, for *Randomized Input Sampling for Explanation of Black-box Models*, is another obfuscation method to compute a saliency map emphasizing the features of interest for a given model prediction. More specifically, RISE allows to identify the features that the model considers as important for the prediction of a given class (not necessarily the predicted one).

The key idea of RISE is to generate a large number of random (binary) masks in low resolution and then to upscale them to the dimension of the input image (the upscaled mask values are therefore in [0, 1]). For each of these masks, a prediction is done by the model which produces a score for the considered class. While many methods consider the masked features as responsible for the perturbed prediction, the authors of RISE make the opposite assumption. In RISE, *unmasked* (or partially masked) features are considered as responsible for the prediction, and, instead of considering how far the perturbed prediction is from the initial prediction, RISE considers that important features are kept if and only if the perturbed prediction score is high.

The main steps of RISE are:

- Compute m masks randomly with probability p in low resolution and upscale them
- Compute m perturbed images using an element-wise product with each mask
- For each perturbed image, compute a *local* explanation by using a product between the score of the prediction and the mask
- Aggregate all *local* explanations into a single saliency map.

You should submit an archive with your work on Moodle. This archive should contain your python code and a short report.