Assignment 4 on Explainability

Provide the explanations of predictions in neural networks. You will use the main three methods described in the lecture on explainability, namely: Network Dissection, LIME, and Grad-CAM. Each of them comes with a short deliverable specified below.

Task Artifacts

Datasets

ImageNet

You can download it from many sources. Check out this helpful thread:
https://stackoverflow.com/questions/65685437/access-to-imagenet-data-download

You can also use the examples of images from ImageNet from this repository:
https://github.com/EliSchwartz/imagenet-sample-images

For LIME and Grad-CAM, use the following images from the ImageNet dataset:

1. https://github.com/EliSchwartz/imagenet-sample-images/blob/master/n02098286_West_Highland_white_terrier.JPEG
2. https://github.com/EliSchwartz/imagenet-sample-images/blob/master/n02018207_American_coot.JPEG
3. https://github.com/EliSchwartz/imagenet-sample-images/blob/master/n04037443_racer.JPEG
4. https://github.com/EliSchwartz/imagenet-sample-images/blob/master/n02007558_flamingo.JPEG
5. https://github.com/EliSchwartz/imagenet-sample-images/blob/master/n01608432_kite.JPEG
6. https://github.com/EliSchwartz/imagenet-sample-images/blob/master/n01443537_goldfish.JPEG
7. https://github.com/EliSchwartz/imagenet-sample-images/blob/master/n01491361_tiger_shark.JPEG
8. https://github.com/EliSchwartz/imagenet-sample-images/blob/master/n01616318_vulture.JPEG
9. https://github.com/EliSchwartz/imagenet-sample-images/blob/master/n01677366_common_iguana.JPEG
10. https://github.com/EliSchwartz/imagenet-sample-images/blob/master/n07747607_orange.JPEG

You can see the images below as well.

Models

The first two models (ResNet18 trained on ImageNet and places 365) are for Network Dissect and ResNet50 is used for LIME and Grad-CAM.

ResNet18 trained on places 365

wget --progress=bar http://places2.csail.mit.edu/models_places365/resnet18_places365.pth.tar

ResNet18 trained on ImagNet

from torchvision.models import resnet18, ResNet18_Weights.IMAGENET1K_V1
 # New weights with accuracy 80.858%
resnet18(weights=ResNet18_Weights.IMAGENET1K_V1)

ResNet50 trained on ImageNet:

This is the standard model provided by PyTorch:

from torchvision.models import resnet50, ResNet50_Weights
 # New weights with accuracy 80.858%
resnet50(weights=ResNet50_Weights.IMAGENET1K_V2)

Main tasks

Task 1: Network Dissect

Task: Network Dissect identifies which neurons are responsible to learn a specific class in a neural network. Analyze the internals of models with Networks Dissect. Label all neurons from the last 3 layers of ResNet18 trained on ImageNet and ResNet18 trained on places 365. Analyze the labeled neurons.

For this task, use the latest library. We recommend leveraging the code from: https://github.com/Trustworthy-ML-Lab/CLIP-dissect Specifically, you can run the following script: https://github.com/Trustworthy-ML-Lab/CLIP-dissect/blob/main/describe_neurons.py Analyze which concepts are learned by most neurons. Compare the concepts learned by ResNet18 trained on ImageNet vs ResNet18 trained on places 365. How many different objects are learned by the two models? What additional analyses can you run and what additional findings can you make?

Deliverable 1: A short report (recommended 1 page) with some visualizations of your findings, e.g. histograms of the counts of neurons that learned different concepts, images that correspond to the neurons, etc. Please describe your findings well.

Task 2: LIME

Task: LIME provides Local Interpretable Model-agnostic Explanations. The main intuition behind LIME is that it learns an interpretable model locally around the prediction. Visualize, for the 10 given ImageNet data points which parts them are responsible for the main prediction using LIME.

Please, follow the tutorial on LIME from:
https://github.com/marcotcr/lime/blob/master/doc/notebooks/Tutorial%20-%20images%20-%20Pytorch.ipynb

Deliverable 2: For the 10 given ImageNet images, obtain and plot the annotations by LIME and analyze your results in a short report.

Task 3: Grad-CAM

Task: For the 10 given ImageNet data points, visualize which parts of the input image are responsible for the main prediction using Grad-CAM. Compute the gradient of the output with respect to the *last convolutional layer*.

Use the following library for Grad-CAM:
https://github.com/jacobgil/pytorch-grad-cam?tab=readme-ov-file#using-from-code-as-a-library
Compare the results with the ones obtained using LIME. Extend the analysis using other types of Grad-CAM, namely: AblationCAM and ScoreCAM.

Deliverable 3: For the 10 given ImageNet images, obtain and plot the annotations by Grad-Cam and analyze your results in a short report.

Task 4: Compare results from LIME and Grad-CAM:

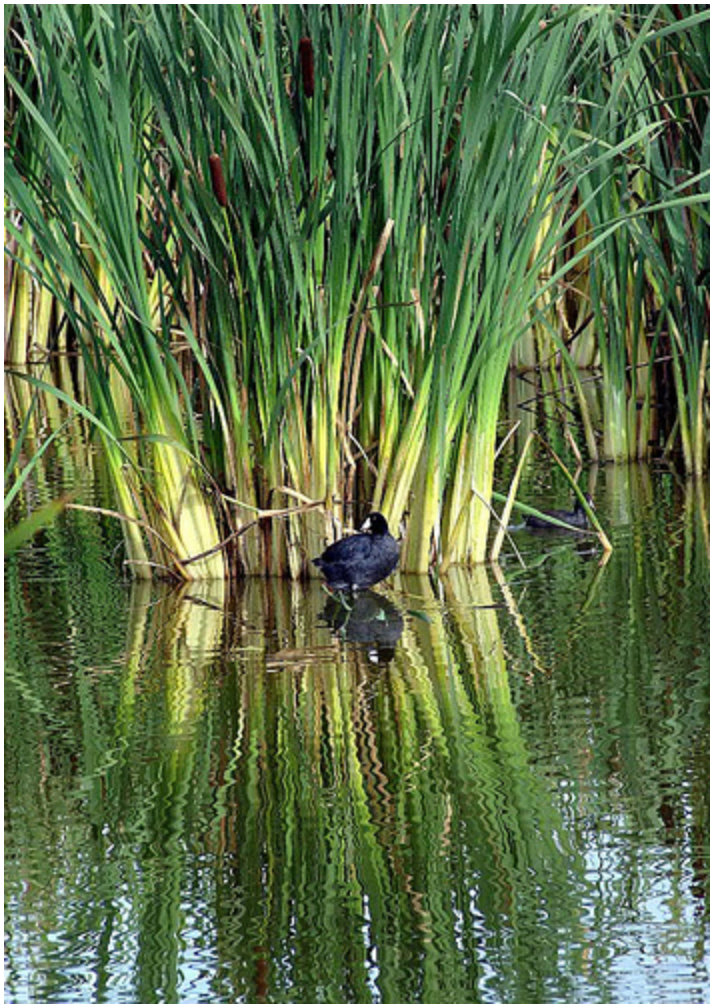
Based on taks 2 and task 3, identify the differences in the results between LIME and Grad-CAM. Compare the highlighted regions, for example, using Intersection over Union (IoU) metric. Example analyses could include questions such as: is it the case that for simpler images, like goldfish, the IoU is higher than for more complex images such as kite, which would indicate that both methods agree more.

Deliverable 4: A short report on the differences you observed and the insights you gained on the differences in the methods (recommended length, including any visualizations, if applicable, 1 page).

README

Please, write the README.md in your GitHub repository for this assignment. Describe how you solved the assignment and point to the most important files and pieces of code. The final grade will depend heavily on how you describe your implementation.

Images




epicphoto.net
DYAL & ASSOCIATES
MOTORSPORT LOGIC