Lazy Explainer Report

General information

Model name: *DenseNet121*
Dataset name: *Imagenette*
Execution time: *76.8 s*
Package version: *0.0.1*
Date: *2022-12-04*
Selected method: *GradCam*

Table of results

Explanation Name	Rank	Faithfulness Est. ↑	Avg Sensitivity ↓	IROF ↑	Sparseness ↑	Time elapsed [s]	Agg. Score
GradCam	1	0.539	0.032	1.011	0.581	0.1	10
Integrated Gradients	2	0.07	0.024	1.011	0.674	0.2	8
Saliency	3	0.477	0.046	1.01	0.572	0.1	4
KernelSHAP	4	-0.024	0.263	1.011	0.37	3.2	2

Table description
Arrow next to the metric names indicates whether larger or smaller values of metric are better. Time elapsed shows time that was required for computation of attribution for given batch of images. When there is a tie in Aggregated Score, the best metric is chosen based on computation time.

Details

Explanations:

- **KernelSHAP**: More information regarding this method and proof of equivalence can be found in the original paper here.
  Explanation's parameters:
  { 'explanation_parameters': { 'baseline_function': baseline_color_mean,
  'baseline_function_name': 'mean',
  'n_samples': 15},
  'mask_parameters': {'n_segments': 50}}

- **Integrated Gradients**: Integrated Gradients is an axiomatic model interpretability algorithm that assigns an importance score to each input feature by approximating the integral of gradients of the model's output with respect to the inputs along the path (straight line) from given baselines / references to inputs. More information regarding method can be found here.
  Explanation's parameters:
  { 'explanation_parameters': { 'baseline_function': baseline_color_mean,
  'baseline_function_name': 'mean',
  'n_steps': 10}}

- **GradCam**: Computes element-wise product of guided backpropagation attributions with upsampled (non-negative) GradCAM attributions. GradCAM attributions are computed with respect to the layer provided in the constructor, and attributions are upsampled to match the input size. GradCAM is designed for convolutional neural networks, and is usually applied to the last convolutional layer. More information regarding method can be found here.
  Explanation's parameters:
  { 'explanation_parameters': { 'selected_layer': 'features.denseblock4.denselayer16.conv2'}}

- **Saliency**: Returns the gradients with respect to inputs. More information regarding method can be found here.
  Explanation's parameters:
  {'explanation_parameters': {'abs': True}}

Metrics:

- **Faithfulness Estimate**: Computes the correlation between probability drops and attribution scores on various points.(Alvarez-Melis et al., 2018)
  Metric’s parameters:
  { ’call’: {’device’: ’cuda’},
  ’init’: { ’disable_warnings’: True,
  ’display_progressbar’: False,
  ’features_in_step’: 256,
  ’normalise’: True,
  ’perturb_baseline’: ’mean’,
  ’softmax’: True}}

- **Average Sensitivity**: Measures the average sensitivity of an explanation using a Monte Carlo sampling-based approximation.(Yeh et al., 2019)
  Metric’s parameters:
  { ’call’: {’device’: ’cuda’},
  ’init’: { ’disable_warnings’: True,
  ’display_progressbar’: False,
  ’lower_bound’: 0.2,
  ’norm_denominator’: fro_norm,
  ’norm_numerator’: fro_norm,
  ’normalise’: True,
  ’nr_samples’: 15,
  ’perturb_func’: uniform_noise,
  ’perturb_radius’: 0.2,
  ’similarity_func’: difference}}

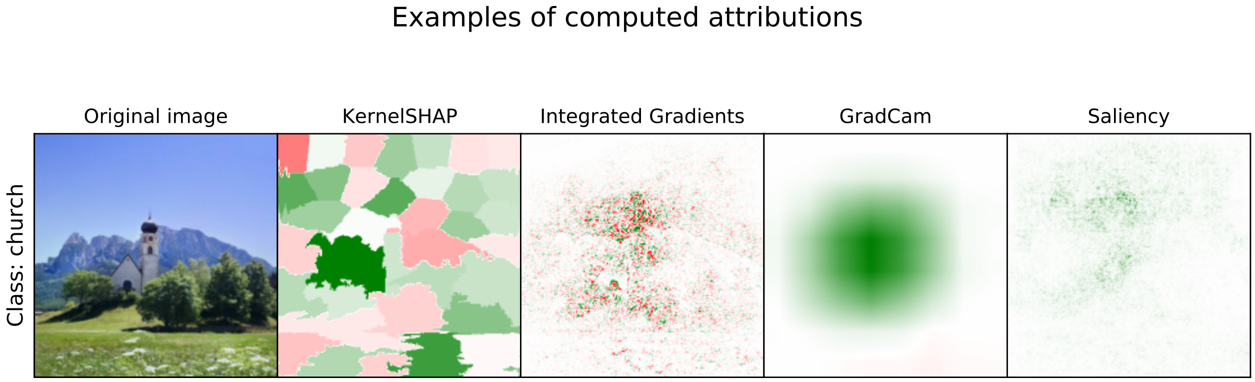
- **Iterative Removal of Features**: Computes the area over the curve per class for sorted mean importances of feature segments (superpixels) as they are iteratively removed (and prediction scores are collected), averaged over several test samples.(Rieger at el., 2020)
  Metric’s parameters:
  { ’call’: {’device’: ’cuda’},
  ’init’: { ’disable_warnings’: True,
  ’display_progressbar’: False,
  ’perturb_baseline’: ’mean’,
  ’return_aggregate’: False,
  ’segmentation_method’: ’slic’,
  ’softmax’: True}}

- **Sparseness**: Uses the Gini Index for measuring, if only highly attributed features are truly predictive of the model output.(Chalasani et al., 2020)
  Metric’s parameters:
  { ’call’: {’device’: ’cuda’},
  ’init’: {’disable_warnings’: True, ’display_progressbar’: False}}

Aggregation parameters

{ ’first_stage_aggregation_function’: ’mean’,
’second_stage_aggregation_function’: ’rank_based’,
’second_stage_aggregation_function_aggregation_parameters’: {}}

Examples of explanations


Examples of computed attributions
Original image
KernelSHAP
Integrated Gradients
GradCam
Saliency
Class: church