AutoeXplainer Report

General information

Model name: *DenseNet121*
Dataset name: *Imagenette*
Execution time: *1992.959 s*
Package version: *0.0.3*
Date: *2023-01-26*
Selected method: *GradCam*
Number of images: *2*

Model performance

Accuracy: *1.0*
F1 macro: *1.0*
Balanced accuracy: *1.0*

Table of results

Explanation Name	Rank	Faithfulness Est. ↑	Avg Sensitivity ↓	IROF ↑	Sparseness ↑	Time elapsed [s]	Agg. Score
GradCam	1	0.500	0.014	45.814	0.560	0.652	10
Saliency	2	0.390	0.055	48.937	0.556	1.755	7
Integrated Gradients	3	-0.142	0.032	21.450	0.681	27.829	5
KernelSHAP	4	0.228	0.318	30.064	0.414	23.955	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**: Uses the LIME framework to approximate Shapley values from game theory.(Lundberg and Su-In Lee, 2017)
  Explanation's parameters:
  ```
  { 'explanation_parameters': { 'baseline_function': baseline_color_black,
  'baseline_function_name': 'black',
  'n_samples': 50},
  'mask_parameters': {'n_segments': 50}}
  ```

- **Integrated Gradients**: Approximates feature importances by computing gradients for model outputs for images from the straight line between the original image and the baseline black image. Later, for each feature, the integral is approximated using these gradients.(Sundararajan et al., 2017)
  Explanation's parameters:
  ```
  { 'explanation_parameters': { 'baseline_function': baseline_color_black,
  'baseline_function_name': 'black',
  'n_steps': 20}}
  ```

- **GradCam**: For the selected layer and a target class, it computes gradients, multiplies its average by layer activations and returns only the positive part of the result. For images with more than one channel, it returns the positive part of the sum of results from all channels.(Selvaraju et al., 2016)
  Explanation's parameters:
  ```
  { 'explanation_parameters': { 'relu_attributions': True,
  'selected_layer': 'features.denseblock4.denselayer16.conv2'}}
  ```

- **Saliency**: Is based on computing gradients. The idea is to approximate CNN's output for a given class in the neighborhood of the image using a linear approximation and interpret the coefficients vector as an importance vector for all pixels.(Simonyan et al., 2013)
  Explanation's parameters:
  ```
  {'explanation_parameters': {'abs': True}}
  ```

Metrics:

- **Faithfulness Estimate**: Evaluates the relevance of the computed explanation by calculating the correlation between computed feature attribution and probability drops after removing features.(Alvarez-Melis et al., 2018)
  Metric’s parameters:
  ```
  { ’call’: {},
  ’init’: { ’disable_warnings’: True,
  ’display_progressbar’: False,
  ’features_in_step’: 256,
  ’normalise’: True,
  ’perturb_baseline’: ’black’,
  ’softmax’: True}}
  ```

- **Average Sensitivity**: A metric that measures an average of how sensitive to perturbations the explanation method is. The implementation uses a Monte Carlo sampling-based approximation.(Yeh et al., 2019)
  Metric’s parameters:
  ```
  { ’call’: {},
  ’init’: { ’disable_warnings’: True,
  ’display_progressbar’: False,
  ’lower_bound’: 0.2,
  ’norm_denominator’: fro_norm,
  ’norm_numerator’: fro_norm,
  ’normalise’: True,
  ’nr_samples’: 20,
  ’perturb_func’: uniform_noise,
  ’perturb_radius’: 0.2,
  ’similarity_func’: difference}}
  ```

- **Iterative Removal of Features**: Iteratively removes the most important features and measures the change in probability in the model prediction for a given class. It plots the probability for a given class with respect to the number of removed features and computes the area over the curve.(Rieger at el., 2020)
  Metric’s parameters:
  ```
  { ’call’: {},
  ’init’: { ’disable_warnings’: True,
  ’display_progressbar’: False,
  ’perturb_baseline’: ’mean’,
  ’return_aggregate’: False,
  ’segmentation_method’: ’slic’,
  ’softmax’: True}}
  ```

- **Sparseness**: With the use of the Gini Index measures how imbalanced feature importances given by the explanation method are.(Chalasani et al., 2020)
  Metric’s parameters:
  ```
  {’call’: {}, ’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