Appendix

Appendix I: Proofs

Nonlinearity \[ ? \] concludes that SM, IG, LRP, DeepLIFT are equivalent for linear models and their proof also applies to SG. We first introduce the following proposition:

**Proposition 1.** All attribution methods mentioned in Sec \[ ?? \] except GradCAM and Guided Backpropagation are equivalent if the model behaves linearly.

**Proof.** As the Proposition 4 and Conclusion 6 in \[ ?? \] prove that Saliency Map, Integrated Gradient, DeepLIFT and LRP are equivalent for a linear model, we just need to prove SmoothGrad is equivalent to Saliency Map if the model is linear.

If a model behaves linearly, we can express the output score \( y_c \) for class \( c \) as a linear combination such that \( y_c = w_c^T x + b_c \). Then the SmoothGrad \( S(x)_c \) is

\[
S(x)_c = \mathbb{E}_{x \sim \mathcal{N}(0,1)} \frac{\partial [w_c^T (x + \epsilon) + b_c]}{\partial x} = \mathbb{E}_{x \sim \mathcal{N}(0,1)} w_c = w_c \frac{\partial y_c}{\partial x}
\]

(Saliency Map)

\[ \square \]

**Proof to Proposition 4** If an attribution method \( A \) satisfies both sensitivity-\( n_1 \) and sensitivity-\( n_2 \), then \( N^k_p(x, A) = 0 \) under the condition if \( \sum_{i=1}^{n_1} s_i = \sum_{j=1}^{n_2} s_j = kS(x, A), s_i \in \pi^+_A(x), s_j \in \tilde{\pi}^+_A(x), k \in [0, 1], \) but not vice versa.

**Proof.** If \( A \) satisfies sensitivity-\( n_1 \), for any given ordered subset \( \pi \), we have

\[
\sum_{i=1}^{n_1} s_i = R(x, \pi)
\]

Same thing happens to \( n_2 \) if \( A \) satisfies sensitivity-\( n_2 \). Under the condition if \( \sum_{i=1}^{n_1} s_i = \sum_{j=1}^{n_2} s_j = kS(x, A), s_i \in \pi^+_A(x), s_j \in \tilde{\pi}^+_A(x), k \in [0, 1], \)

\[
N^k_p(x, A) = |R(x, \pi_A(x)) - R(x, \tilde{\pi}^+_A(x))| = |\sum_{i=1}^{n_2} s_i - \sum_{j=1}^{n_2} s_j| = |kS(x, A) - kS(x, A)| = 0
\]

\[ \square \]

Appendix II: Implementation Details

**Models**

We evaluate N_Ord, S_Ord for all attribution methods mentioned in Section \[ ?? \]. We evaluate on 9600 images from ImageNet \[ ?? \] with pre-trained on VGG16\[ ?? \].

**Attribution Methods**

**Saliency Map**

As discussed in Sec \[ ?? \], we use \( \text{grad} \times \text{input} \) to represent the Saliency Map, instead of the vanilla gradient.

**Integrated Gradient**

We use the black image as the baseline for all images and we use the 50 samples on the linear path from the baseline to the input.

**Smooth Gradient**

As discussed in Sec \[ ?? \], we use \( \text{smooth_grad} \times \text{input} \) to represent the Smooth Gradient. We pick a noise level of 20 % as it appears to be the best parameter in its original paper \[ ?? \]. We randomly sample 50 points from the Gaussian distribution for the aggregation.

**DeepLIFT**

We use the black image as the baseline for all images and we use the RevealCancel rule for DeepLIFT \[ ?? \].

**LRP**

We use the implementation of LRP-\( \alpha 2/31 \) with generalization tricks mentioned by \[ ?? \] who argues this rule is better for image explanations.

**Guided Backpropagation**

To implement Guided Backpropagation, we modify the ReLU activation in the network to filter out the negative gradient in tensorflow.

```python
@ops.RegisterGradient("GuidedBackProp")
def _GuidedBackProp(op, grad):
dtype = op.inputs[0].dtype
return grad * tf.cast(grad > 0., dtype) * 
          tf.cast(op.inputs[0] > 0., dtype)
```

**GradCAM**

We use the last convolutional layer to compute the GradCAM for all images.

Appendix III

More examples of evaluating each images with TPN and TPS are shown in Fig \[ 1 \]

\[ ^1 \text{We use the release code on https://github.com/kundajelab/deeplift} \]
Figure 1. More visualizations of different attribution methods. Red checks mark the winner of Total Proportionality for Necessity and blue checks mark the winner of Total Proportionality for Sufficiency.