xAI with Layer-wise Relevance Propagation

Anna Saranti & Andreas Holzinger

human-centered.ai

11.05.2021
Outline

Introduction
LRP vs. Sensitivity Analysis (SA)
Whole dataset analysis with LRP
LRP on LSTMs and Perturbation Analysis
LRP for Pruning
LRP for NNs
LRP task
Benefits of LRP
Literature
Questions
Heatmaps

- Binary classification task
- Cancer or healthy?

What is a good heatmap?

Sensitivity of a pixel $p$ is the norm over all partial derivatives:

$$h_p = \| \frac{\partial}{\partial x_p} f(x) \|$$

How much a small change in the pixel $p$ affects the prediction (output) of the NN.

The direction of change is lost because of the norm.

Needs (locally) differentiable neurons.
LRP vs. SA (2/4)

- Left: Local gradient at prediction point
- Right: Taylor approximation w.r.t. root point on decision boundary

Blue color denotes negative relevance Evidence **against** the predicted class

Whole dataset analysis (1/2)

- PASCAL VOC2007 data set: horse images have a tag
- Classification by high-performing NN
- Use LRP and detect Clever Hans predictions

Whole dataset analysis (2/2)

- Semi-automated Spectral Relevance Analysis
- Improve the model and the dataset

How does word deleting affect performance?

Left: Correct classification, decreasing relevance

Right: Misclassification, increasing relevance

LRP for Pruning CNNs

- Compress the model but keep performance
- Fight overparameterization: More parameters than training samples
- Use xAI to find out most relevant units (weights, filters) automatically

Fully Connected Neural Network

- Input image $x$ processed by neural network (NN) f.e. for a classification task
- The neural network computes the function $f(x)$
- Function $f(x) = 0$: No object in image
  $f(x) > 0$: Object in image with a degree of certainty
Result of LRP when applied in a Convolutional Neural Network (CNN)

- Decompose the decision of the NN into the contributions of individual pixels
  \[ x = \{ x_p \} \]
- To what extent the pixel \( p \) contributes to explaining the classification decision \( f(x) \)
- Heatmaps for correctly classified and misclassified

NN training procedure

- Feedforward, although the network is trained by backpropagation of the error of its prediction with the training data
- LRP is applied after the end of the training procedure
- Training must have good performance; this will influence the quality of the explanations
- The computations of LRP use one backward pass in an already trained NN.
- For the GNN case, multiple backpropagation passes are needed - not to be confused with NN training with backpropagation.
Computational flow of Deep Taylor Decomposition (1/2)

Computational flow of Deep Taylor Decomposition (2/2)

Taylor Decomposition

- Taylor expansion of a function $f(x)$ at point $a$:
  \[ f(x) = f(a) + \frac{f'(a)}{1!}(x - a) + \frac{f''(a)}{2!}(x - a)^2 + \frac{f'''(a)}{3!}(x - a)^3 + \cdots \]

- $\sum_j R_j = \left( \frac{\partial (\sum_j R_j)}{\partial \{x_i\}} \right) \bigg|_{\partial \{\tilde{x}_i\}}^T (\{x_i\} - \{\tilde{x}_i\}) + \epsilon = $

- $\sum_i \sum_j \left. \frac{\partial R_j}{\partial x_i} \right|_{\partial \{\tilde{x}_i\}} (x_i - \tilde{x}_i) + \epsilon$
Pixel-wise decomposition of a function

- Goal: redistribute the neural network output onto the input variables; the relevance $R_j$ to lower-level relevances $\{R_i\}$

How to find the neighbouring point \( \tilde{x} \) ?

- Find a neighbouring point \( \tilde{x} \), for which \( f(\tilde{x}) = 0 \) (root point)
- A good root point is the one that removes elements of the data point \( x \) that cause the \( f(x) \) to be positive (object detected)
- Similar image, object not recognizable from the classifier - hence the output \( f(\tilde{x}) = 0 \)
Properties

- Conservation: \[ \sum_i R_i = \sum_j R_j \]  
  \( (i \text{ and } j \text{ are layers}) \)

- Rectified Linear Unit (ReLU):
Example (1/3)


\[ x_j = \max(0, \sum_i x_i w_{ij} + b_j) \] (ReLU nonlinearity)

\[ x_k = \sum_j x_j \] (Sum pooling)
Example (2/3)

$R_k$ of output layer: Total relevance that must be backpropagated:

$\triangleright R_k = x_k = \sum_j x_j$

$R_j$ of hidden layer: Taylor decomposition on $\{\tilde{x}_j\} = 0$:

$\triangleright R_j = R_k(\tilde{x}) + \left. \frac{\partial R_k}{\partial x_j} \right|_{\{\tilde{x}_j\}} \cdot (x_j - \tilde{x}_j) = x_j = \max(0, \sum_i x_i w_{ij} + b_j)$

$\triangleright$ For which $\tilde{x}$ is $R_k(\tilde{x}) = 0$?

Since ReLU ensures that $\{\forall j : \tilde{x}_j \geq 0\}$ and

$\frac{\partial R_k}{\partial x_j} = \frac{\partial \sum_j x_j}{\partial x_j} = 1$
Example (3/3)

$R_i$ of input layer:

$\quad R_i = \sum_j \left. \frac{\partial R_i}{\partial x_i} \right|_{\tilde{x}_i(j)} \cdot (x_i - \tilde{x}_i(j))$

$\quad R_i = \sum_j \frac{w_{ij}^2}{\sum_{i'} w_{i'j}^2} R_j$ : Relevance weighted proportionally

Use the equations above to compute numerically the relevance of all layers of the network depicted in the figure.

Use your weight values ($w_{ij}$) but think on weighting schemes that are typically used in neural networks. See https://keras.io/initializers/

Verify that the conservation and positivity rules properties apply.

Provide descriptions of the interpretations

Code: https://github.com/albermax/innvestigate
Benefits of LRP

1. More interpretable heatmaps
   Positive and negative relevance
2. Discover artefacts in big datasets
   Actionable insights
3. Principles applied to new NN architectures (GNN)
   Supports Quality Management (QM)
Main LRP paper:

Differences with Sensitivity Analysis (SA):


Whole dataset analysis:


LRP on LSTMs and Perturbation Analysis:


LRP for Pruning:

Questions?

Dipl. -Ing. Anna Saranti
anna.saranti@medunigraz.at