Layerwise Relevance Propagation - Implications for Feature Extraction

Carwyn Collinsworth
M.Sc. Computer Science
Stony Brook University
crcollinswor@cs.stonybrook.edu

Abstract

Layer-wise Relevance Propagation (LRP) is a technique to analyze a classifiers decision making. It is often categorized as a method of explainable Artificial Intelligence (AI). LRP evaluates a classification decision, and is able to quantify each inputs' and intermediate representations' "importance" to said decision. This paper makes two main strides in this area. First, we test the applicability of LRP results by removing image areas quantified as important, and image areas quantified as unimportant and compare via subsequent classification outputs. We found that LRP is effective at identifying important features. Second, we use the results of LRP over a sampled dataset and remove inputs deemed unimportant. The ability to classify remained mostly unchanged; we successfully applied LRP to minimize the computational requirements of the network and optimize effeiciency.

1 Introduction

Neural Networks (NN) have provided machine learning a classifier with unique benefits and detriments. They are simultaneously scalable and tunable function optimizers capable of seemingly endless applicability, and black boxes which provide no introspection about their decisions. However, recent methods such as Layerwise Relevance Propagation (LRP) have been making strides in bringing explainability to NNs. LRP is employed after training a model, at which point it finds out the input features the network learned. This is done by taking the network prediction, and propagating it back through the network - accounting for previous activations and weights - to the inputs. Relevance values are obtained for the input nodes, which show the inputs that are most relevant to the predictions.

2 Background

Introduced to the field of Explainable AI (XAI) in 2015 [1], LRP was introduced to provide insight into how machine learning models actually use inputs to classify. It closes the gap between classification and interpretability both for multilayered NNs and Bag of Words over nonlinear kernels. The original paper derived solutions for these two particular classification architectures and evaluated them for meaningfulness using pixel-wise decomposition - a concept used to judge the contribution of each pixel to the output in the image classification task.

2.1 Origin

Conceptually, LRP assumes that each node can be assigned a "relevance" value, representing how much this node contributes to a certain classification. Directly from the paper, "Rd < 0 contributes evidence against the presence of a structure which is to be classified while Rd > 0 contributes evidence for its presence", where Rd is the relevance value of node d. Assuming conservation of the sum of relevances in each layer, the authors are able to compute relevances for each layer from the output to the input nodes. This relevance assignment technique worked so well, it was attempted to be patented in early 2018 [4], however, its status is still pending. This paper defined an LRP rule - the Basic Rule (Coined LRP-0 by [2]). This rule is the simplest rule, and therefore easiest to understand. LRP-0 works by providing a function to compute the relevance of a previous node given the activations and nodes following it. In layman's terms, the function computes a node's relevance by weighting (by the connection weights)

the normalized (divided by all inputs to the next node) activations of the nodes of the next layer. This was the main finding of the paper, but the authors were far from finished.

2.2 Recent Advancements

Müller et al. [2] continued researching and provided an overview of all the rules they came up with in 2019. Complete with LRP-0, LRP-ϵ (LRP-0 with a regularization term in the denominator), LRP-γ (a rule that assigns more bias to positive effects than negative effects on relevance scores), LRP rules to deal with the non-smoothness of ReLU, and more, this paper served as a summary of findings of the entire project. They even mention an efficient implementation of their relevance score computation algorithm, and LRP for unsupervised models. Even though this paper wasn't able to provide complete insights and detailed analysis of LRP, it does provide a high-level overview with many possible variations for further research.

3 Experimental Setup

The experimental setup is defined by two main components.

- Dataset
- The classification method, in which a variety of network structures are used to predict on the MNIST dataset
- Testing LRP upon important vs unimportant feature removal

3.1 Dataset

To ensure stable results, a pre-cleaned high quality and accredited dataset was chosen. The MNIST dataset was used, and implemented in Python via PyTorch - specifically Torchvision's dataloader.

3.2 Classifier Setup

The first classifier setup was a simple two-layer fully connected neural network. An activation layer with various command-line-defined activation functions - such as ReLU, Sigmoid, and Tanh - was constructed and built between the two linear network layers. The hidden size of the network defaulted to 500, but the option to manually set the dimension is given via command line arguments.

Other networks may have been constructed by other contributors.

3.3 Feature Removal Setup

The purpose of feature removal is to test the implications of LRP. That is, we wish to ensure that highly "relevant" features are most if not solely important to image classification. To test this, we set up a system of feature removal. First, an image representation of a digit is classified. Using that classification, and as LRP specifies, we compute and assign relevance scores to each of the input features (pixels). Then, from this image, we perform two separate transformations. First, we obtain an important-feature image by removing 15% of unimportant (lowest relevance scoring) features. Similarly, we obtain an unimportant-feature image by removing 15% of the most important features from the original LRP image. Finally, these images are classified, and confidence values are computed for the prediction. This confidence, as well as the correctness of the prediction, are noted. Both of these metrics are used to draw conclusions about the interpretability of LRP, as well as how LRP can be used to lower data size and improve network efficiency.

4 Training

By default, the network is trained for 10 epochs with a batch size of 100 and a learning rate of 0.001. The Adam optimizer is used. The loss over training iterations of the two-layer sigmoid network is shown in figure 1.

Figure 1

After training, the accuracy of the model is quite strong, consistently scoring above 95% ac-

curacy. To demonstrate this, an image sample was taken, and incorrectly classified images were highlighted. This is displayed in figure 2.

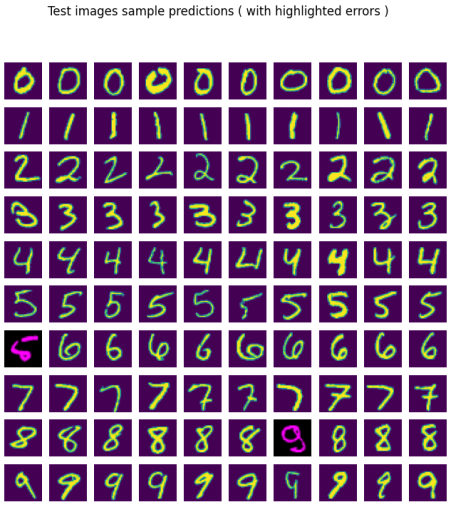

Figure 2: Incorrectly classified samples are highlighted.

Many options are offered in the command line arguments for user-input. Keep this in mind if running the code.

5 Feature Removal Results

After classifying input samples and obtaining LRP results, the most important and least important features (pixels) obtained from the LRP results are removed from the original input image. A sample of the resulting images are displayed in figure 3.

These images are then classified using the same classifier with no extra training. The classification results are saved, and then LRP is run again simply for the visualization aspect. Another sample of these images is shown in figure 4.

Table 1: LRP-Gamma Confidence Before and After Unimportant and Important Feature Removal
IFRC : Important Feature Removal Confidence
UFRC : Unimportant Feature Removal Confidence

Digit	Prediction Confidence	IFRC	UFRC
0	1.16566463756	0.45139701585	0.86510152969
1	1.20136409854	1.2415735656	0.72104469882
2	0.56574845815	0.2009244325	0.45478654217
4	0.82067896009	0.35773196138	0.51314271285
7	0.72628648672	0.72557972278	0.54451235838
Average	0.895948528212	0.515071566622	0.619717568382

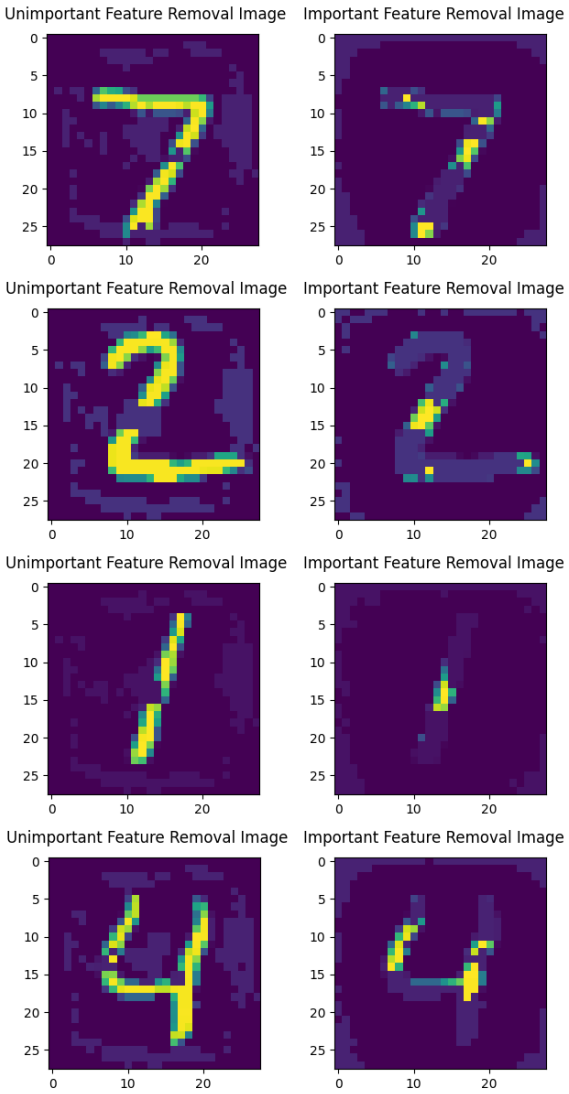

Figure 3: Feature Removal (Important and Unimportant) Input Samples.

Table 2: LRP-0 Confidence Before and After Unimportant and Important Feature Removal
IFRC : Important Feature Removal Confidence
UFRC : Unimportant Feature Removal Confidence

Digit	Prediction Confidence	IFRC	UFRC
0	1.02459982	0.4404331435	0.6737431539
1	1.155775559	1.207955318	0.05174808878
2	0.554388028	0.2143313923	0.3035719208
4	0.834408098	0.6057936905	0.7088502038
7	0.7649750856	0.7137641151	0.7351282032
Average	0.895948528212	0.6364555319	0.4946083141

Results of a sample of these digits can be found in Table 1 and Table 2. Note that the red cells represent incorrectly classified digits. Furthermore, note that in these two tables, the values that exist within the tables are confidence values, which in this case is defined as

$$\frac{argmax(predictions) - secondArgmax(predictions)}{argmax(predictions)}$$

This formula gives approximate confidence - a metric of how significant the margin is between

the maximum predicted class and second maximum. With close inspection, the tables show that when important features are removed, the models have a significantly more challenging time classifying the images than when the unimportant features are removed. Therefore, LRP worked exactly as expected.

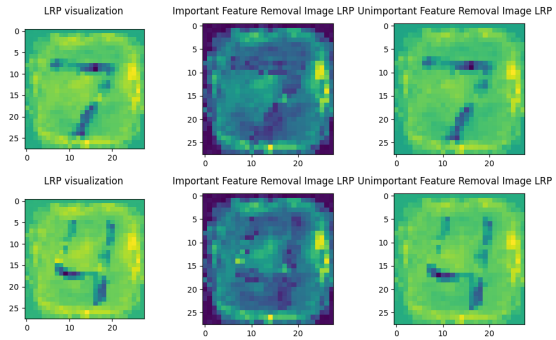

Figure 4: Feature Removal (Important and Unimportant) LRP Samples.

6 Evaluation

Table 1 and Table 2 both show similar results. Mainly, removing important features makes the model classify significantly worse than when unimportant features are removed.

However, using LRP to remove features does not seem to consistently show this trend. For example, using LRP-Gamma digits 0,2, and 4 all follow this trend, while digits 1,7 do not. With LRP-0, digits 0,2,4, and 7 follow this trend, while digit 1 does not. There seems to be a consistent trend. digits that are curved classify well with LRP unimportant feature removal, while straight and more linear digits do not.

With these results, it is unclear whether LRP can be used to remove features in a DNN to improve efficiency and decrease network structure and computation. With our dataset, it would significantly hinder accuracy percentage and therefore is not a viable strategy.

7 Conclusion

A two-layer fully-connected network was constructed with a single sigmoid activation. Using the MNIST dataset, images were classified upon this DNN with a 95+% accuracy. These predictions were used to compute LRP images. Using the LRP images, the least and most important features were computed and used to construct two images from the original images - one image without important features, and one without unimportant features. These images were then fed as input to the network, and classified. These classifications proved that the important features represented by higher LRP values are more necessary to achieve a high accuracy than the unimportant features with lower LRP values. However, these differences are not significant enough. Even when removing unimportant features, the confidence of the predictions suffers, and it is likely the accuracy suffers too. While we proved that LRP generally is viable as an indicator to be used for feature removal techniques, further research is necessary to do so without lowering the prediction confidence and accuracy.

8 Future Works

There are a few directions the work could go. First, adding training post-feature removal could make this strategy of using LRP for determining the possibility of feature removal viable. The table data shows that without extra training, LRP for feature removal is not viable, however the LRP images of post-feature removal shown in Figure 4 support this possibility since the digits are still fully visible and legible. In essence, this research project needs a slight addition to make it a fully-fledged feature-compression method.

Another direction in the future is to experiment with CNN's. In our main chunk of work, a two-layer sigmoid DNN flattens the input and runs it through fully connected layers. The intermediate layers have no legible visual representation. However, using a CNN, the image is scaled down via convolution and sampling. Each layer has its own visual representation that looks like the digits. Using these legible intermediate representations, and the fact that CNN's usually produce great results via image inputs, the LRP results and classification predictions may be significantly improved.

9 References

References

[1] Bach S, Binder A, Montavon G, Klauschen F, Müller K-R, Samek W 2015. *On Pixel-Wise Explanations for Non-Linear Classifier Decisions*

by Layer-Wise Relevance Propagation., PLoS ONE 10(7): e0130140.

[2] Bach S, Binder A, Montavon G, Klauschen F, Müller K-R, Lapuschkin S, Samek W. 2019. *Layer-Wise Relevance Propagation: An Overview.*, PLoS ONE 10.1007/978-3-030-28954-6_10.

[3] Deng, L. 2012. *The mnist database of handwritten digit images for machine learning research.*, IEEE Signal Processing Magazine 29(6), 141–142.

[4] Bach, S., Samek, W., Mueller, K.R., Binder, A. and Montavon, G. Fraunhofer Gesellschaft zur Forderung der Angewandten Forschung eV and Technische Universitaet Berlin. 2018. *Relevance score assignment for artificial neural networks.*, U.S. Patent Application 15/710,455.