**Understanding Image Classification Tasks Using Layer-wise Relevance Propagation**

**Ritu Singh**

University Politehnica of Bucharest

Computer Science Department

Master of Science in Artificial Intelligence, 1st year

singhritu0709@gmail.com

*Under the guidance of*

**Professor Adina Magda Florea**

**Keywords:** Deep Neural Networks, Layer wise relevance propagation, Image Classification

# Introduction

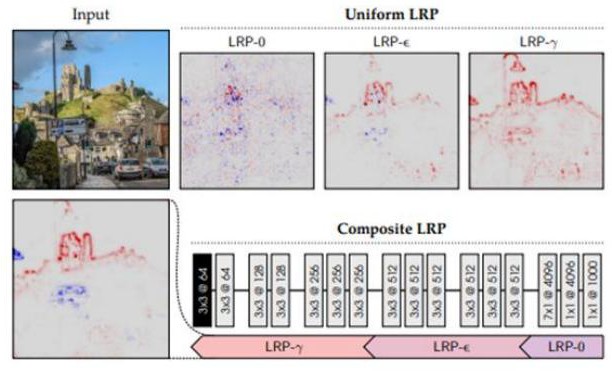
Machine learning techniques such as Deep Neural Networks are successfully in solving a plethora of tasks, e.g., image recognition and text analysis, and provide novel predictive models for complex physical, biological and chemical systems. However, due to the nested complex and non-linear structure of many machine learning models, this comes with the disadvantage of them acting as a black box, providing little or no information about the internal reasoning. This black box character hampers acceptance and application of non-linear methods in many application domains, where understanding individual model predictions and thus trust in the model’s decisions is critical. Recently, the domain of Explainable Artificial Intelligence (ExAI), and in particular ExAI for deep neural networks has gain a lot of attentions. Several methods were proposed in the literature in the attempt to explain how the network has come to a decision.

In this abstract, I focus on a specific method called Layer wise Relevance Propagation (LRP) [1]. In LRP a deep neural network (black box model) makes a prediction based on a given input, then LRP decomposes the model’s evaluated prediction function into differential contributions for all input components, by performing a backward pass through the model. Specifically, the method computes scores for image pixels and image regions denoting the impact of the particular image region on the prediction of the classifier for one particular test image. The resulting relevance map can then optionally be visualized for interpretation by a human observer.

# Layer-wise Relevance Propagation Method

LRP is computed with a backward pass on the network. The algorithm starts at the output layer and assigns the relevance of the target neuron equal to the output of the neuron itself and the relevance of all other neurons to zero. The algorithm proceeds layer by layer, redistributing the prediction score until the input layer is reached. Deep LIFT [2] proceeds in a backward fashion, similarly to LRP. Each unit is assigned an attribute that represents the relative effect of the unit activated at the original network input compared to the activation at some reference. Input Local and Global attribution methods [2] highlight how all the gradient-based methods considered are computed from a quantity that depends on the weights and the architecture of the model, multiplied by the input itself. Similarly, Occlusion-1 [2] can also be interpreted as the input multiplied by the average value of the partial derivatives, computed varying one feature at the time between zero and their ﬁnal value. An attribution method satisﬁes Sensitivity-n [2] when the sum of the attributions for any subset of features of cardinality n is equal to the variation of the output Sc caused by removing the features in the subset.

I consider the application of LRP to deep neural networks with rectifier (ReLU) nonlinearities. It includes well-known architectures for image recognition such as VGG-16 and Inception v3. Here, we consider LRP in the light of two general and well-agreed desirable properties of an explanation: fidelity and understandability. Figure 1 shows for a given input image various LRP explanations of the VGG-16 output neuron ‘castle’. These explanations are obtained by the uniform application of a single propagation rule at all layers, or by a composite strategy where different rules are used at different layers.



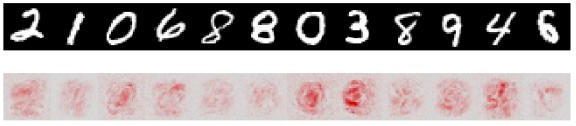
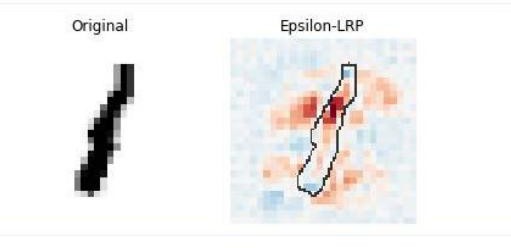
*Figure 1. Input Image and pixel-wise explanations of the output neuron 'castle' trained with various LRP*

*procedures Parameters* [3]

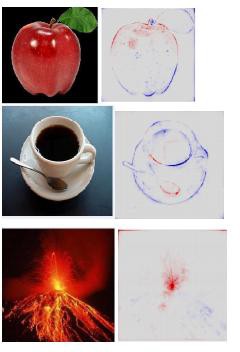
# Experiments

My first implementation is with tensor flow for the MNIST data set[[1]](#footnote-1) with data input (img shape: 28\*28), num\_classes = 10 (0-9 digits), as presented in Figure 2(a). I have created and trained the model, constructed the model by defining loss and optimizer, and later evaluating the model with test logics and while running initializer through, I have calculated batch entropy and obtained accuracy for MNIST test image. Then I have created a DeepExplain context which I called from utils.py , and we run `explain()` several time to compare different attribution methods and finally recreate the network graph using the same weights that have been already trained, and then plotting the resultant of test image.

My second implementation is with PyTorch. In which I considered two models: (1) a simple plain deep rectifier network trained on the MNIST handwritten digits data, and (2) the VGG-16 network trained on Image Net[[2]](#footnote-2) and applicable to general image classification. First I have done Numpy Implementation for a FullyConnected Network by loading 12 exemplary MNIST test digits. After that, I have predicted the class for the MNIST digit and tried explaining prediction with LRP. Then I have done PyTorch Implementation for the VGG-16 Network. In this case, LRP rules are more conveniently implemented by casting the operations of the four-step procedure above as forward and gradient evaluations on these layers. These operations are readily available in neural network frameworks such as PyTorch and Tensor Flow and can therefore be reused for the purpose of implementing LRP. Here, I take the VGG-16 pertained network for image classification. The image is first loaded in the notebook, then converted to a torch tensor of appropriate dimensions and normalized to be given as input to the VGG-16 network. The VGG-16 network is then loaded and its top-level dense layers are converted into equivalent 1x1 convolutions. The results are presented in Figure 2(b)



(a)



(b)

*Figure 2. Obtained results using LRP: (a) first implementation (b) second implementation*

# Conclusion & Future Work

I have briefly presented the subject of my M.Sc. research, namely Gradient\*Input,-LRP, Integrated Gradients and Deep LIFT (Rescale) & Sensitivity-n, and different LRP rules for different layers. Then, I have briefly presented my work aimed to understand and analyze the LRP method through two implementations. In the implementations, I have gained good result for some images but for dense images it could be better, therefore more finetuning is needed. As future work, I am going to experiment the LRP method with various architectures and some other datasets for human face recognition and human gesture recognition.

**References**

[1]Wojciech Samek1 Thomas Wiegand1,2 Klaus-Robert Muller,” EXPLAINABLE ARTIFICIAL INTELLIGENCE: UNDERSTANDING,VISUALIZING AND INTERPRETING DEEP LEARNING MODELS” In:

https://arxiv.org/abs/1708.08296 file:///E:/AI%20MAS%20Research%20Project/EXPLAINABLE%20ARTIFICIAL%20INTELLIGENCE%202017.pdf

[2]Marco Ancona, Enea Ceolini, Cengiz Öztireli, Markus Gross” TOWARDS BETTER UNDERSTANDING OFGRADIENT-BASED ATTRIBUTION METHODSFOR DEEP NEURAL NETWORKS” In:

https://arxiv.org/abs/1711.06104

file:///E:/AI%20MAS%20Research%20Project/TOWARDS%20BETTER%20UNDERSTANDING%20OF%20gradient%20based%20attribution%20methods%202018.pdf

[3]Gr´egoire Montavon1, Alexander Binder2, Sebastian Lapuschkin3, Wojciech Samek3, and KlausRobert M¨uller1,4,). “Layer-Wise Relevance Propagation: An Overview” In:<http://iphome.hhi.de/samek/pdf/MonXAI19.pdf>

1. http://yann.lecun.com/exdb/mnist/ [↑](#footnote-ref-1)
2. http://www.image-net.org/ [↑](#footnote-ref-2)