ANALYSING NEURAL NETWORK ARCHITECTURES WITH LAYERWISE RELEVANCE PROPAGATION

Thesis submitted to the SASTRA Deemed to be University in partial fulfillment of the requirements for the award of the degree of

B. Tech. Information Technology

Submitted by

Ramanadhan Lakshmi Keerthana(120015077)

May 2020

SCHOOL OF COMPUTING

THANJAVUR, TAMIL NADU, INDIA - 61340

ANALYSING NEURAL NETWORK ARCHITECTURES WITH LAYERWISE RELEVANCE PROPAGATION

Thesis submitted to the SASTRA Deemed to be University in partial fulfillment of the requirements for the award of the degree of

B. Tech. Information Technology

Submitted by

Ramanadhan Lakshmi Keerthana(120015077)

May 2020

SCHOOL OF COMPUTING

THANJAVUR, TAMIL NADU, INDIA - 613401

SCHOOL OF COMPUTING

THANJAVUR - 613401

Bonafide Certificate

This is to certify that the thesis titled "**Analysing Neural Networks with Layerwise Relevance Propagation**" submitted in partial fulfillment of the requirements for the award of the degree of B. Tech. Information Technology to the SASTRA Deemed to be University, is a bona-fide record of the work done by **Ms. Ramanadhan Lakshmi Keerthana**(Reg. No. 120015077) during the final semester of the academic year 2019-20, in the **School of Computing**, under my supervision. This thesis has not formed the basis for the award of any degree, diploma, associateship, fellowship or other similar title to any candidate of any University.

Signature of Project Supervisor: S.Srikrishna

Name with Affiliation: Srikrishna Sadula, Sr. Data Scientist, Antworks

Date: 28 May 2020

Project *Viva voce* held on _____

Examiner1 **Examiner2**

SCHOOL OF COMPUTING

THANJAVUR - 613401

Declaration

I declare that the thesis titled "**Analysing Neural Networks with Layerwise Relevance Propagation**" submitted by me is an original work done by me under the guidance of **Mr. Srikrishna Sadula, Senior Data Scientist, Antworks** during the final semester of the academic year 2019-20. The work is original and wherever I have used materials from other sources, I have given due credit and cited them in the text of the thesis. This thesis has not formed the basis for the award of any degree, diploma, associate-ship, fellowship or other similar title to any candidate of any University.

Signature of the Candidate : R L Keerthana

Name of the Candidate : Ramanadhan Lakshmi Keerthana

Date : 28 May 202

ACKNOWLEDGEMENTS

First and foremost I thank the Almighty for helping me successfully complete this project and present it before you. I express my sincere gratitude to **DR.S. VaidhyaSubramaniam**, Vice Chancellor and **Dr.R. Chandra Mouli,** Registrar, SASTRA Deemed to be University, for providing the opportunity to do this project as a part of my curriculum. I am extremely grateful to **Dr. A. Umamakeswari, Dean, School of Computing, SASTRA Deemed University** for providing all the required resources for the successful completion of this project. I express my profound gratitude to **Dr. Muthaiah R**, **Associate Dean of Information Technology**, SASTRA Deemed to be University for his complete support throughout the Project. My heartfelt thanks to **Dr. Prasanna Shrinivas V, Vice President Cognitive Machine Learning, AntWorks** for recommending the topic and **Mr. Srikrishna Sadula, Senior Data Scientist, AntWorks** for his valuable suggestions and guidance throughout the project. I will be failing in duly if I do not acknowledge the authors of the references and other literature referred to in this report. Last but not the least, I am very much thankful to my parents for guiding me in every step which I took

ABSTRACT

Machine learning models are solving a plethora of tasks, especially in the fields of medicine, vehicle automation etc. In order to check the feasibility of a particular model, it is important to keep the predictions transparent. However, these neural network decisions are often perceived as being highly abstract and the process of arriving at a decision is obscure. A prior knowledge regarding the functioning of each model is required for better understanding. Layer-wise Relevance Propagation(LRP) is one such technique which analyses any network and brings about the explainability. This method provides a detailed explanation for non-linear classifiers using the backward pass which is a conservative relevance redistribution procedure. This procedure indicates the neurons that contribute the most to the higher-layer receive most relevance from it which can be seen in a heatmap. The propagation rules used by LRP can be used for many architectures and can be easily implemented in most of the programming languages. This ensures both flexibility and transparency, helping us gain trust in the prediction.

Keywords: Deep Neural Networks, Heat Mapping, Layerwise Relevance Propagation

TABLE OF CONTENTS

Title	Page No.
Bona-fide Certification	i
Declaration	ii
Acknowledgements	iii
Abstract	iv
List of Figures	vii
Abbreviations	viii
Notation	ix
Certificate	x
1. Introduction	1
2. Layerwise Relevance Propagation	3
2.1 LRP Rules	3
2.2 Variants of LRP Rules for DNNs with ReLu Activation	4
2.2.1 Basic Rule	4
2.2.2 Epsilon Rule	5
2.2.3 Gamma Rule	5
2.3 Practical Implementation of LRP Rule	5
3. LRP as a Deep Taylor Decomposition	7
4. Experimental Results	10
4.1 LRP Application on MNIST Dataset	10
4.1.1 Output Layer	10
4.1.2 Higher Layer	10

4.1.3 Lower Layer 11

4.2 LRP Application on VGG-16 Network 12

4.2.1 Convolution Layers 12

4.2.2 Pooling Layers 12

4.3 Applications of LRP in various Domains 14

5. Conclusion and Future Plans 16

6. References 17

LIST OF FIGURES

Figure No.	Title	Page No.
1.1	The input image passed through vgg-16 and the heatmap computed by LRP	1
2.1	Computational flow of LRP procedure	4
2.2	Computational Decomposition of LRP generic rule	6
2.3	Pictorial representation of the variables used in the decomposition	6
3.1	Decomposing prediction to calculate Relevance scores	7
4.1	Neural Network Architecture used to train MNIST Handwritten Data.	10
4.2	Output of MNIST model	11
4.3	The VGG-16 Architecture	12
4.4	Different LRP Rules at different layers of VGG-16 Network	13
4.5	LRP output of VGG-16 model	14

ABBREVIATIONS

DNN	Deep Neural Networks
DTD	Deep Taylor Decomposition
LRP	Layer-wise Relevance Propagation
ReLU	Rectified Linear Unit
VGG	Visual Geometry Group

NOTATION

English Symbols

a_i	Activation of the neuron *i*
$b_i^{(l)}$	Bias of the neuron *i* in the layer *l*
R_j	Relevance Score of the neuron *j*
$w_{ij}^{(l,m)}$	Weight of the edge from neuron *i* layer *l* to neuron *j* in layer *m*
$x_i^{(l)}$	Input to the neuron *i* in the layer *l*

Greek Symbols

ε	Epsilon
γ	Gamma
Σ	Summation

Miscellaneous Symbols

ã	Root point for Taylor Series Expansion
f(x)	Activation score in the output layer
max(a,b)	Returns the Maximum input parameter

CERTIFICATE

MAPLES IMAGING SOLUTION
An ArtWorks Company

June 03, 2020

TO WHOMEVER IT MAY CONCERN

This is to certify that **Ms. Ramanadhan Lakshmi Keerthana (I0051)** has successfully completed 6 months Internship with us from 9-Dec-2019 to 29-May-2020.

She was working as **Intern - Machine Learning** with Cognitive - Data Science Business Unit.

During the internship she demonstrated self-motivated attitude to learn new things and contributed to the creative ideas.

We wish her all the best for her future endeavors.

For Maples Imaging Solution Private Limited

Kevin Bradley
Senior Vice President – Human Wellbeing

Maples Imaging Solution Private Limited
2nd Floor, Core 2, Module No. 2L, Pacifica Tech Park, Survey No. 76, No. 23 Rajiv Gandhi Salai, OMR Navalur, Chennai – 600 130
+91-44-6699-9696 | www.art.works
CIN No: U74900PN2013PTC176432

1.INTRODUCTION

Deep Neural Networks (DNNs) turned out to be enormously used in the domain of automated image classification by showing incredible results on large benchmark data sets. They have also been used in other fields such as Natural Language Processing, Speech Recognition, Human Action Recognition, Physics etc.Though these models are highly productive in terms of results, they have a downside in behaving like a black box, not presenting any detail about what made them conclude at a specific decision. When certain fields such as medical applications use Network predictions, where understanding the interpretation of a decision is as important as the decision itself, this lack of transparency and clarity is a significant drawback as it prevents us from being able to verify and understand the system's logic. Deep Learning researchers are using a wide range of tools and techniques to supervise the Neural Network's learning process to combat this problem.

LRP has become one such popular approach in Explainable Machine Learning to provide an explanation for any Neural Network's output in the domain of its input. LRP deals with the problem of understanding what neurons encode and which neurons are most important for the prediction of an image. Relying on this detailed evidence, it is possible to delete bad features, and the model can be retrained to boost model performance.

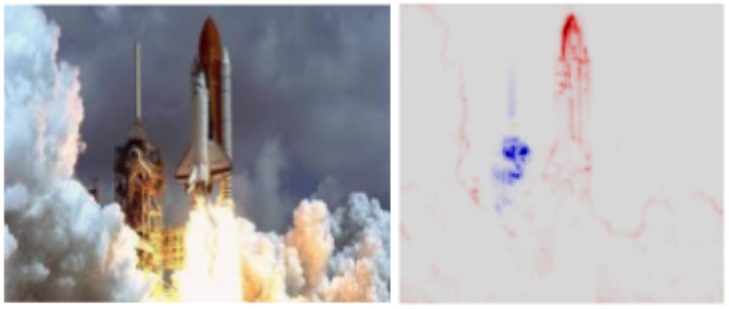

Fig 1.1.The input image passed through vgg-16 and the heatmap computed by LRP.

For example, Fig.1.1 shows a correctly classified image for class *Space Shuttle* by VGG-16 network. But it is not clearly known what made the model correctly classify the input image.A heatmap computed with Layer-wise Relevance Propagation shown on the right side of Fig.1.1 shows that the most contributory areas are the shade of red outlining the space shuttle's apex and the ejected smoke's silhouette.

This technique, however, does not seek to assign a defined interpretation to the assigned pixel values, except for the fact that they will form a pattern that is visually interpretable.

The LRP method applies to each Network Neuron for certain models of the Convolutional Neural Network, a propagation rule with a conservative property which results in the assignment of values to pixels which can be directly understood as their impact in the classification decision. While highly quantitative, the choice of propagation rules was mainly heuristic and lacked a clear theoretical explanation which can be realized in the Taylor decomposition of the Relevance function by estimating the impact of input variables for a prediction discussed in the chapters to come.

2. LAYERWISE RELEVANCE PROPAGATION

2.1 LRP RULES

LRP strategies can be implemented to Neural Network models, where inputs can be pixels, motion of pixels or document. LRP operates by further propagating the estimate function, f(x), backward within the Neural Network, using intentionally formulated propagation rules that are prone to a conservation rule. Whatever a neuron has already gathered has to be redistributed uniformly to the lower levels. A DNN seems to be a feed-forward graph upon the elements of basic computing, each of which has a simple type:

$$x_k^{(l+1)} = \max(0, \sum_j x_j^{(l)} w_{jk}^{(l,l+1)} + b_k^{(l+1)}) \quad (2.1)$$

where k indexes a neuron to a specific layer l+1, where $\sum_j$ runs over neurons of the lower layer connected to neuron k. Here, $w_{jk}^{(l,l+1)}$ and $b_k^{(l+1)}$ are parameters that are unique to adjacent neuron pairs and learned from the results. By implementing the following rule, the absorption of relevant scores (Rk)k at a particular layer on lower layer neurons is evaluated.

$$R_j = \sum_k \frac{z_{jk}}{\sum_j z_{jk}} . R_k \quad (2.2)$$

Here, on any consecutive layers, j and k are two neurons, and the quantity on z_{jk} models to what degree neuron j has apparently made neuron k significant. As the relevance R is defined in the output layer, we must start from there and iteratively use the Eq.(2.2) to measure R for each neuron of the previous layer. The propagation process stops until the input functions have been reached as seen in Fig.2.1.

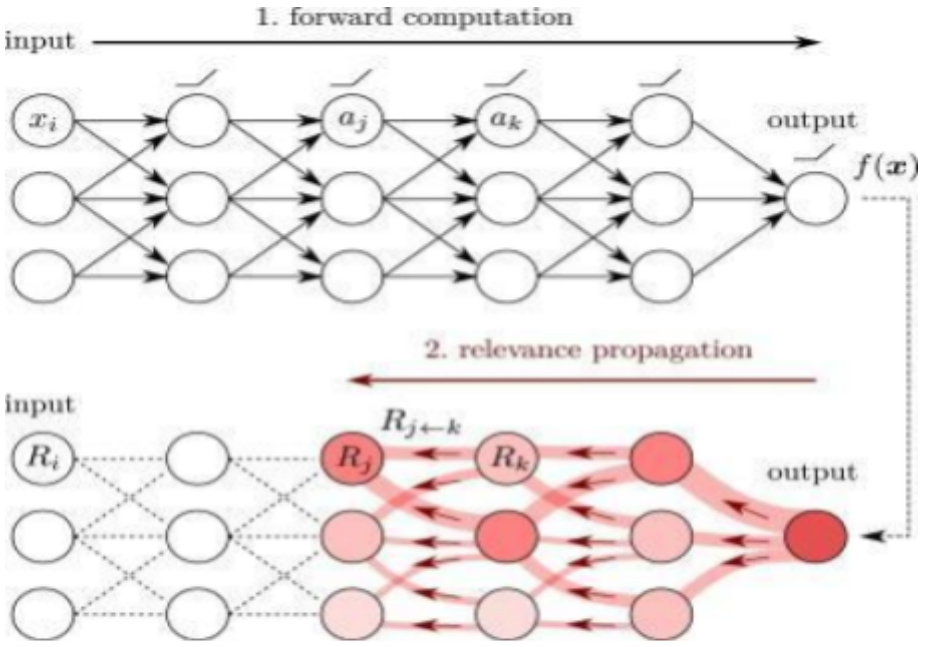

Fig. 2.1.Computational flow of LRP procedure.

It is observed from Fig.2.1 that LRP satisfies the conservation property which is mathematically written as,

$$\sum_i R_i = f(x) \quad (2.3)$$

2.2 VARIANTS OF LRP RULES FOR DNNs WITH ReLu ACTIVATION

Here, we analyse the impact of LRP to DNNs with ReLU nonlinearities. The activation function is these networks is,

$$a_k = \max(0, \sum_{0,j} a_j w_{jk}) \quad (2.4)$$

In addition to an additional neuron representing the bias, the number, $\sum_{0,j}$, extends across the smaller layer activations (aj). Those networks have three variants of rule of propagation.

2.2.1 Basic Rule (LRP- 0):

This is the simplest and also called the basic rule that meets the basic LRP principle dealt

within Eq.(2.2). This rule redistributes activation of the neurons according to each input 's contributions. By implementing this principle to the entire network gives the model explainability.

$$R_j = \sum_k \frac{a_j w_j}{\Sigma a_j w_{jk}} R_k \quad (2.5)$$

2.2.2 Epsilon Rule (LRP - ε):

The gradient of a DNN is usually chaotic and hence unstable. We can make the basic rule more robust by appending a small positive constant in the divisor. The constant **ε** is added to dissipate some relevance when the activation of neuron *k* is either inadequate or inconsistent.

$$R_j = \sum_k \frac{a_j w_{jk}}{\varepsilon + \Sigma_{0,j} a_j w_{jk}} R_k \quad (2.6)$$

As ε advances in number, the absorption survives only the most contentious explanatory features which tends to sparse explanations regarding the input characteristics and less congested which can be observed in the generated heatmap.

2.2.3 Gamma Rule (LRP - γ):

Another enhancement can be accomplished by emphasizing the effects of positive impacts against negative impact. Positive contributions are preferred by the parameter γ. As γ rises, negative contributions slowly disappear, resulting in a more stable neural network explanation. The Gamma law is laid down as follows:

$$R_j = \sum_k \frac{a_j \cdot (w_{jk} + \gamma w^+_{jk})}{\Sigma_{0,j} a_j \cdot (w_{jk} + \gamma w^+_{jk})} R_k \quad (2.7)$$

2.3 PRACTICAL IMPLEMENTATION OF LRP RULE

This section addresses the generic rule for implementation purposes. The rules for the propagation of relevance on the lower layers described in the preceding section are special cases of the more general propagation rule:

$$R_j = \sum_k \frac{a_j \cdot \rho(w_{jk})}{\varepsilon + \Sigma_{0,j} a_j \cdot \rho(w_{jk})} R_k \quad (2.8)$$

As seen in Fig.2.2 below, the estimation of this propagation rule can be decomposed into

four segments, and the pictorial representation is shown in Fig.2.3.

$$\forall_k : \; z_k = \epsilon + \sum_{0,j} a_j \cdot \rho(w_{jk}) \qquad \text{(forward pass)}$$
$$\forall_k : \; s_k = R_k / z_k \qquad \text{(element-wise division)}$$
$$\forall_j : \; c_j = \sum_k \rho(w_{jk}) \cdot s_k \qquad \text{(backward pass)}$$
$$\forall_j : \; R_j = a_j \, c_j \qquad \text{(element-wise product)}$$

Fig. 2.2. Computational Decomposition of LRP generic rule.

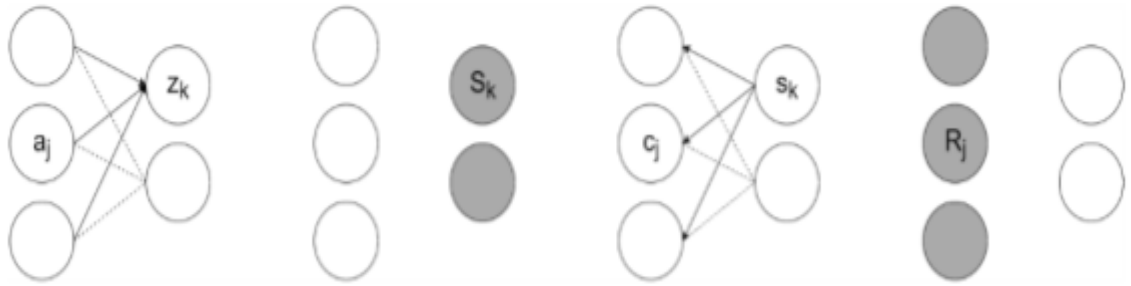

Fig. 2.3.Pictorial representation of the variables used in the decomposition.

The *first step* identifies the sum of influences in the higher layer for each neuron to which we add a small constant ϵ and roll it up around the weights a function ρ . This makes the formula more general and contains all potential LRP laws. The sum goes into any neuron *j* in the lower layer as well as the neuron bias. The bias will be overlooked in all the following phases, since we want relevance to only pass to the input neurons and not to eventually wind up in static neurons.

During the *second step* the significance of each neuron in the higher layer is segregated by the *z* value determined in the preceding step. It guarantees the preservation of property, as stated in Eq.(2.3).

In the *third step* a basic element-wise calculation on determined quantity c for each neuron in the preceding layer. Therefore it could be seen as a backwards pass. This *c* can be loosely seen as the sum of significance from the subsequent layer which trickles down to neuron j.

Finally, in the *fourth step*, the relevance that comes from the above step is multiplied by

the neuron activation to calculate its own relevance.

3. LRP AS A DEEP TAYLOR DECOMPOSITION

LRP is an Interpretability technique that works well but is still heuristic. We need a theory to understand what this technique is actually doing. The rules for the propagation of LRP may be interpreted within the framework of the Deep Taylor Decomposition(DTD). The Relevance scores calculated using LRP tells the importance of every input variable. These Relevance scores are actually formed by decomposing the multivariate prediction function f(x1,x2,...., xd), where x1,x2,....xd are the input variables, which is shown in the figure below.

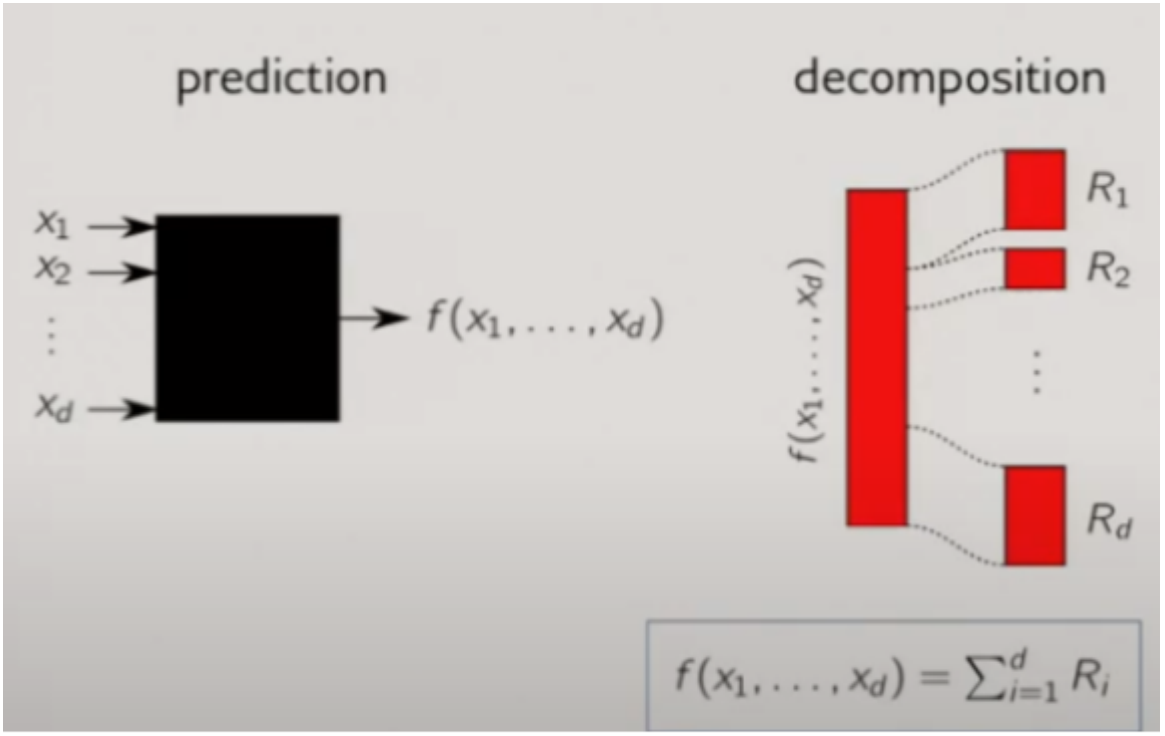

Fig. 3.1.Decomposing prediction to calculate Relevance scores.

The Relevance propagation from one layer to the other layer explained in Eq.(2.2) is now expressed as a taylor decomposition. We choose the DTD framework over simple Taylor Decomposition because the root point in the latter is hard to find and most of the DNNs have shattering gradient structures. So let's assume that relevance(R_k) essentially is a product of the activation and a constant.Therefore, from Eq.(2.2)

$$R_k = a_k c_k \tag{3.1}$$

Assuming Eq.(3.1) is true, Eq. 2.2 can be written as,

$$R_j = a_j \sum_k w^+_{ij} \frac{max(0, \sum_l a_l w_{jk})}{\sum_j a_l w^+_{jk}} \cdot c_k \quad (3.2)$$

Assuming all the values in Eq.(3.2), are constant except a_j. The equation can be rewritten as:

$$R_j = a_j c_j \quad (3.3)$$

From the above equations, it can be interpreted that the Relevance Score is a product of activation and a constant. This interpretation holds true in the top layer because we set the Relevance value to the output activation score. Now, we will let the Relevance function depend on the lower layer,

$$R_k((a_j)_j) = a_k((a_j)_j) \times c_k((a_j)_j)$$

$$\begin{aligned} R_k((a_j)_j) &= \max(0, \textstyle\sum_j a_j w_{jk} + b_k) \times c_k((a_j)_j) \\ &= \max(0, \textstyle\sum_j a_j w_{jk} c_j + b_k c_j) \\ &= \max(0, \textstyle\sum_j a_j w'_{jk} + b'_k) \end{aligned} \quad (3.4)$$

Now we apply taylor expansion to this particular neuron as follows:

$$R_k((a_j)_j) = R_k((\tilde{a}_j)_j^{(k)}) + \sum_j \frac{\partial R_k}{\partial a_j}\Big|_{(\tilde{a}j)j(k)} \cdot (a_j - \tilde{\mathbf{a}}_j^{(k)}) + \varepsilon \quad (3.5)$$

Here, we have a Relevance function that depends on the input scores. And at the root point, $\tilde{\mathbf{a}}$, Taylor expansion is computed where the first term has the First order values and the remaining terms have higher order values. From the Top view of the neural network architecture, the network has only the linear terms. So Eq.(3.5) can be rewritten as,

$$R_k((a_j)_j) = 0 + \sum_j \frac{\partial R_k}{\partial a_j}\Big|_{(\tilde{a}j)j(k)} \cdot (a_j - \tilde{\mathbf{a}}_j^{(k)}) + 0$$

which can be approximated as,

$$R_{j \leftarrow k} = \frac{v_{jk} w_{jk}}{\sum_i v_{jk} w_{jk}} R_k \quad \text{with, } v_{jk} \propto a_j - \tilde{\mathbf{a}}_j^{(k)}, \text{ the root search direction} \quad (3.6)$$

When we compare Eq.(2.2) and Eq.(3.6), they look almost similar. The only differences are:

1) The summation, because in Eq.(3.6), we haven't considered all the Relevant neurons coming from the higher level. This can be compensated by adding pooling relevance over all the outgoing neurons.

2)We have to choose a specific root point to match both the equations. The search

direction should be chosen such that,

$$v_{jk} = a_j \, 1_{w_{jk} > 0} \tag{3.7}$$

Each root point choice, $\tilde{\mathbf{a}}$,gives rise to a realignment. Particular root point selection inevitably leads a way to the LRP propagation rules defined in Section 2.2. LRP-0 is extracted with an option of $\tilde{\mathbf{a}} = 0$. LRP-ε is obtained by choosing $\tilde{\mathbf{a}} = \varepsilon \cdot (a_k + \varepsilon)^{-1} \mathbf{a}$.

For Deep Rectifier Networks, LRP can be explained as a DTD of the Neural Network function. Thus, by demonstrating the relation between the propagation law and the input domain, DTD offers insights into LRP.

4. EXPERIMENTAL RESULTS

The LRP method has been applied to two Neural Network models: MNIST and VGG-16 and the results are discussed in the following section.

4.1 LRP APPLICATION ON MNIST DATASET

The digits stored as a 784-dimensional pixel-value vector are fed to a fully connected 2-hidden-layer network. For each hidden layer the activation function ReLU is used. The Neural Network architecture is represented in Fig.4.1.

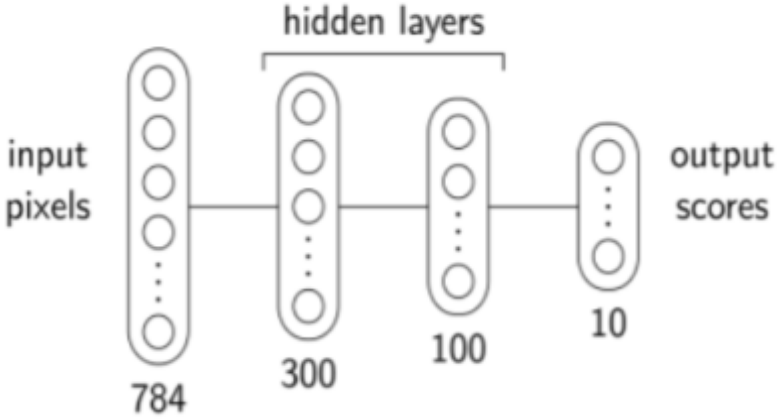

Fig. 4.1.Neural Network Architecture used to train MNIST Handwritten Data.

LRP offers consistency on which layer to implement rules. There were clear variations between the reasons obtained by applying the single propagation rule uniformly to all layers and those obtained by means of a hybrid method. So different propagation laws were merged and added to the model to get the best results. For each layer the rules were applied as follows:

4.1.1 Output Layer

The top layer relevance scores are set to the respective activations that we multiply by a label indicator to retain only the actual class proof. Therefore, the LRP-0 rule is ideal to use.

4.1.2 Higher Layer

That is the hidden layer next to the output layer in this instance. The LRP-ε rule is applied with ε = 0.1std, since the constant ε will nullify a small amount of contradictory data.

4.1.3 Lower Layer

The layer next to the input layer is the lower layer. Since this layer is so close to the output heat map, the interpretation is kept simple so that the explanation should be clearer, simpler and less distracting. To this reason, the LRP-γ theory is used to assign positive evidence significantly over misleading evidence.

The LRP Rules are applied layer by layer till input layer before reaching the pixels. To propagate the relevance scores until the pixels, we apply z^{β}-rule given by:

$$R_j = \sum_j \frac{a_i w_{ij} - l_i w^{+}_{ij} - h_i w^{-}_{ij}}{\sum_i a_i w_{ij} - l_i w^{+}_{ij} - h_i w^{-}_{ij}} R_j \quad (4.1)$$

The obtained pixel-wise relevance scores can be produced as a heatmap after applying the LRP Rules to all layers accordingly. The heatmap generated for an incorrectly classified image is shown in Fig.4.2.

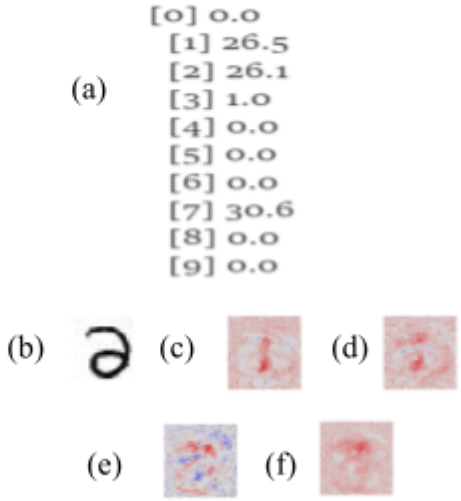

Fig. 4.2.(a) Scores to the each class given by the model (b) Input image passed to the model (c) Heatmap produced backprogating from *Class 1* (d) Heatmap produced backprogating from *Class 2* (e) Heatmap produced backprogating from *Class 3* (f) Heatmap produced backprogating from *Class 7.*

Even the following picture belongs to *Class 2*. But the model predicted it as the *Class 7* image. So we propagated the model back from the classes that are likely to better represent the input, and generated the heatmap to understand the prediction of the Neural Network. The process of reaching that particular decision by the Neural Network is made

clear by the heatmap.

4.2 LRP APPLICATION ON VGG-16 NETWORK

VGG-16 is an architecture of a Convolution Neural Network, as shown in Fig.4.3. Here we use the pretrained VGG-16 network for the classification of images. LRP rules are implemented more easily using the four-step procedure operations, as shown in Fig.2.2, as forward and gradient evaluations on those layers. These operations are available instantly on neural network frameworks such as PyTorch and Tensorflow.

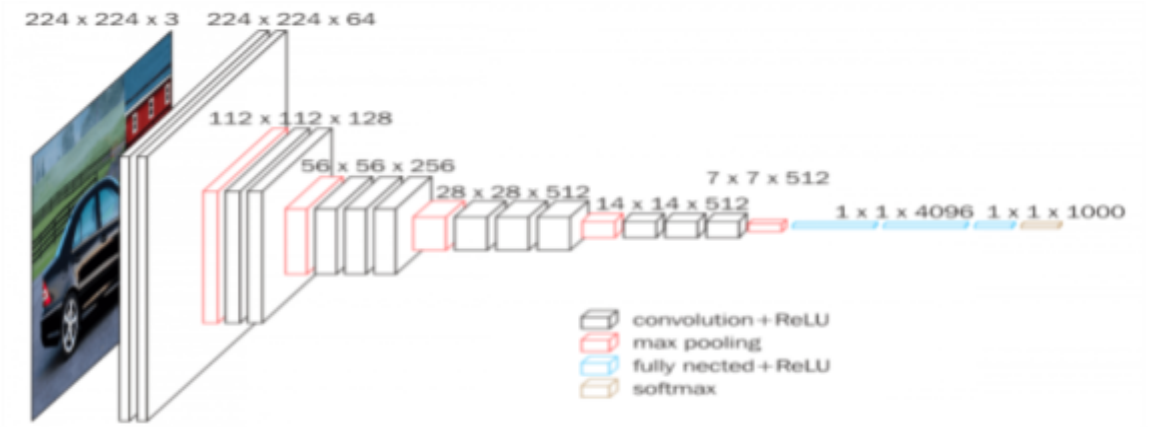

Fig.4.3.The VGG-16 Architecture.

In reverse order, we iterate from the top layer to the first layer and apply propagation rules at each layer. Activations of the top layer are first compounded by the mask so as to preserve only the estimated evidence. As discussed in Section 4.1, various LRP rules are implemented at different levels as seen in Fig.4.4.

4.2.1 Convolution Layers

Since they are also a type of Linear Layers, we are going to apply the rules in the same four-step procedure. But, *Step3* is applied slightly in a different way. From Fig.2.2, the input activations available in the PyTorch System can be computed as a gradient.

$$c_j = [\nabla \ (\sum_k z_k(a) \cdot s_k)]_j \quad (4.2)$$

4.2.2 Pooling Layers

These are basically used in between the convolution layers, to subside the number of

parameters and to reduce the computation cost . Here in the backward pass, max-pooling layers are treated as average pooling layers which is considered as a linear layer. The same propagation rules apply as for the Convolution layers

We have applied LRP-0 rule for the layers 31-38, LRP-ε for layers 17-30 and LRP-γ for the layers 1-16 as shown below in Fig.4.4.

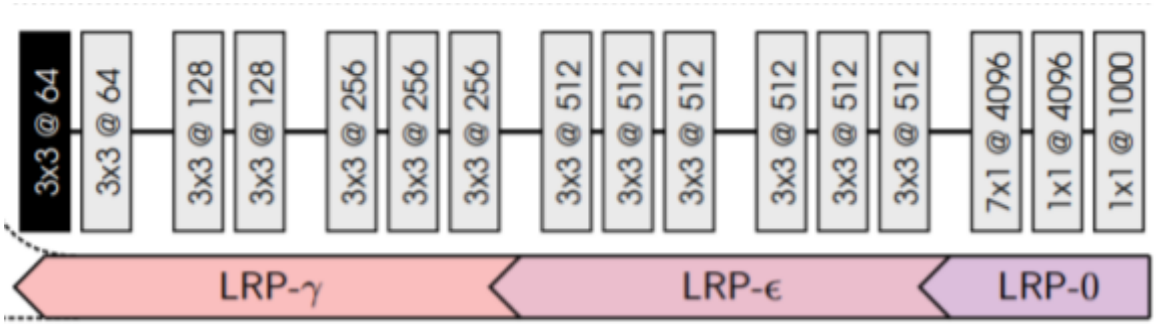

Fig. 4.4.Different LRP Rules at different layers of VGG-16 Network.

Fig.4.5 shows an explanation of the LRP rules that apply to the model. The model has precisely predicted the image from the image to be a *castle* as seen in Fig.4.5(a). It is apparent upon backpropagation that the red pixels highlighting the castle towers are the reason for the decision. From more research, we will get to various possibilities in related groups such as *Traffic Light* and *Street Sign*.Fig.4.5(b) indicates the probability of a *Traffic Light* image, by the heatmap that encapsulates the features in red pixels. Fig.4.5(c) shows the probability that the image will be a *Street Sign* due to the red pixels highlighted at the outline. The two groups *Traffic Light* and *Street Sign* had a comparatively lower ranking, and were thus considered less important.

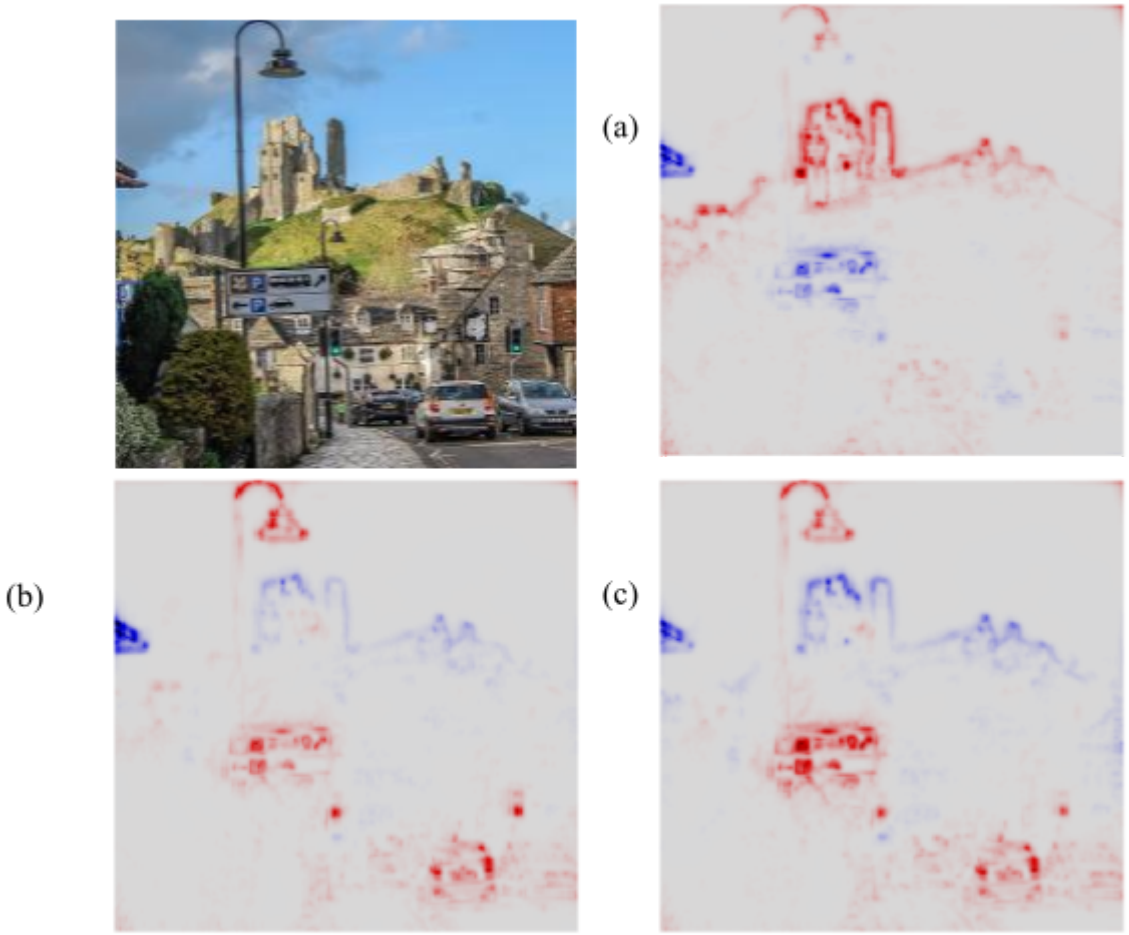

Fig.4.5 shows how the image has been correctly classified as class castle. a) Here the red pixels highlight the towers of the castle and has the highest relevance b) In this image, the red pixels highlight the traffic lights which is apparent through backpropagation c) Shows the street signs heavily marked in red pixels.

From the above examples, it is understood that the application of LRP rules on the models can help come up with an explanation on how the models arrive at a particular decision with a visual representation in the form of heatmaps. This also paves way in understanding how a model arrives at an inaccurate decision and further improvises the architecture of the Neural Network model. The application of the rules, thus helps us increase the efficiency and error detection in the given model.

4.3 APPLICATIONS OF LRP

LRP is easy to implement and has a very short processing time. It has diverse applications. Some of them include:

a. Machine Learning predictions in Clinical Gait Analysis can be easily interpreted.

b. Applications such as, topic categorization, sentiment analysis, machine translation, structured information extraction, and automatic summarization need a huge amount of data. LRP enables one to extract relevant information from text documents without an explicit information extraction step.

c. LRP can also be extended to other Image and Face Recognition applications, such as age & gender classification, facial expression recognition, etc. The description approach gives valuable insights into the essence of the basic model functionality, allowing one to assess the basic model's aptitude for a specific transfer learning task.

d.LRP is also used, by calculating the relevant scores, to determine the most relevant elements such as weights or filters. Using these scores one can prune transfer-learned CNN models easily without sacrificing the results. Since it has a computational cost in the order of gradient measurement and is fairly easy to implement without the need to change the hyperparameters for pruning, it can be applied on any Neural Network Architecture.

5. CONCLUSION AND FUTURE PLANS

LRP works well for most of the Neural Networks and can be applied efficiently and modularly. From the above chapter, we can infer that LRP is not a mere gradient detector. Unlike a gradient detector, it highlights only the edges which are relevant to the prediction.This feature is very much useful in knowing the black box architecture of many complex Neural Network structures. The concept can be further improved by finding a way to choose the parameters efficiently for each layer and which rule can be to which layer. A strategy for fine tuning the LRP rule would help us achieve more explainability of the network. To conclude, LRP can be further extended to a broad range of problems as discussed in the Section(4.3) and a large number of real applications where theoretical justification is required.

REFERENCES

1.**Alexander Binder, Sebastian Bach, Gregoire Montavon, Klaus-Robert Muller and Wojciech Samek**(2016), Layer-wise Relevance Propagation for Deep Neural Network Architectures.

2. **Gregoire Montavon, Alexander Binder, Sebastian Lapuschkin, Wojciech Samek and Klaus-Robert Muller**(2019), Layer-Wise Relevance Propagation: An Overview.

3. **Gregoire Montavon, Sebastian Lapuschkin, Alexander Binder, Wojciech Samek and Klaus-Robert Muller**(2016), Explaining nonlinear classification decisions with deep Taylor decomposition

4. http://www.heatmapping.org/

5.https://towardsdatascience.com/indepth-layer-wise-relevance-propagation-340f95deb1ea