

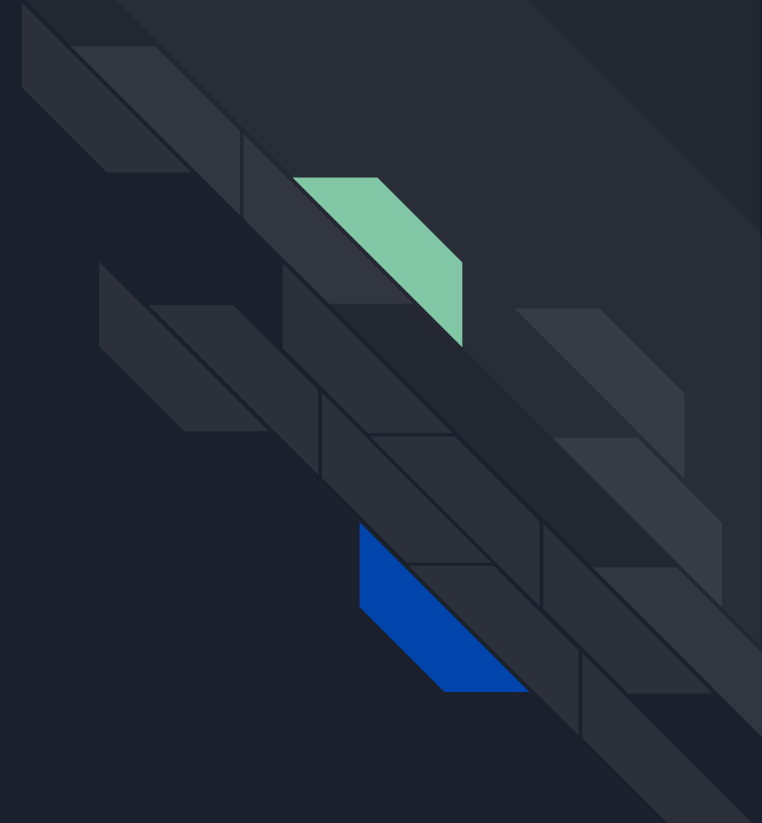
The background is a dark navy blue. In the top-left corner, there are two overlapping geometric shapes: a blue parallelogram and a light green parallelogram. In the bottom-left corner, there is a circular inset showing a detailed, grayscale image of a printed circuit board (PCB) with various electronic components. In the top-right corner, there is a faint, grayscale image of a complex circuit board layout with many traces.

# LRP

Seimandi Juliette - Thoirey Romane - Robin Gilles

# TABLE OF CONTENTS

1. Introduction
2. Theoretical study
3. Demonstration
4. Conclusion





# What is LRP?

- Layer-wise Relevance Propagation (LRP) : It's an explanation tech applied to neural networks, where pictures video, text
- The goal : Generate explanation of the classification decisions made by the algo through analyzing deeply the model neuron by neuron
- The functionment is based on backward propagation



# Visual introduction

[Explainable AI Demos \(fraunhofer.de\)](https://www.fraunhofer.de/en/explainable-ai-demos)

# How works LRP?

Propagation :

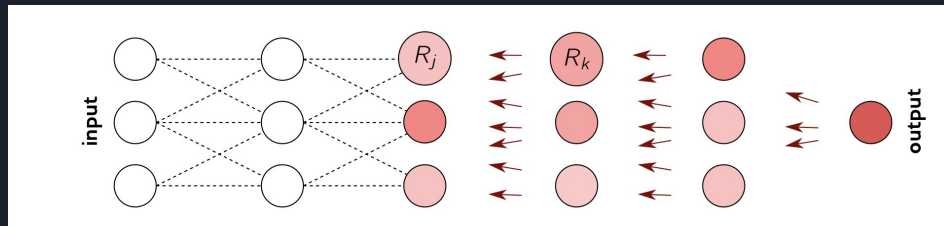
$$X_i^{L+1} = \phi(\mathbf{X}^L \mathbf{W}_i^L + \mathbf{b}_i^L)$$

Taylor decomposition :

$$\begin{aligned} f(x_0 + h) &= f(x_0) + hf'(x_0) + \frac{h^2}{2!}f^{(2)}(x_0) + \cdots + \frac{h^n}{n!}f^{(n)}(x_0) + h^n\varepsilon(h) \\ &= \sum_{k=0}^n \frac{h^k}{k!}f^{(k)}(x_0) + h^n\varepsilon(h) \end{aligned}$$

Propagating relevance scores

$$R_j = \sum_k \frac{z_{jk}}{\sum_j z_{jk}} R_k.$$



# Rules of LRP

Basic Rule (LRP-0):

$$R_j = \sum_k \frac{a_j w_{jk}}{\sum_{0,j} a_j w_{jk}} R_k$$

Epsilon Rule (LRP- $\epsilon$ ) :

$$R_j = \sum_k \frac{a_j w_{jk}}{\epsilon + \sum_{0,j} a_j w_{jk}} R_k$$

Gamma Rule (LRP- $\gamma$ ) :

$$R_j = \sum_k \frac{a_j \cdot (w_{jk} + \gamma w_{jk}^+)}{\sum_{0,j} a_j \cdot (w_{jk} + \gamma w_{jk}^+)} R_k$$

Name	Formula	Usage	DTD
LRP-0 [7]	$R_j = \sum_k \frac{a_j w_{jk}}{\sum_{0,j} a_j w_{jk}} R_k$	upper layers	✓
LRP- $\epsilon$ [7]	$R_j = \sum_k \frac{a_j w_{jk}}{\epsilon + \sum_{0,j} a_j w_{jk}} R_k$	middle layers	✓
LRP- $\gamma$	$R_j = \sum_k \frac{a_j (w_{jk} + \gamma w_{jk}^+)}{\sum_{0,j} a_j (w_{jk} + \gamma w_{jk}^+)} R_k$	lower layers	✓
LRP- $\alpha\beta$ [7]	$R_j = \sum_k \left( \alpha \frac{(a_j w_{jk})^+}{\sum_{0,j} (a_j w_{jk})^+} - \beta \frac{(a_j w_{jk})^-}{\sum_{0,j} (a_j w_{jk})^-} \right) R_k$	lower layers	×*
flat [30]	$R_j = \sum_k \frac{1}{\sum_j 1} R_k$	lower layers	×
$w^2$ -rule [36]	$R_i = \sum_j \frac{w_{ij}^2}{\sum_i w_{ij}^2} R_j$	first layer ( $\mathbb{R}^d$ )	✓
$z^B$ -rule [36]	$R_i = \sum_j \frac{x_i w_{ij} - l_i w_{ij}^+ - h_i w_{ij}^-}{\sum_i x_i w_{ij} - l_i w_{ij}^+ - h_i w_{ij}^-} R_j$	first layer (pixels)	✓



# Implementation



# Conclusion :

