CV Project Report

21K-3153

21K-3372

21K-4596

**Group Report: Understanding Layer-wise Relevance Propagation (LRP)**

**Objective**

In this project, we explored **Layer-wise Relevance Propagation (LRP)**, a technique for interpreting the predictions of neural networks. Our goal was to understand how LRP works in practice and to apply it to visualize how deep learning models make decisions, particularly on image data. By using the official lrp\_toolbox, we aimed to gain hands-on experience with relevance-based interpretability methods in computer vision.

**What We Learned About LRP**

LRP is a technique used to **explain how a machine learning model, especially a neural network, made a specific decision**.

Imagine you give an image to a neural network, and it says, *"This is a cat."* LRP helps you **understand which parts of the image (like the ears, eyes, or whiskers) were most responsible** for that prediction. It does this by assigning **"relevance scores"** to the input features (e.g., pixels).

**Layer-Wise Relevance Propagation (LRP)** is a method to **understand how neural networks make decisions**. It helps figure out which parts of the input (like pixels in an image) were most responsible for the network’s final output (e.g., classifying an image as a "cat").

Think of it like **reverse-engineering the decision** made by the model — starting from the output and tracing back to the input to see **what was important**.

**🔄 How does LRP work?**

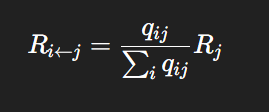
1. **Forward Pass**: A neural network processes an input (like an image) using weights and biases to produce an output. The model has already made a decision — say, 98% confidence it's a cat.
2. **Backward Pass with LRP**: Instead of gradients, LRP sends back **relevance scores** from the output to the input layer to find out what contributed most to the decision. LRP takes this output (e.g., 0.98) and **propagates it backward** through the layers of the network all the way to the input layer (the pixels of the image).
3. **Divide the Output into Contributions:**  
   It breaks down the prediction into parts (relevance scores) that **show how much each input contributed** to the final result.  
   So, if R1, R2, ..., Rd are the relevance scores of the input features (pixels), they all add up to the final prediction value:  
   R1 + R2 + ... + Rd = f(x).
4. **Make a Heatmap:**  
   These scores can be visualized as a **heatmap** over the input (e.g., the image), so you can **see which regions were important** for the prediction.

At each neuron:

* The relevance coming into a neuron is **split among its input neurons**, based on how much each input contributed.
* This split follows a **conservation rule**: the total relevance going into a neuron equals the total relevance passed to its inputs.

**🔢 How are relevance scores split?**

The key formula is:



This means:

* Rj is the relevance score of neuron **j**.
* It's split among input neurons **i** in proportion to qij, which tells us **how much i contributed to j**.

**⚙️ What are the different rules?**

Different types of **LRP rules** define how to compute qij, and they are chosen depending on the type of layer or activation function:

**LRP rules** are formulas that define **how to compute the share of relevance** each input neuron should get when we’re distributing relevance backward from a neuron in the next layer.

Since different layers (or models) behave differently, there’s **no single way** to split up the relevance — so researchers developed several **rules** to handle different situations. Each rule defines how to compute a value called **qij,** which helps in deciding how important neuron iii is to neuron j's output.

| **Rule** | **Formula** | **Used for** |
| --- | --- | --- |
| **αβ-rule** | Uses a mix of positive and negative contributions | Deep networks |
| **w²-rule** | Uses squared weights | Bottom layers of Atari agents |
| **ε-rule** | Adds a small stabilizing term | Our purpose |

Each rule changes how relevance is spread backward.

**What is qij​?**

a **score or contribution** from input neuron **i** to output neuron **j**.

* It measures **how much neuron i influenced** neuron j’s activation during the **forward pass**.

**Conclusion**

Through this project, we developed a foundational understanding of LRP as a principled and practical method for deep neural network interpretation. We gained experience not only in using interpretability tools but also in thinking critically about model behavior and trustworthiness. This work has given us valuable insight into the growing field of explainable AI, which is becoming essential for responsible machine learning development.