Dependable AI (CSL7370)
Explaining neural network predictions using iNNvestigate

TEAM DESCRIPTION

Rishav Aich (B21AI029)
Pre-final Year (B.Tech. Artificial Intelligence & Data Science)

Tanish Pagaria (B21AI040)
Pre-final Year (B.Tech. Artificial Intelligence & Data Science)

IIT Jodhpur Undergraduates

OVERVIEW

The project aims to tackle a classification problem utilizing neural networks while employing iNNvestigate tools to provide insightful explanations for model predictions.

iNNvestigate

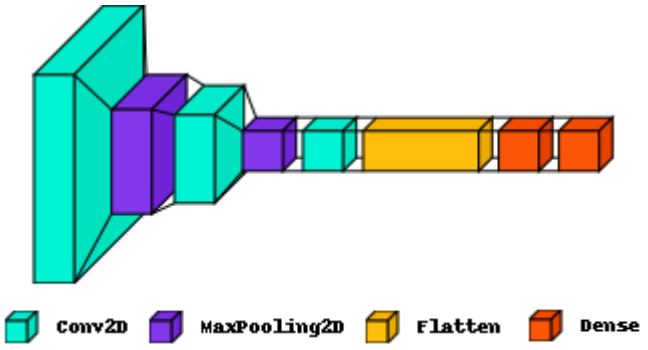
`iNNvestigate` is a Python library designed to investigate neural networks' predictions, providing a toolbox for analyzing neural networks. It is built on top of Keras and TensorFlow2.
It offers various methods for understanding how neural networks make their predictions, including gradient, SmoothGrad, DeConvNet, Guided BackProp, DeepTaylor, LRP, and Integrated Gradients.

DATASET: MNIST

The `MNIST` dataset is a collection of 60,000 small square 28×28 pixel grayscale images of handwritten single digits between 0 and 9, used for the task of classifying a given image of a handwritten digit into one of 10 classes representing integer values from 0 to 9.

MODEL ARCHITECTURE

The model was a Convolutional Neural Network (CNN) designed for image classification tasks, constructed using the Keras Sequential API.
It comprised three convolutional layers with ReLU activation functions, each followed by a max pooling layer to reduce spatial dimensions and computational complexity. After the convolutional layers, a Flatten layer transformed the 2D feature maps into a 1D vector for input into the dense layers. The architecture included two dense layers: the first with 64 neurons and ReLU activation for feature extraction, and the second with 10 neurons and softmax activation for multi-class classification. It was structured to process 28x28 grayscale images.



MODEL TRAINING

The model was compiled with the Adam optimizer, categorical cross entropy as the loss function, and accuracy as the metric for evaluation.
The model gave **99.19% accuracy** on the testing dataset for the MNIST digit image classification task.

iNNvestigate ANALYSIS

The `analyzer` class of the `iNNvestigate` library was employed for analyzing the model predictions. The class offered the following analysis methods, as shown below:

- **Gradient Methods**
  - **`Gradient`**
    The gradient is computed via the library's network reverting.

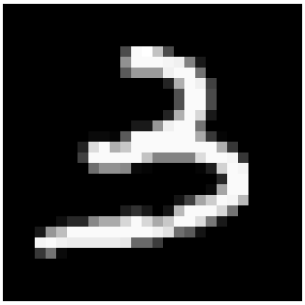
  - **`InputTimesGradient`**
    This method analyzes Input*Gradient.

  - **`Deconvnet`**
    The DeconvNet algorithm analyzes Convolutional Neural Networks (CNNs) by reversing their operations, enabling the visualization of feature maps and understanding the model's decision-making process.

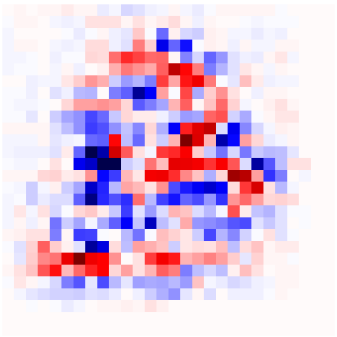
  - **`GuidedBackprop`**
    Guided Backpropagation is a gradient-based visualization technique.
    Given an input image and a pre-trained network, guided backpropagation tells which pixels in the input image matter for the correct classification.

  - **`IntegratedGradients`**
    Integrated Gradients is a technique for attributing a classification model's prediction to its input features by computing the integral of the gradients of the prediction output with regard to the input features.

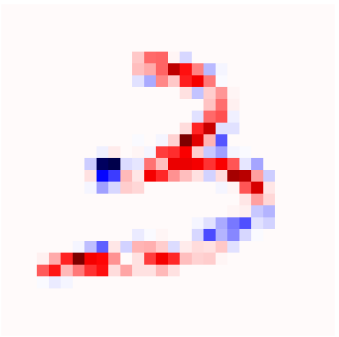
  - **`SmoothGrad`**
    SmoothGrad is an algorithm that improves the interpretability of Convolutional Neural Networks (CNNs) by generating multiple saliency maps from slightly different noisy versions of an input image and averaging them to produce a smoothed, noise-reduced saliency map.



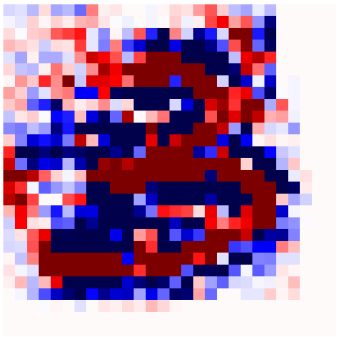
Original Image



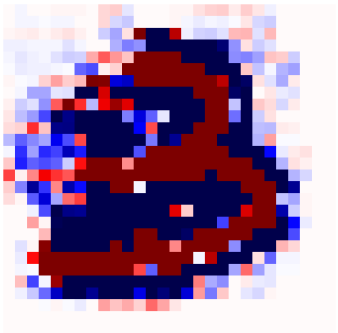
`Gradient`



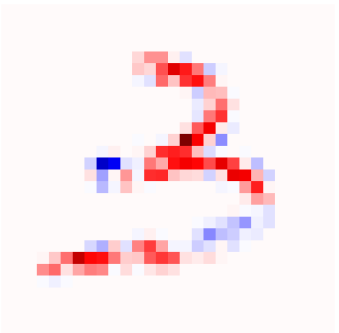
`InputTimesGradient`



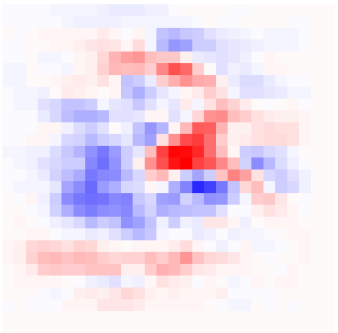
`Deconvnet`



`GuidedBackprop`

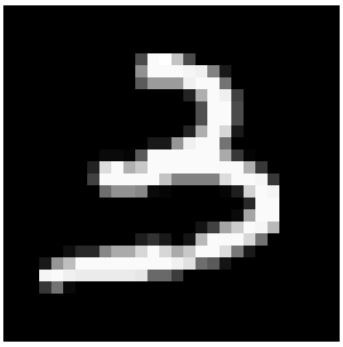


`IntegratedGradients`

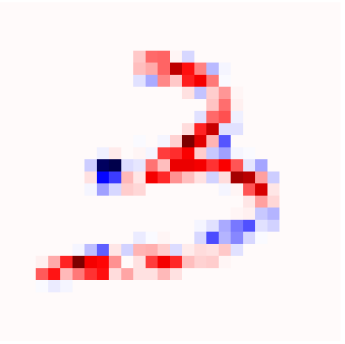


`SmoothGrad`

- **Layer-Wise Relevance Propagation (LRP)**
  It is a technique for explaining the decisions of deep neural networks by propagating relevance values from the output layer to the input layer, providing insights into the importance of input features in the prediction process.

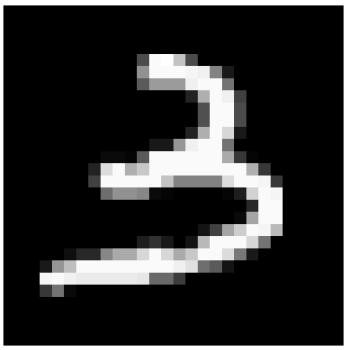


Original Image



`LRP`

- **Deep Taylor Decomposition (DTD)**
  It is a method for explaining deep neural networks by decomposing the network's output into contributions from its input features, using the Taylor series expansion to approximate the non-linear functions in the network.

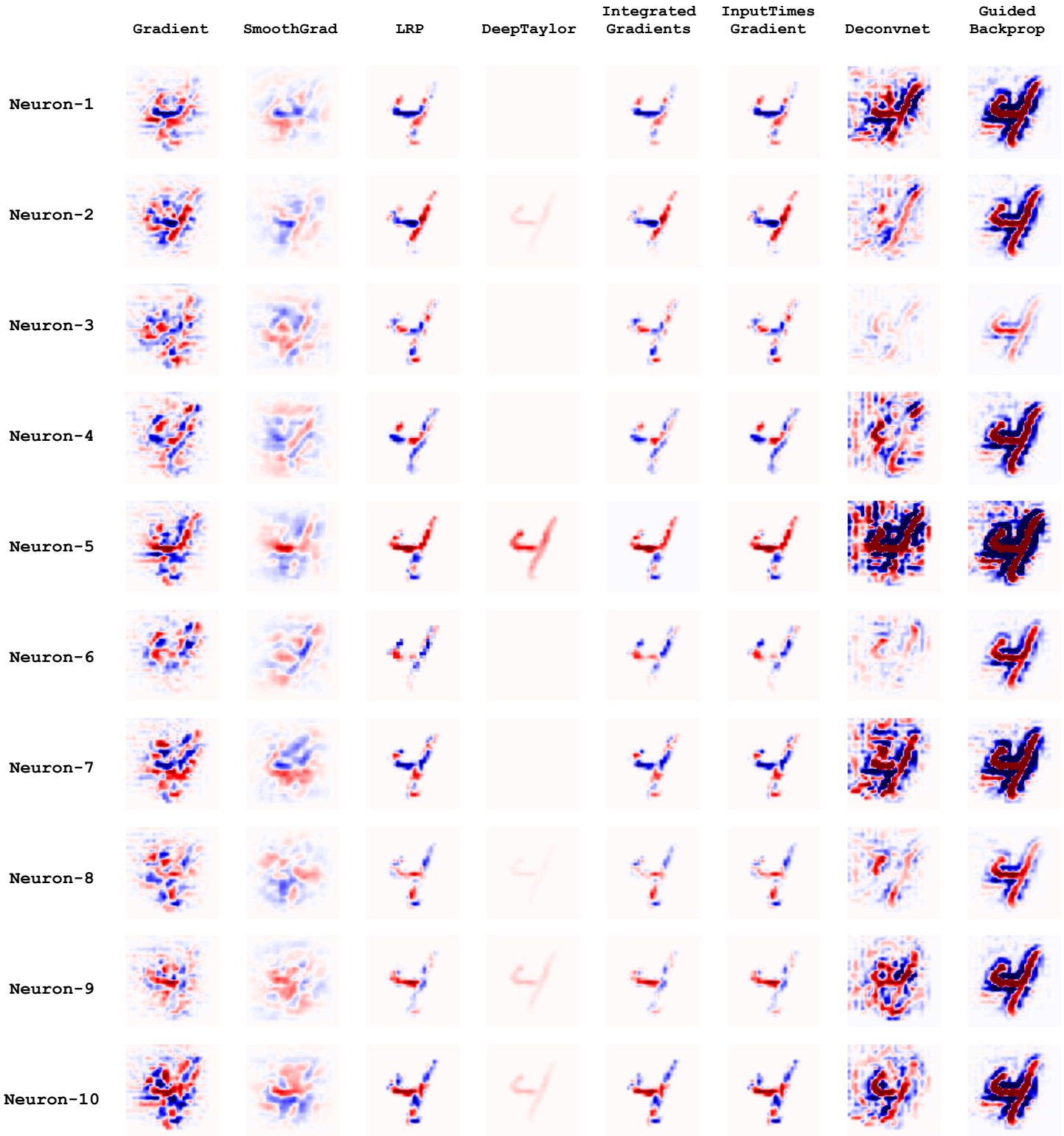


Original Image

`DeepTaylor`

iNNvestigate NEURON SELECTION

The saliency maps computed for output neurons for each analyzer are shown below:



In the above visualizations, the positive and negative attributions of all the neurons are shown. These can be used to analyze the prediction at each neuron.

REFERENCES

iNNvestigate Python Library

https://github.com/albermax/innvestigate
https://pypi.org/project/innvestigate/1.0.8/
https://innvestigate.readthedocs.io/en/latest/

iNNvestigate GitHub Repository

https://github.com/albermax/innvestigate/blob/master/examples/introduction.ipynb
https://github.com/albermax/innvestigate/blob/master/examples/mnist_compare_methods.ipynb
https://github.com/albermax/innvestigate/blob/master/examples/mnist_neuron_selection.ipynb