NN Interpretability

Aleksandar Aleksandrov
Hans Hao-Hsun Hsu

Outline

- Project Introduction
- Model-Based Interpretability Methods
- Decision-Based Interpretability Methods
- Uncertainty
- Interpretability Overview
- Demo

Interpretability Introduction

Introduction

- What is NN interpretability?

Black-Box Models (Deep Neural Network)

Predicted as “Dog”

Why is it predicted as “Dog” ?

Introduction

- What types of NN interpretability methods are there?
    - **Model-based methods** (e.g. Activation Maximization) try to explain what does the concepts learned from a model look like. (**How does a “dog” typically look like?**)

    - **Decision-based methods** (e.g. Layerwise Relevance Propagation) try to explain why did the model assign a certain concept to a premeditated input. (**Why is this example classified as “dog”?**)

Model-based Methods

Activation Maximization (AM)

- AM is a **model-based approach** that searches for an **input pattern which elicits a maximum model response** for a class of interest.

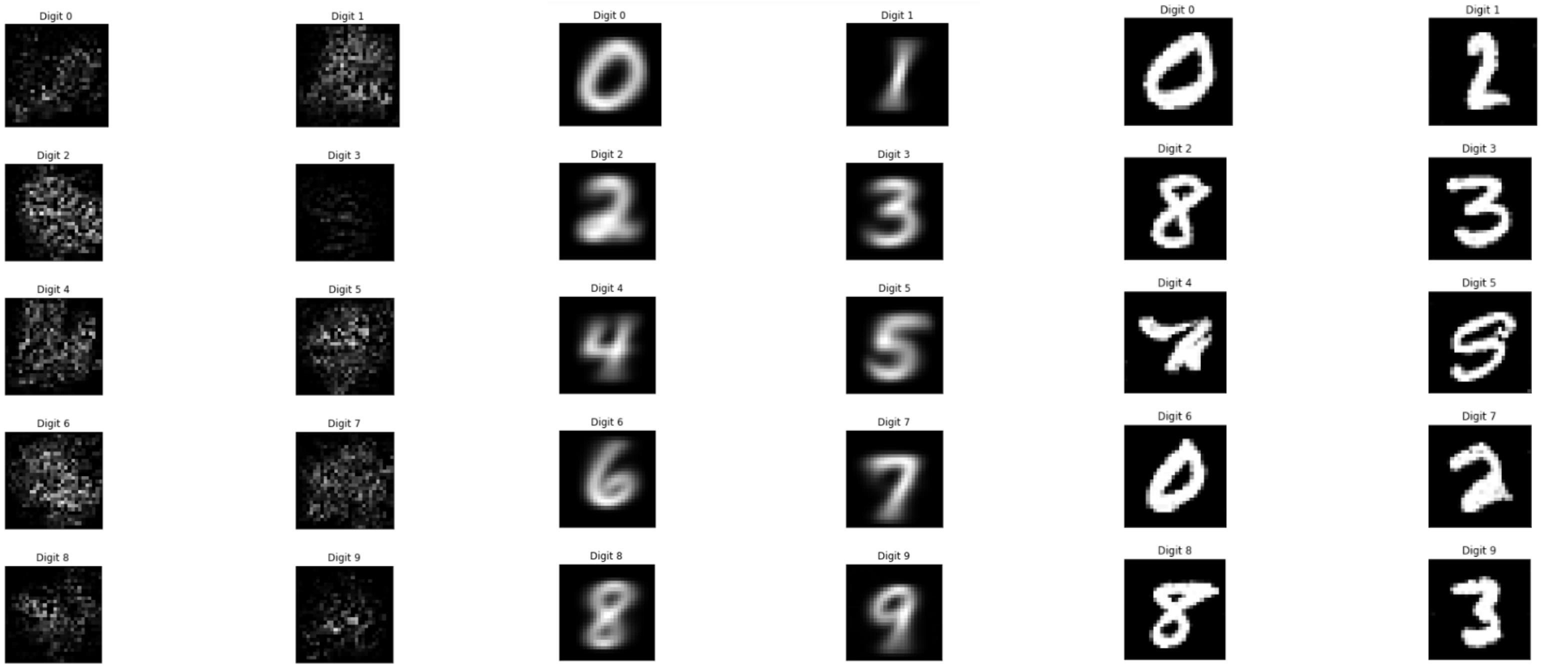
- **General AM**:
    - Different starting points
        - Random image from dataset
        - Random noise
        - Mean class image

$$\max_{\boldsymbol{x}} \quad \log p(\omega_c|\boldsymbol{x}) - \lambda\|\boldsymbol{x}\|^2.$$

- **AM in Codespace**
    - Different generative models
        - Simple GAN
        - Pretrained DCGAN

$$\max_{z\in Z} \quad \log p(\omega_c \mid g(z)) - \lambda\|z\|^2,$$

AM Results



Left: General AM - Random Noise; **Middle:** General AM- Mean Image; **Right:** AM in Codespace with DCGAN

DeepDream

- Set a **whole layer** as our **Activation Maximization** objective.

$$\max_{\boldsymbol{x}} \quad \log p(\omega_c|\boldsymbol{x}) - \lambda\|\boldsymbol{x}\|^2.$$

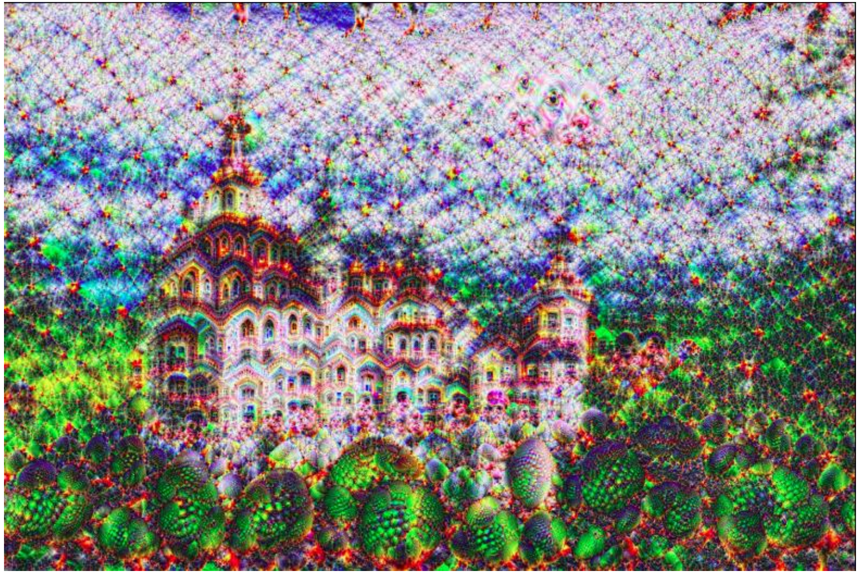
Weights of a certain layer

- Starting point

- Pretrained VGG-16

DeepDream

- Techniques for feature visualization (1.) **L2 Regularization** (2.) **Add Gaussian Noise** (3.) **Random Roll** (4.)**Multiple Octaves** (5.) **Split Image into Tiles**



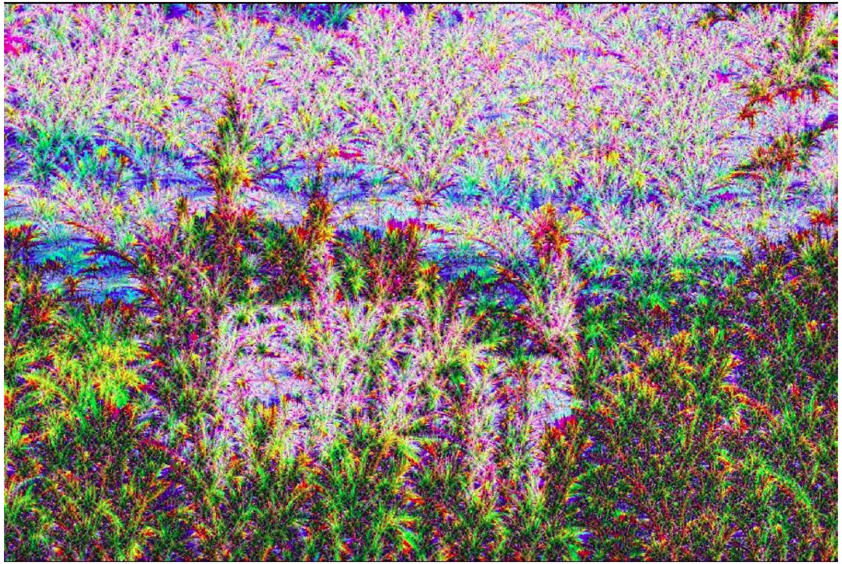
25th layer

27th layer

DeepDream

- Set a **channel** as our **Activation Maximization** objective.

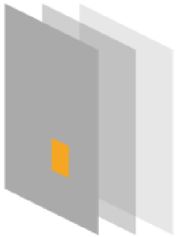
25th layer channel No.30



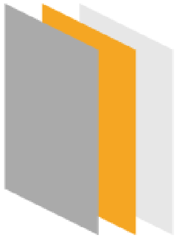
25th layer channel No.150

Model-based Methods Conclusion

Optimization Objectives

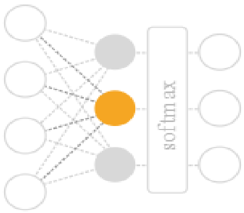


Neuron

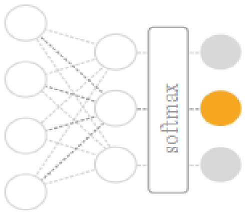


Channel

Layer
(DeepDream)



Class Logits
(General AM)



Class Probability

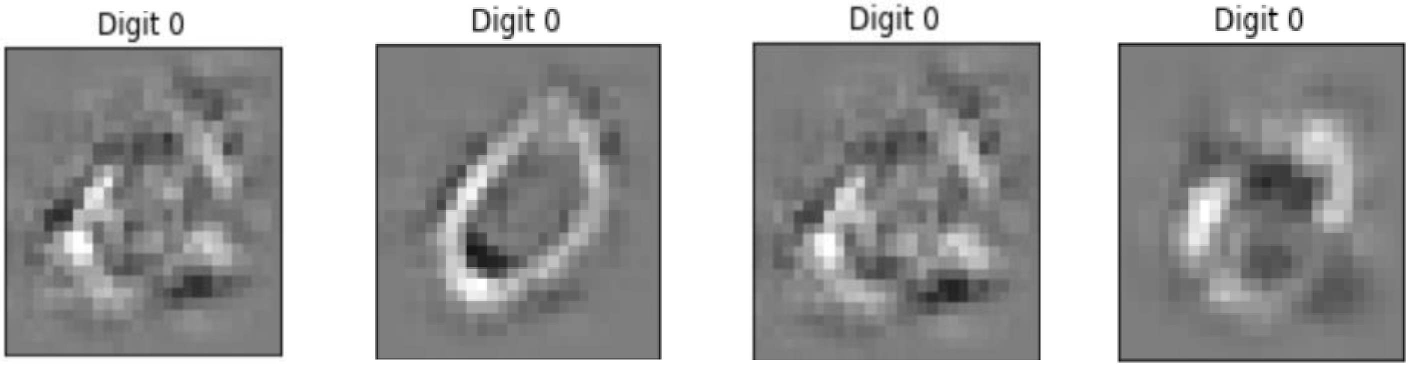
Decision-based Methods

Backprogagation (Gradient based)

- **Vanilla backpropagation** (Saliency map) compute attribution based on **gradients at input**

- Drawbacks of vanilla backpropagation:
    - Low Visual Quality -> **Guided Backpropagation** (Positive Gradients)
    - Break Sensitivity -> **Integrated Gradients** (Path Method)
    - Too Noisy -> **SmoothGrad** (Adding Gaussian Noise)

Backprogagation (Gradient based)



Vanilla Backprop | Guided Backprop | Integrated Gradients | SmoothGrad

- Which pixels **influence** this prediction **the most** ? (If you use gradient based)

$$\neq$$

- Which pixels cause the prediction?

$$f(x) \approx f(\tilde{x}) + \boxed{\frac{\partial f}{\partial x}(\tilde{x})} \cdot (x - \tilde{x})$$

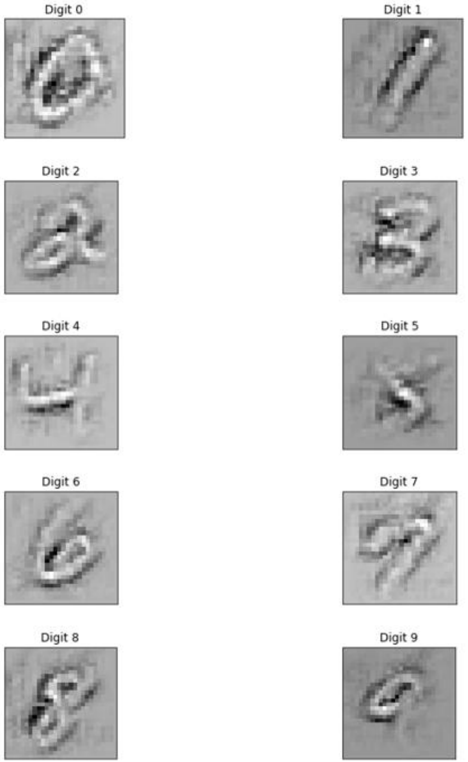
Deconvolution

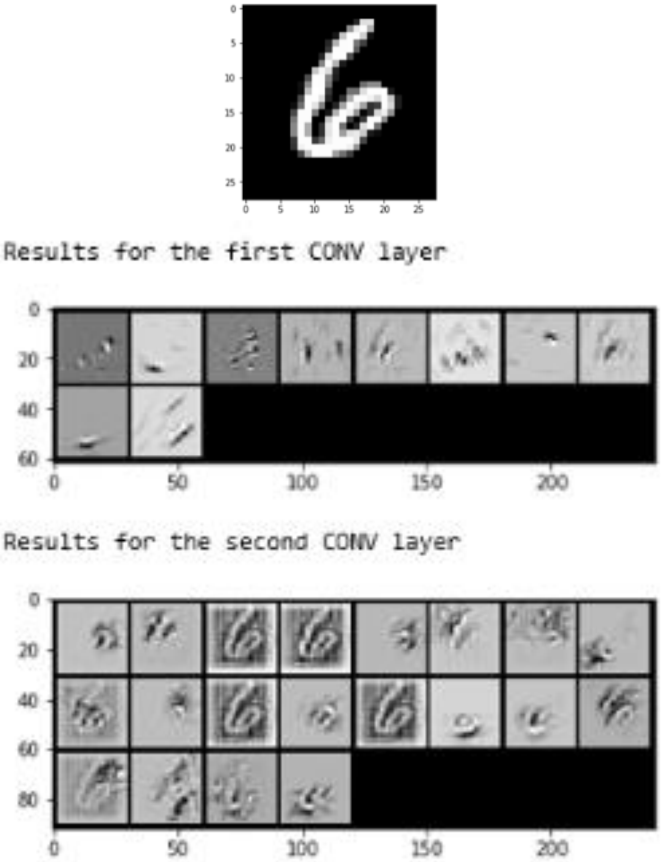
- Deconvolution is a **decision-based approach** for mapping feature activities back to the input pixel space.

- Deconvolution has the **reversed structure** of a concrete CNN model and **reuses the initially learned weights**.

- **Applications**
    - DeConvNet input reconstruction
    - Single Filter Projection in input space
    - LRP

DeConvNet Results





Left: Full DeConvNet; **Right:** Single Feature Projection for each filter in each layer

Class Activation Maps (CAM)

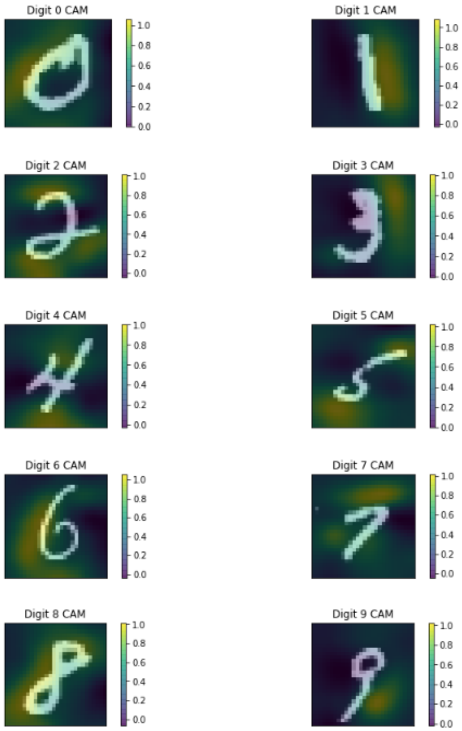
- CAM is a **decision-based approach** that highlights the parts of the input which had the highest influence to the model's decision.

- Variations
  - CAM
    - Limited to CNN layers ending with a single AVGPOOL layer followed by a Linear layer.
    - Huge dependency on model architecture
  - Grad-CAM
    - CAM generalization by utilising the gradient
    - Support for all CNN architectures
    - Can be applied to any CONV layer in a CNN

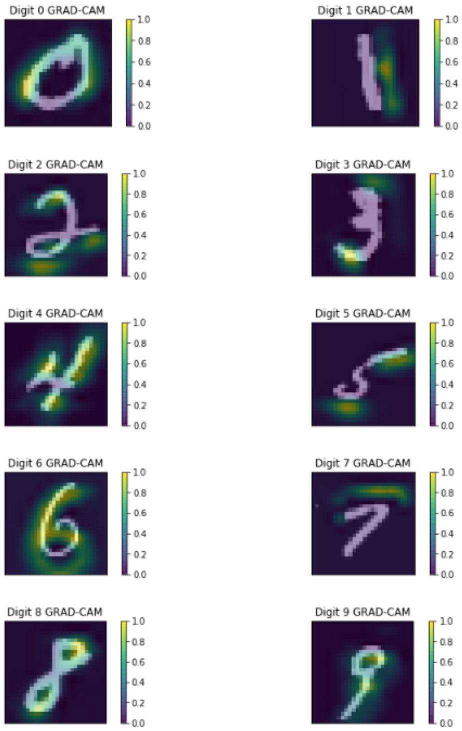
$$M_c(x, y) = \sum_k w_k^c f_k(x, y).$$

$$L^c_{\text{Grad-CAM}} = ReLU \underbrace{\left(\sum_k \alpha_k^c A^k \right)}_{\text{linear combination}}$$

CAM & Grad-CAM Results



CAM Results



Grad-CAM Results: CONV 2

Deep Learning Important FeaTures (DeepLIFT)

- DeepLIFT is a **decision-based approach** which aims to dissect the output by **backpropagating the contributions** of every layer back to the input. DeepLIFT tries to **explain the decision in the context of a predefined reference input**.

- Different Rules
  - ***Linear*** - Linear & CONV layers
  - ***Rescale*** - ReLU activations
  - ***RevealCancel*** - ReLU activations
  - **Combinations**

$$
\begin{aligned}
C_{\Delta x_i^+ \Delta y^+} &= 1\{w_i \Delta x_i > 0\} w_i \Delta x_i^+ \\
C_{\Delta x_i^- \Delta y^+} &= 1\{w_i \Delta x_i > 0\} w_i \Delta x_i^- \\
C_{\Delta x_i^+ \Delta y^-} &= 1\{w_i \Delta x_i < 0\} w_i \Delta x_i^+ \\
C_{\Delta x_i^- \Delta y^-} &= 1\{w_i \Delta x_i < 0\} w_i \Delta x_i^-
\end{aligned}
$$

Linear Rule: Contributions

$$
\begin{aligned}
\Delta y^+ &= \frac{\Delta y}{\Delta x} \Delta x^+ = C_{\Delta x^+ \Delta y^+} \\
\Delta y^- &= \frac{\Delta y}{\Delta x} \Delta x^- = C_{\Delta x^- \Delta y^-}
\end{aligned}
$$

Rescale Rule: Contributions

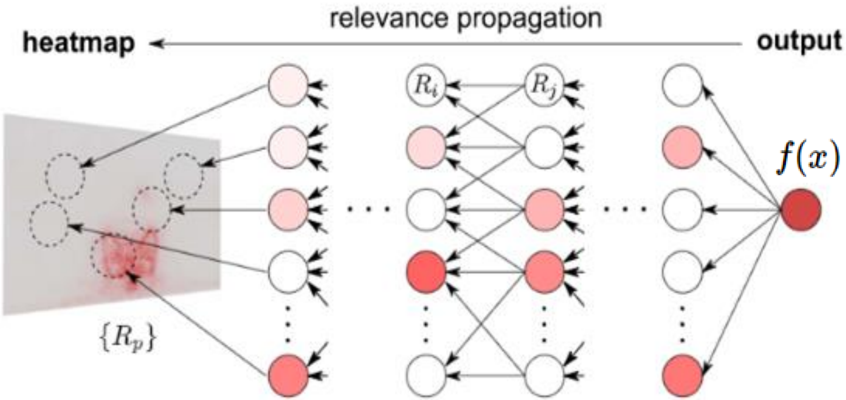
DeepLIFT Results

Left: Vanilla Backpropagation; **Middle:** LinearRescale; **Right:** LinearRevealCancel

Layer-wise Relevance Propagation

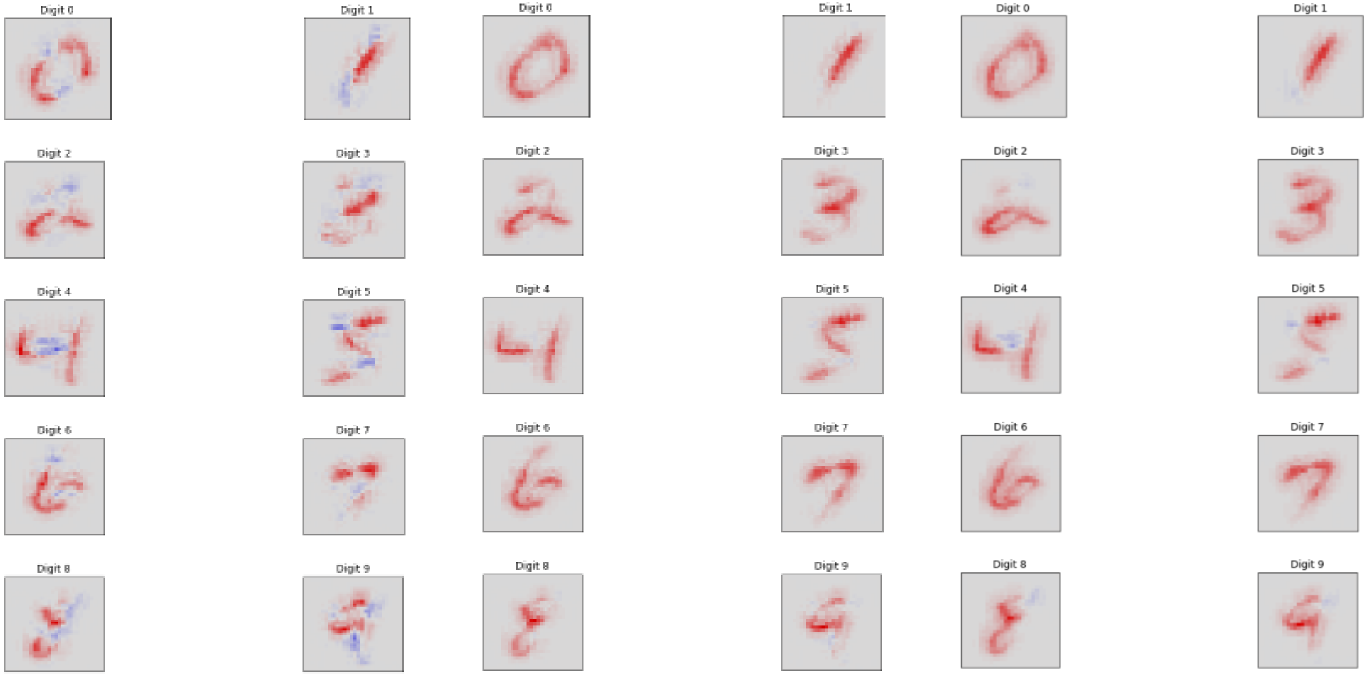
- Layer-wise Relevance Propagation(LRP) backpropagates **contribution** of every pixel **from a predicted class logit** using a set of LRP rules. During the propagation each layer preserve **same quantity of contribution.**

- Variations:
  - LRP-0 Rule $$R_j = \sum_k \frac{a_j w_{jk}}{\sum_{0,j} a_j w_{jk}} R_k$$
  - LRP-ε Rule $$R_j = \sum_k \frac{a_j w_{jk}}{\epsilon + \sum_{0,j} a_j w_{jk}} R_k$$
  - LRP-γ Rule $$R_j = \sum_k \frac{a_j \cdot (w_{jk} + \gamma w_{jk}^+)}{\sum_{0,j} a_j \cdot (w_{jk} + \gamma w_{jk}^+)} R_k$$
  - LRP-αβ $$R_j = \sum_k \left(\alpha \frac{a_j w_{jk}^+}{\sum_j a_j w_{jk}^+} - \beta \frac{a_j w_{jk}^-}{\sum_j a_j w_{jk}^-} \right) R_k$$



LRP Results

- Which pixels cause the prediction of digit X?



LRP-0 LRP-γ Rule Composite LRP

Red: Positive Contribution

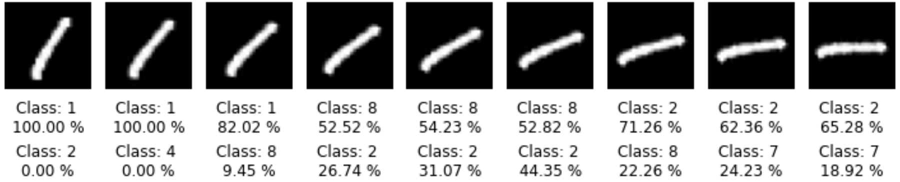
Blue: Negative Contribution

Uncertainty

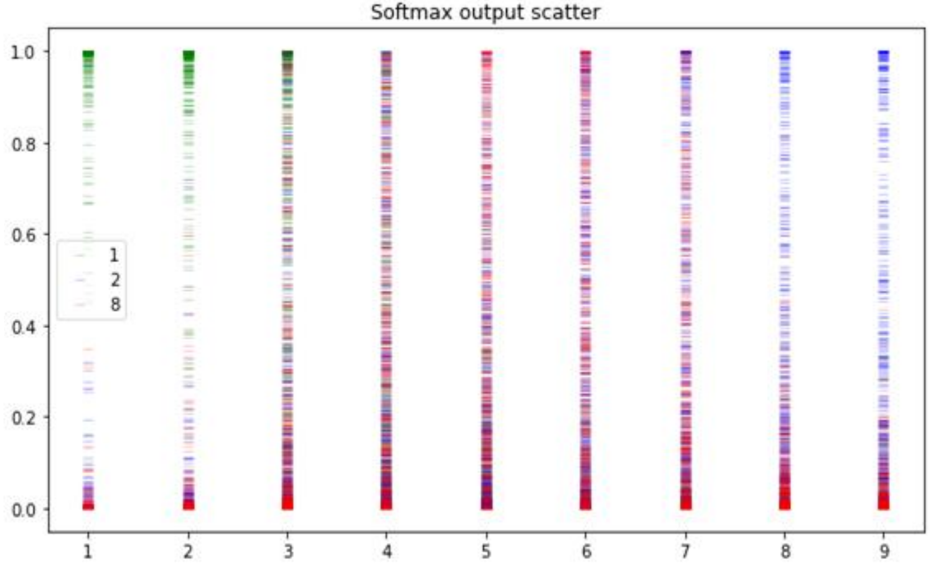
Monte Carlo Dropout

- Dropout as bayesian approximation. Same training as standard approach but enable dropout at test time.
- Overlapping colors mean high uncertainty
- **Softmax ≠ Confidence**

Prediction **without** dropout



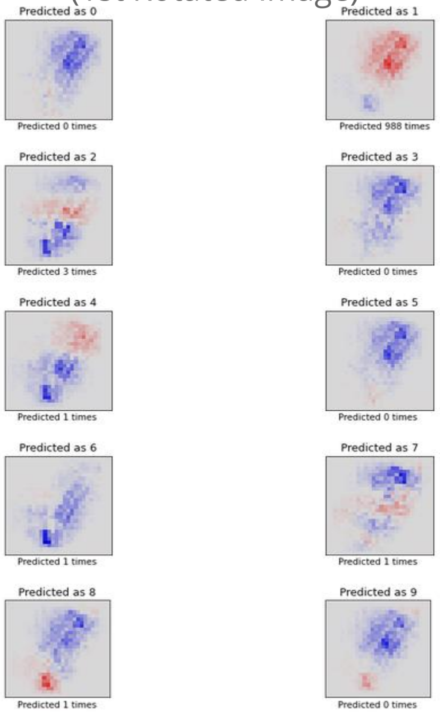
Prediction **with** dropout



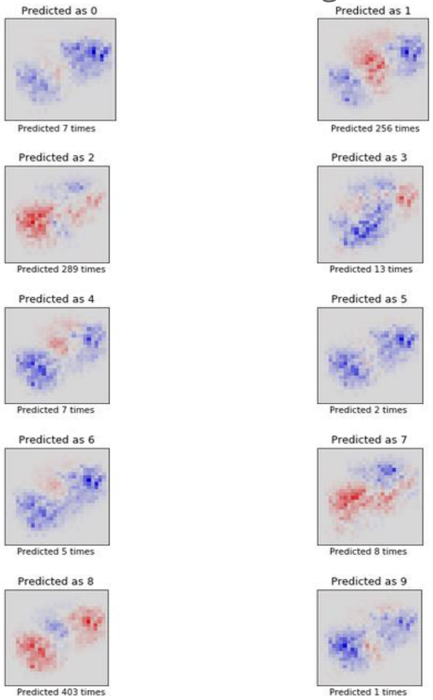
Interpret MC Dropout with LRP

- Observe 1000 times predictions for **each class**, backpropagate with LRP starting from the **assigned class** and sum the heatmaps of those 1000 observations.

Prediction under **low uncertainty** (1st Rotated Image)



Prediction under **high uncertainty** (3rd Rotated Image)



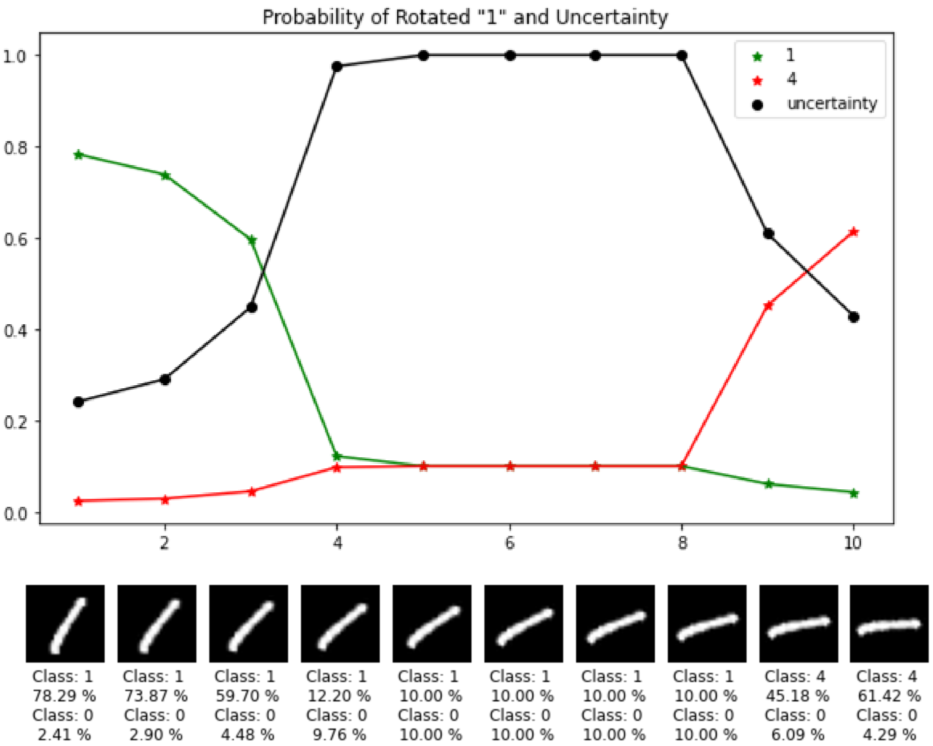
Evidential Deep Learning

- Replace the softmax layer with a **ReLU** to get non-negative **evidence.** Take “evidence+1” as alpha in **Dirichlet Distribution**(prior)

$$D(\mathbf{p}|\boldsymbol{\alpha}) = \begin{cases} \frac{1}{B(\boldsymbol{\alpha})} \prod_{i=1}^{K} p_i^{\alpha_i - 1} & \text{for } \mathbf{p} \in \mathcal{S}_K, \\ 0 & \text{otherwise,} \end{cases}$$

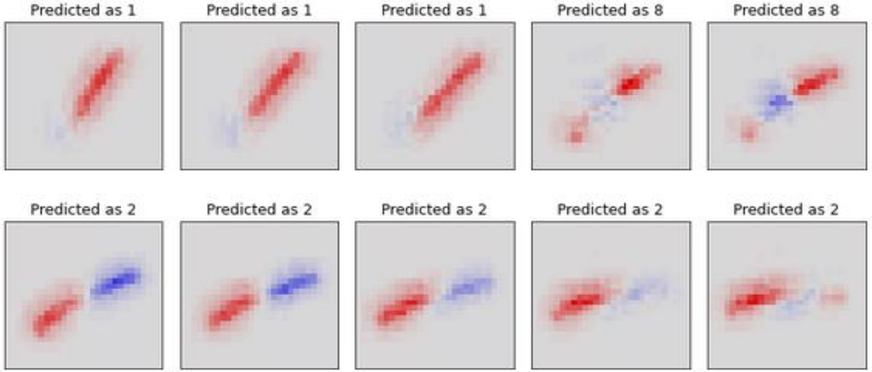
- Introduce uncertainty during our training

Prediction of evidential deep learning model

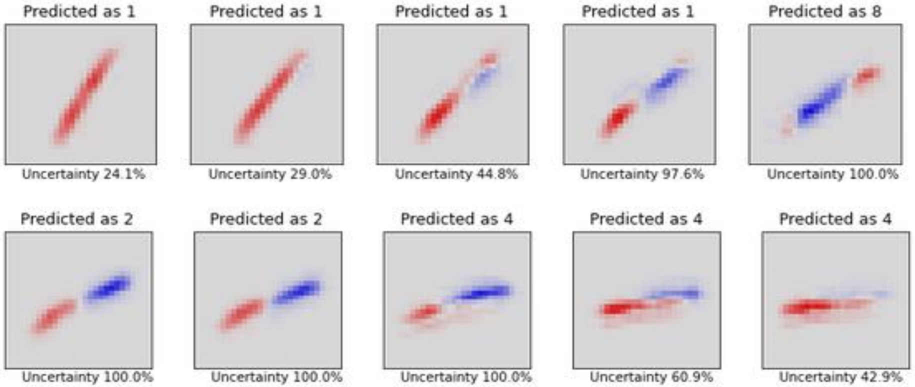


Base Model vs. Evidential Deep Learning Model

- Interpret with LRP



Model **without** uncertainty awareness



Model **with** uncertainty awareness

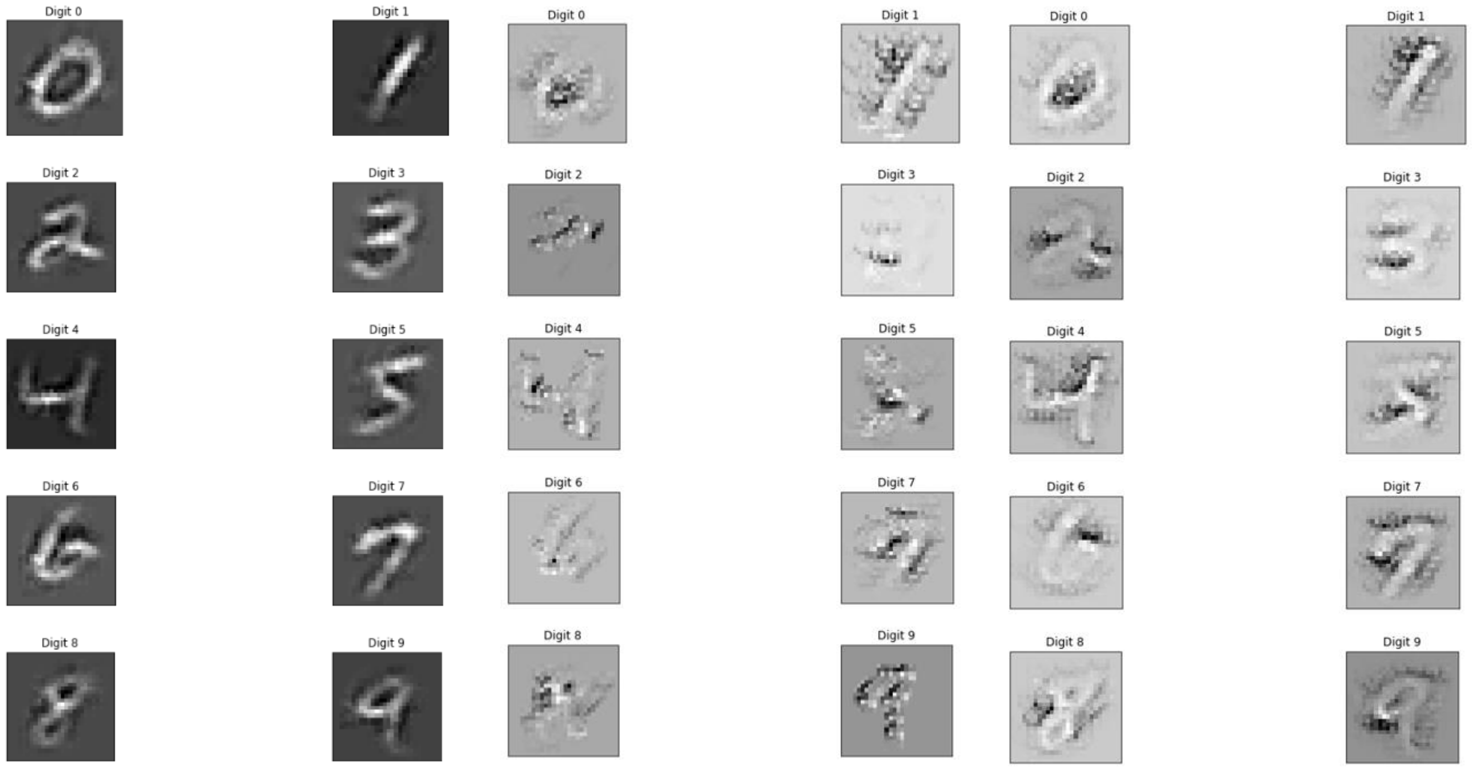
Uncertain DeepLIFT

- Motivation: **How does DeepLIFT behave under uncertainty?**

- Experiments
    - **Experiment #1:** Standard CNN model vs. Dropout CNN Model (single- and multi-pass)
    - **Experiment #2:** Input with random noise
    - **Experiment #3:** Temperature scaling
        - Standard CNN model & MNIST
        - Pretrained AlexNet & ImageNet (1k images validation subset)

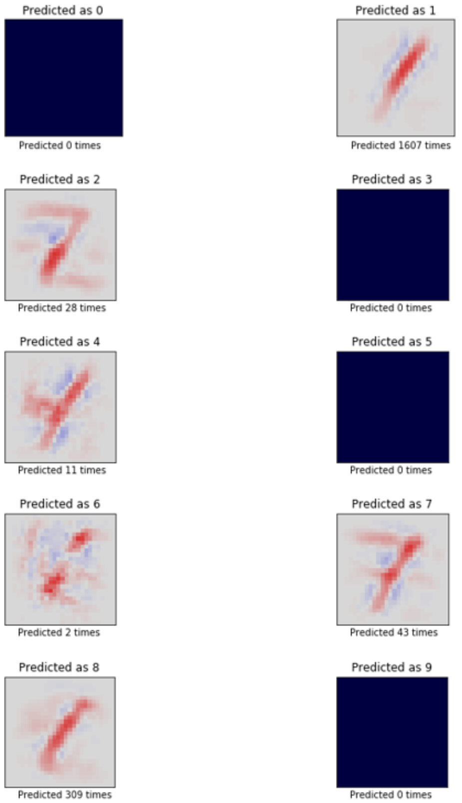
$$\hat{q}_i = \max_k \ \sigma_{\mathrm{SM}}(\mathbf{z}_i / T)^{(k)}.$$

Experiment #1: CNN vs Dropout-CNN

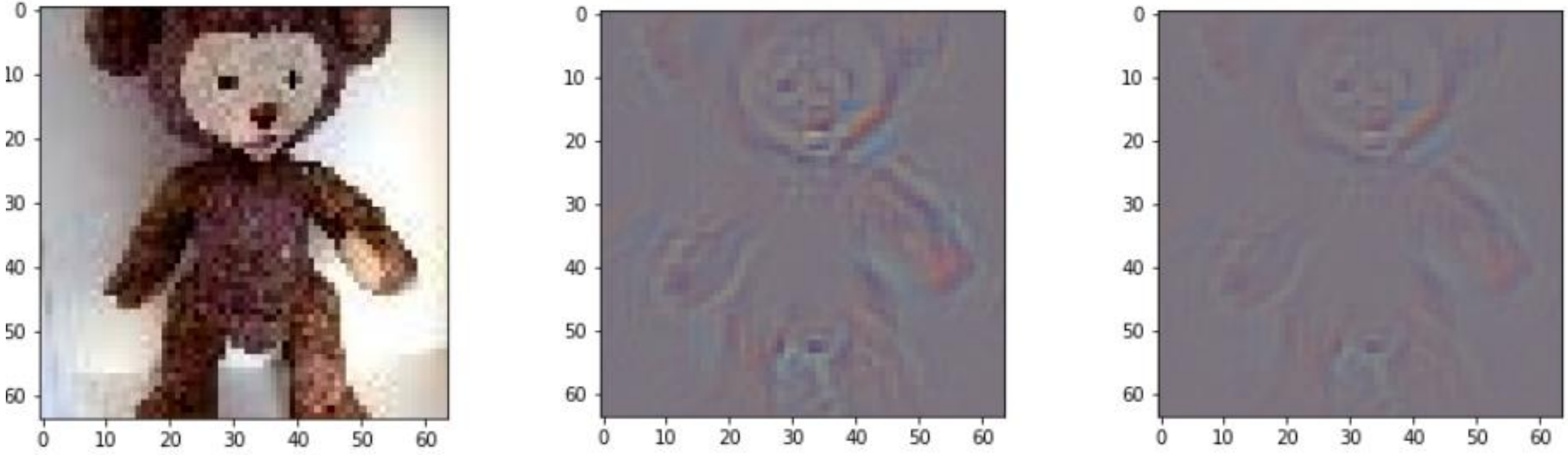


Left: Standard CNN; **Middle:** Single-pass Dropout-CNN; **Right:** Multi-pass Dropout-CNN

Experiment #2: Input with Random Noise



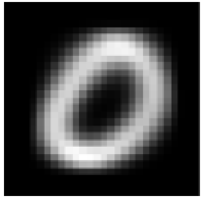
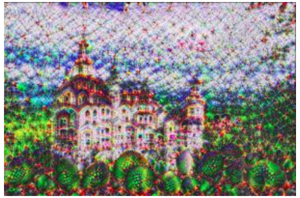
Experiment #3: Temperature Scaling (ImageNet)



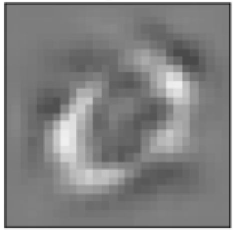
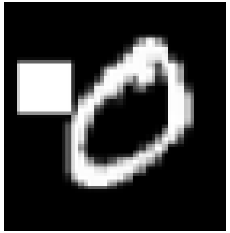
Left: Input Image; **Middle:** DeepLIFT for AlexNet; **Right:** DeepLIFT for TS-AlexNet

Summary

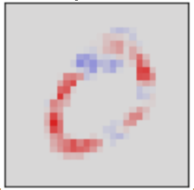
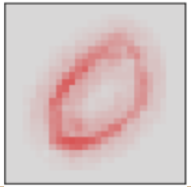
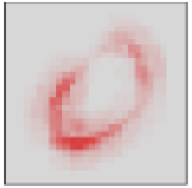
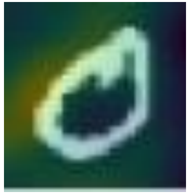
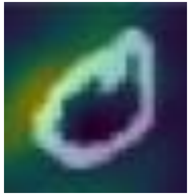
Interpretability Overview I

	AM 	DeepDream 	Deconvolution 	DeepLIFT 
Type	Model	Model	Decision	Decision
Use case	Find the prototype of each class	Find the prototype of each layer	Reconstruct output from input	Assign contributions to input pixels
Complexity	**high**	**middle**	**middle**	**middle**
Support	No restrictions	Deep NN	Need MAXPOOL	No restrictions
Drawback	Unstable, huge overhead	Several visualization techniques	Need for second NN	Lots of rules

Interpretability Overview II

	Saliency Map 	Guided Backprop 	Integrated 	SmoothGrad 	Occlusion sensitivity 
Type	Decision	Decision	Decision	Decision	Decision
Use case	Changing which pixel will change the decision the most	Changing which pixel will change the decision the most	Changing which pixel will change the decision the most	Changing which pixel will change the decision the most	Showcase parts of the input which upon occlusion lead to output change
Complexity	**low**	**low**	**low**	**low**	**low**
Support	No restrictions	Require ReLU	No restrictions	No restrictions	All CNNs
Drawback	Noisy, Shattered gradients	Shattered gradients Easy to fail with uniform background	Shattered gradients	Shattered gradients	Limited expression power

Interpretability Overview III

	Simple Taylor Decomposition 	Deep Taylor Decomposition 	LRP 	CAM 	Grad-CAM 
Type	Decision	Decision	Decision	Decision	Decision
Use case	Show pixel direct contributions	Show pixel direct contributions	Show pixel direct contributions	Highlight important parts of the input	Highlight important parts of the input
Complexity	**middle**	**middle**	**middle**	**low**	**low**
Support	No restrictions	No restrictions	No restrictions	Subset of CNN	All CNNs
Drawback	Hard to find root point	Doesn’t show negative contribution	Hard to choose between rules	Strong support limitations	Interpolation issues

Demo Time

Sources

- Montavon, Grégoire, Wojciech Samek, and Klaus-Robert Müller. “Methods for Interpreting and Understanding Deep Neural Networks.” Digital Signal Processing 73 (2018): 1–15. Crossref. Web.
- Matthew D. Zeiler and Rob Fergus (2013). Visualizing and Understanding Convolutional NetworksCoRR, abs/1311.2901.
- Gregoire Montavon, et al., “Layer-Wise Relevance Propagation: An Overview"
- Olah, et al., "Feature Visualization", Distill, 2017.
- Olah, et al., "The Building Blocks of Interpretability", Distill, 2018.
- Karen Simonyan, Andrea Vedaldi, & Andrew Zisserman. (2013). Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps.
- Jost Tobias Springenberg, et al., “Striving for Simplicity: The All Convolutional Net”
- Mukund Sundararajan, et al., “Axiomatic Attribution for Deep Networks”
- Daniel Smilkov, et al., “SmoothGrad: removing noise by adding noise”
- Alexander Mordvintsev, et al., “Inceptionism: Going Deeper into Neural Networks”
- Yarin Gal, et al., “Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning”
- Chuan Guo, et al., “On Calibration of Modern Neural Networks”
- Murat Sensoy, et al., “Evidential Deep Learning to Quantify Classification Uncertainty"
- Avanti Shrikumar, et al., “Learning Important Features Through Propagating Activation Differences.”

Backup Slides

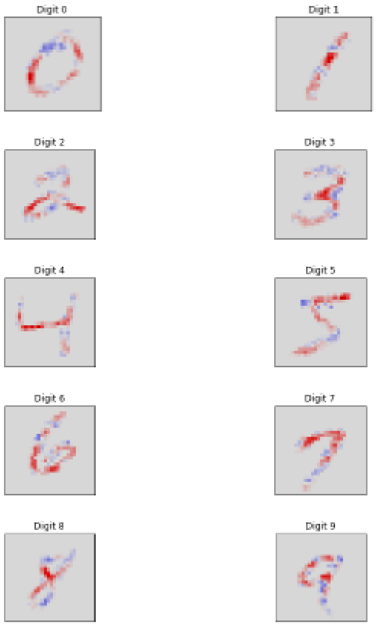
Taylor Decomposition

- Apply Taylor Expansion to the entire network **without** considering the structure of our model. -> Simple Taylor Decomposition

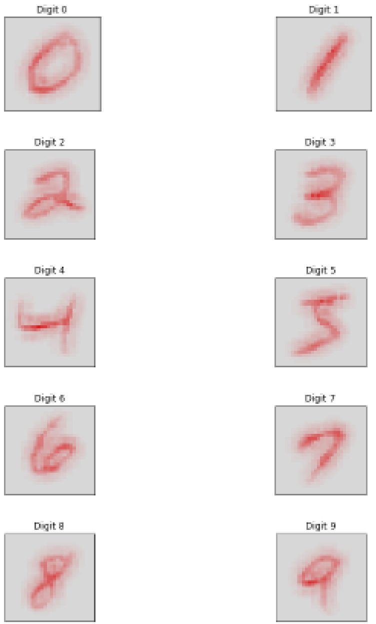
- Apply Taylor Expansion to a **layer-wise manner** by considering relevance score **layer by layer**. -> Deep Taylor Decomposition

Taylor Decomposition Results

- Red pixels: positive contribution; Blue pixels: negative contribution



Simple Taylor Decomposition



Deep Taylor Decomposition

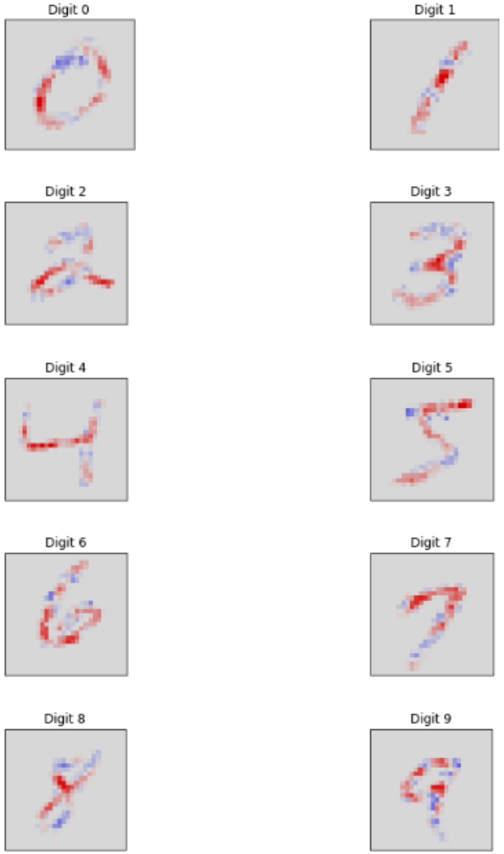
Taylor Decomposition

$$f(x) \approx f(\tilde{x}) + \frac{\partial f}{\partial x}(\tilde{x}) \cdot (x - \tilde{x})$$

- We set $\tilde{x}$ on the decision boundary of NN -> $f(\tilde{x}) = 0$
- ReLU is piecewise linear and satisfy $f(tx) = tf(x)$
  -> $\tilde{x} = \lim \varepsilon x$
  -> $f(x) = \sum_{i=1}^{V} \underbrace{\frac{\partial f}{\partial x_i} \cdot x_i}$

Relevance Score $R(x_i)$

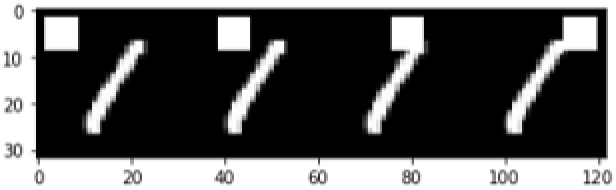
- Which pixels have a **positive/negative** contribution



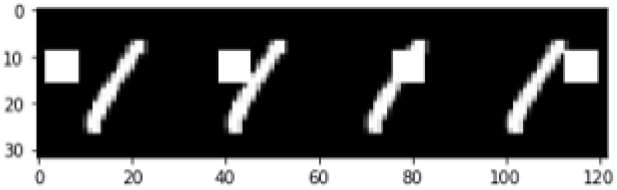
Occlusion Sensitivity

- Occlusion Sensitivity is a **decision-based approach** in which parts of the input are **deliberately obstructed to mislead** the decision of the model
- **Example:**

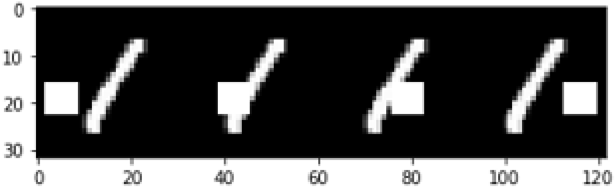
['100.0%', '20.94%', '98.26%', '100.0%']



['100.0%', '0.0%', '99.44%', '98.45%']



['100.0%', '97.84%', '0.07%', '100.0%']



['100.0%', '100.0%', '100.0%', '100.0%']