NN Interpretability

Aleksandar Aleksandrov
Hans Hao-Hsun Hsu

Outline

- Phase I Recap
- Phase II
  - Activation Maximization in Codespace and with an Expert
  - Gradient-based methods
  - Class Activation Maps (CAM) and Grad-CAM
- Interpretability Methods Overview
- Phase III Plans

Introduction

- What is NN interpretability?
    - NN interpretability refers to the process of **mapping of abstract concepts in a human-understandable domain**. A collection of features in the human-interpretable domain allows us to **provide possible explanations for the decisions of a model**.

- 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”?)

Phase I Recap

- Model-based Methods
  - General Activation Maximization

- Decision-based Methods
  - Deconvolutional Network
  - Occlusion Sensitivity
  - Saliency Maps
  - Layer-wise Relevance Propagation

Activation Maximization (AM)

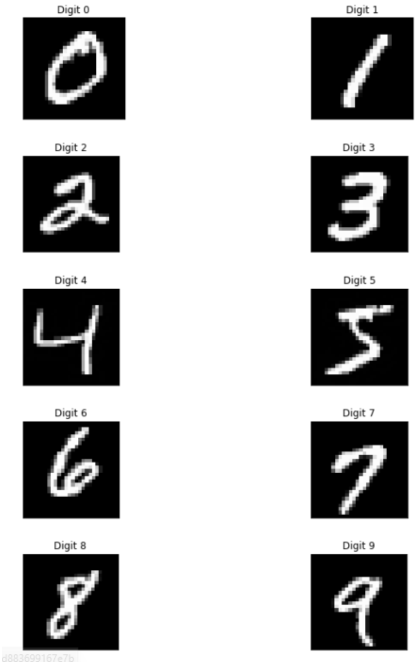
- AM is a **model-based approach** that searches for an **input pattern which elicits a maximum model response** for a class of interest.

- Variations:
    - General AM $\max_{\boldsymbol{x}} \ \log p(\omega_c|\boldsymbol{x}) - \lambda\|\boldsymbol{x}\|^2.$

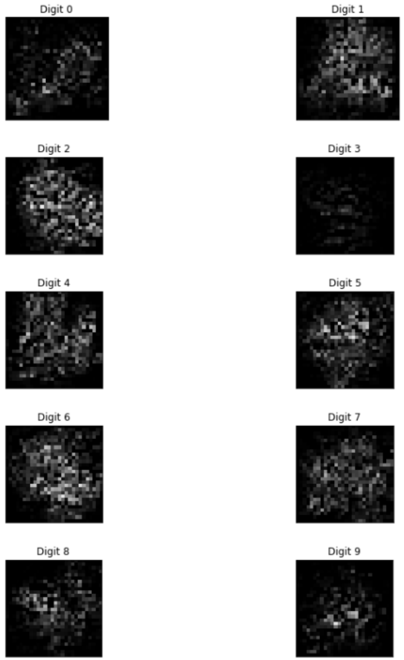
    - **AM with an Expert** $\max_{\boldsymbol{x}} \ \log p(\omega_c|\boldsymbol{x}) + \log p(\boldsymbol{x}).$

    - **AM in Codespace** $\max_{\boldsymbol{z} \in \mathcal{Z}} \ \log p(\omega_c \mid g(\boldsymbol{z})) - \lambda\|\boldsymbol{z}\|^2,$

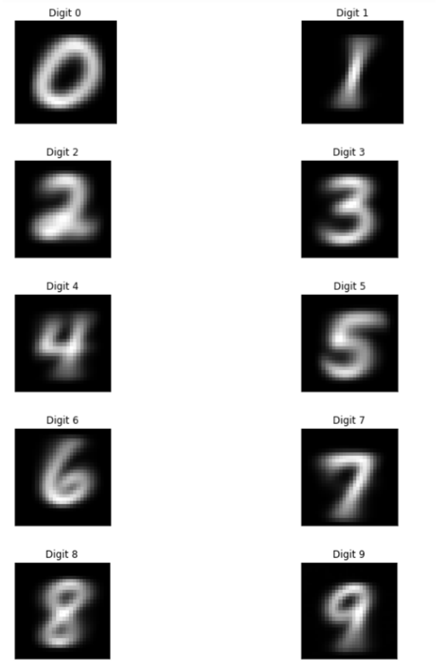
General AM - Update



General AM random image



General AM random noise



General AM Mean image

AM in Codespace

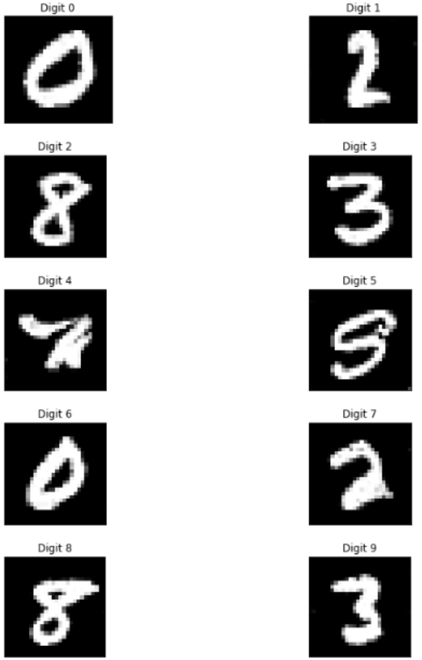
- AM in Codespace is a **model-based approach** which extends the general AM by introducing a **generative model for the generation of images**.

- We **sample in the latent space** and run the sample through the generative model before we **optimize for the chosen class.**

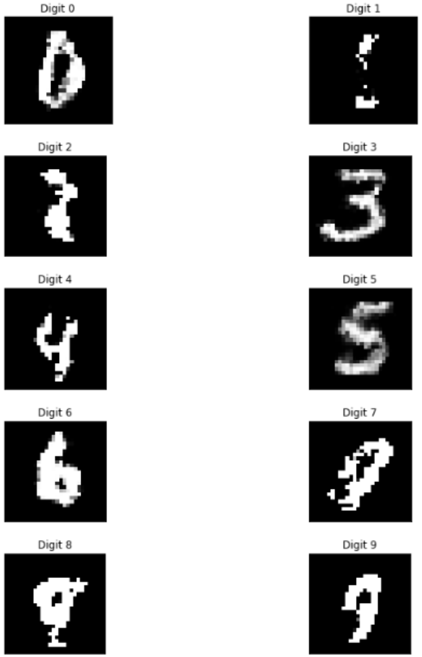
- We have done experiments with both pretrained models and GAN models that we .

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

AM in Codespace - Results & Comparison



AM in Codespace
Pretrained DCGAN



AM in Codespace
Trained simple GAN

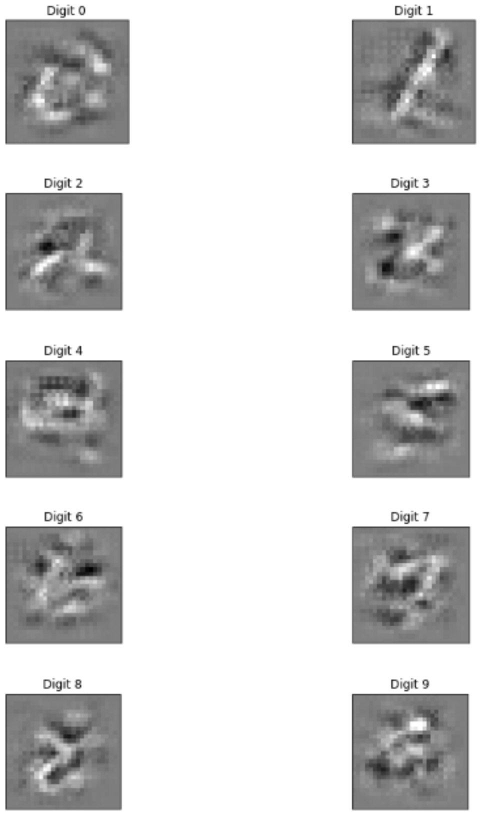
Recap - Saliency Map

- Also called Vanilla Backpropagation
- First-order Taylor expansion

$$S_c(I) \approx S_c(I_0) + \left.\frac{\partial S_c}{\partial I}\right|_{I_0} \underbrace{(I - I_0)}_{\delta}$$

$$w = \left.\frac{\partial S_c}{\partial I}\right|_{I_0}$$

- Which pixels need to be changed the least to affect the class score the most
- Problem: too noisy, breaks sensitivity

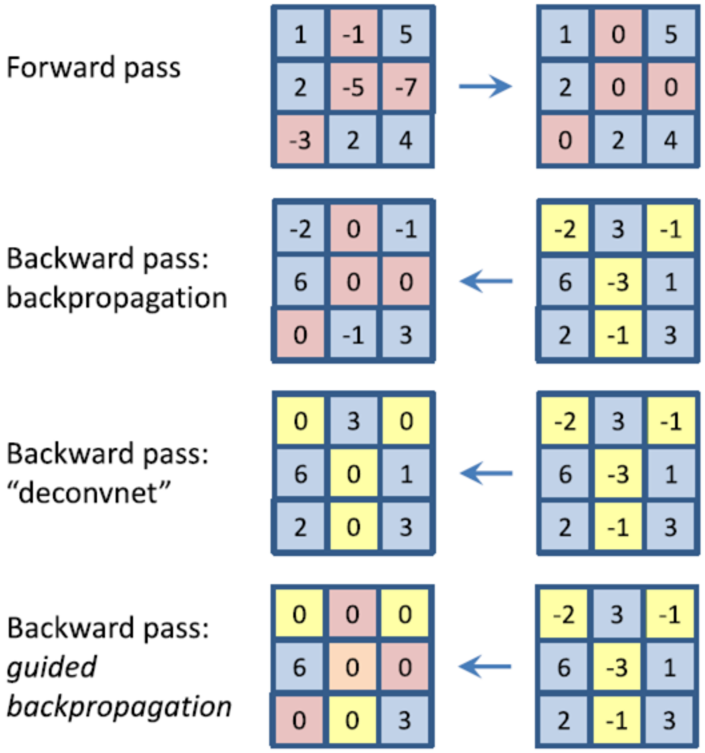


Vanilla Backpropagation

Guided Backpropagation

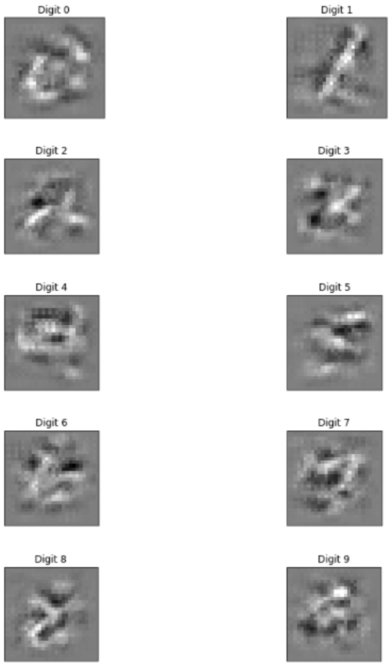
- Guided backpropagation is a **decision-based approach**

- **Gradients through ReLU** are masked out
  (1) **bottom data** (vanilla backpropagation) is **negative**
  (2) **the top gradient** (deconvnet) is **negative**

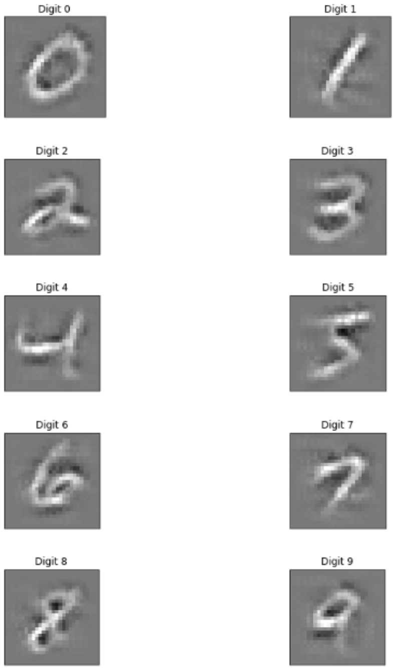


Different methods of propagating back through a **ReLU** nonlinearity.

Guided Backpropagation - Results & Comparison



Vanilla Backpropagation



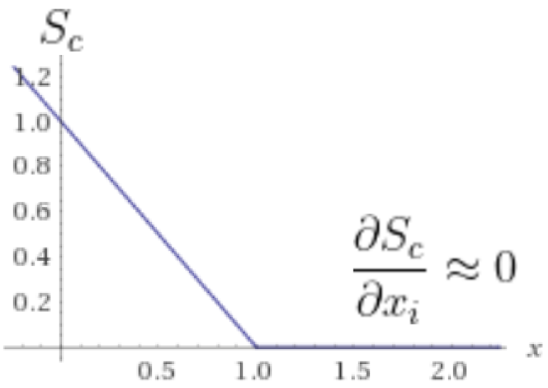
Guided Backpropagation

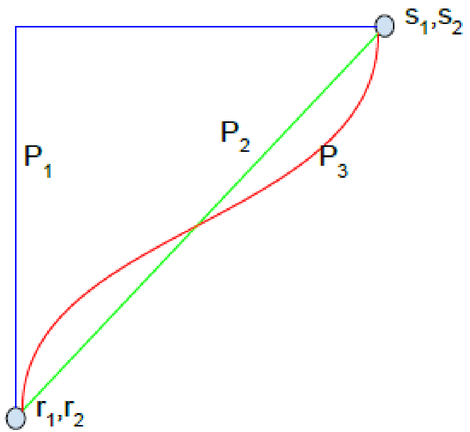
Integrated Gradients

- Integrated Gradients is a **decision-based approach**
- **Motivation:**
  Changing from the 1.0 to 2.0 gives attribution of **0 gradient** to x -> Gradients break sensitivity

- Integrate gradients at all points along a **straight line path** from **baseline**(black image) to the **input**

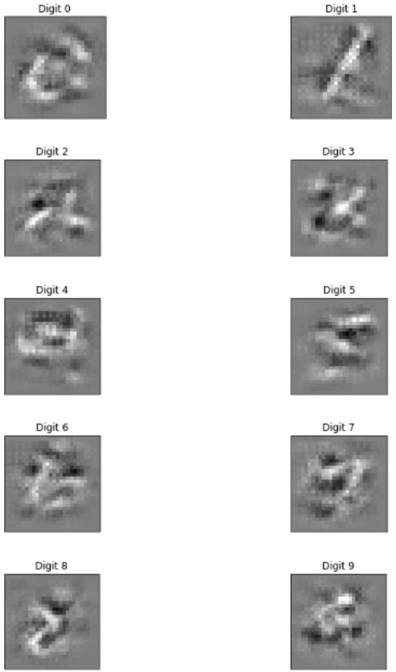
$$Integrated\ \ Gradient = \int_{\alpha=0}^{1} \frac{\partial S_c(x' + \alpha \times (x - x'))}{\partial x} d\alpha$$



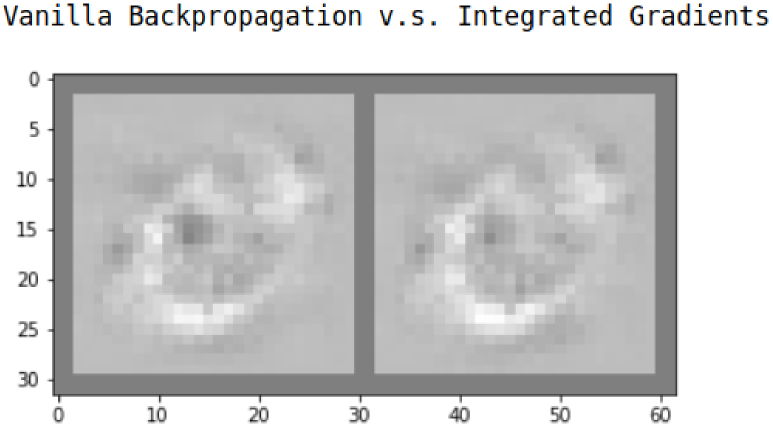


Path Methods

Integrated Gradients - Results & Comparison

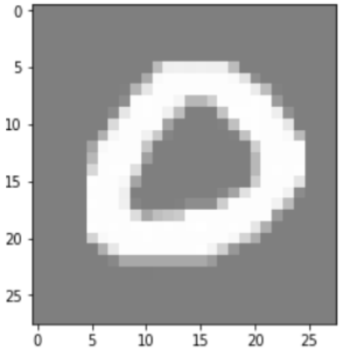


Integrated Gradient

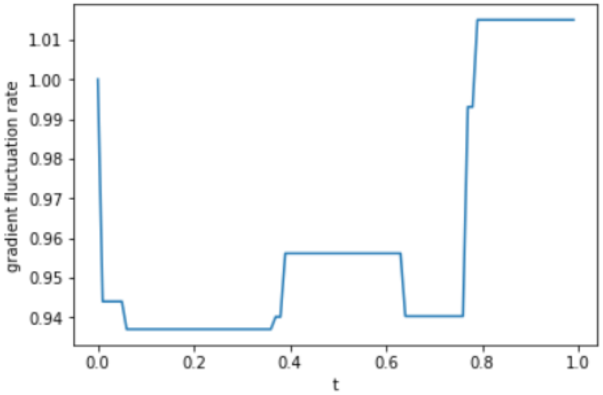


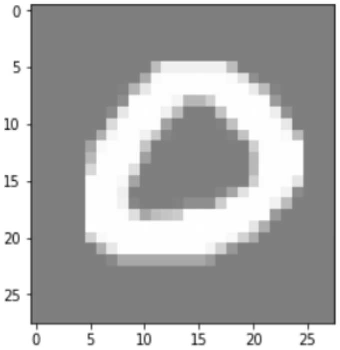
SmoothGrad

- SmoothGrad is a **decision-based approach**
- **Motivation:**
  Noise in vanilla backpropagation comes from **meaningless local variations** in partial derivatives
- Smooth ∂S_cj with a Gaussian kernel. In practice $\frac{1}{n}\sum_{1}^{n}\frac{\partial S_c(x+\mathcal{N}(0,\sigma^2))}{\partial x}$



baseline image x

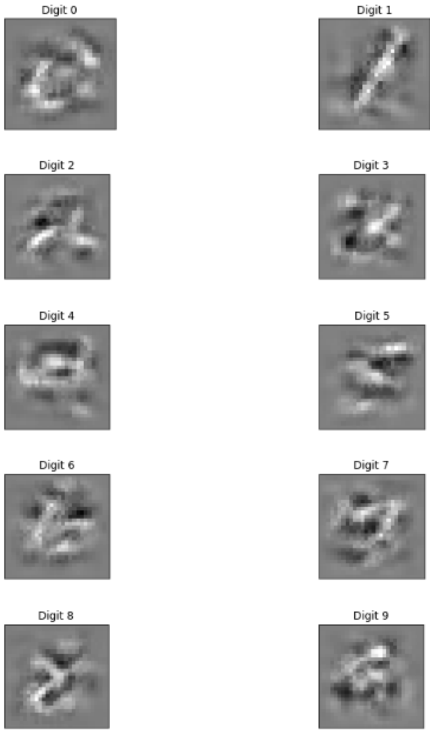




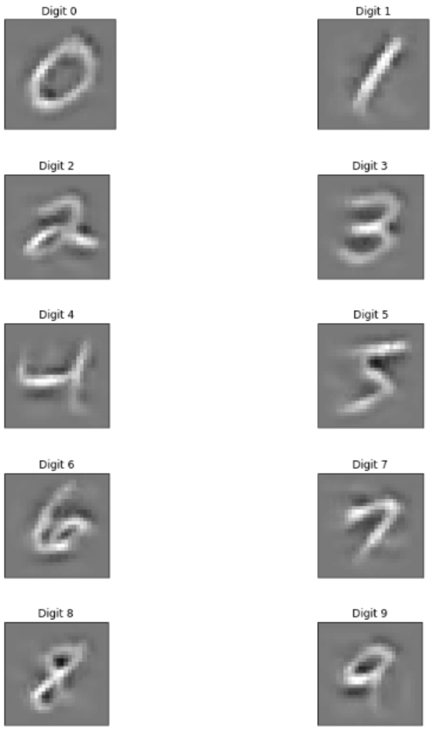
final image x + ϵ $\mathcal{N}(0, 0.001^2)$

ϵ is a random sample from

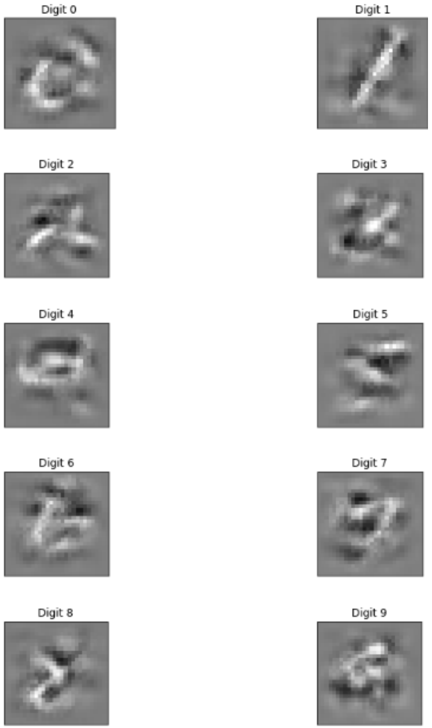
SmoothGrad - Results



SmoothGrad



SmoothGrad + Guided Backprop



SmoothGrad + Integrated Gradients

Class Activation Maps (CAM)

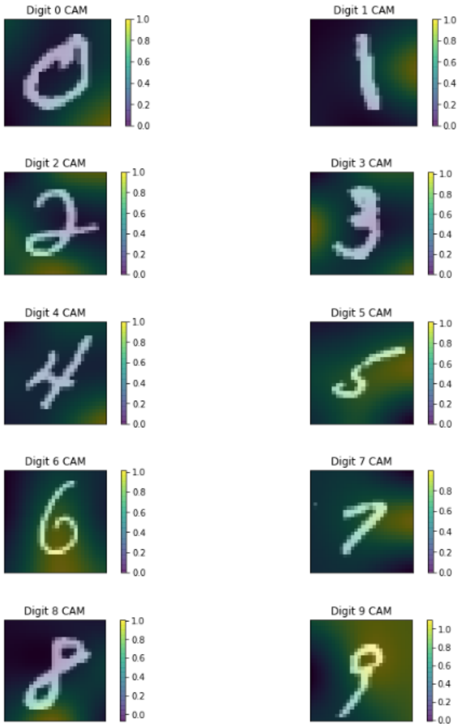
- CAM is a **decision-based approach** that highlights the parts of the input which had the highest influence to the model’s decision.

- It is **limited to a specific family of CNNs** which end with a AVGPOOL layer followed by a single classifier DENSE layer.

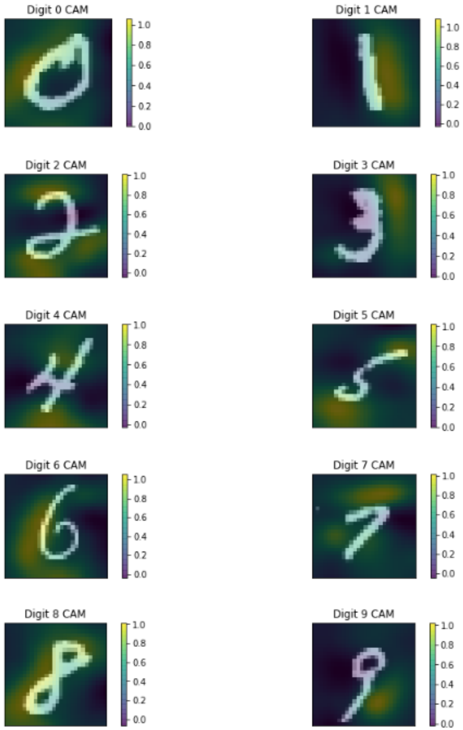
- CAM are generated by **multiplying the activation maps of the last CONV layer with the weights of the classifier layer.**

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

CAM Results



CAM Results: Model 1



CAM Results: Model 2

Gradient-Weighted Class Activation Map (Grad-CAM)

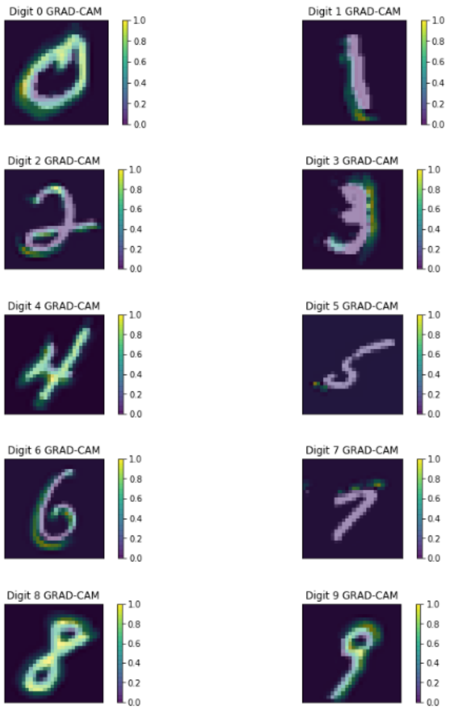
- Grad-CAM is a **decision-based method** which represents a **generalization of the CAM method.**

- Grad-CAM provides a **support for all CNN architectures and CONV layers** within a network.

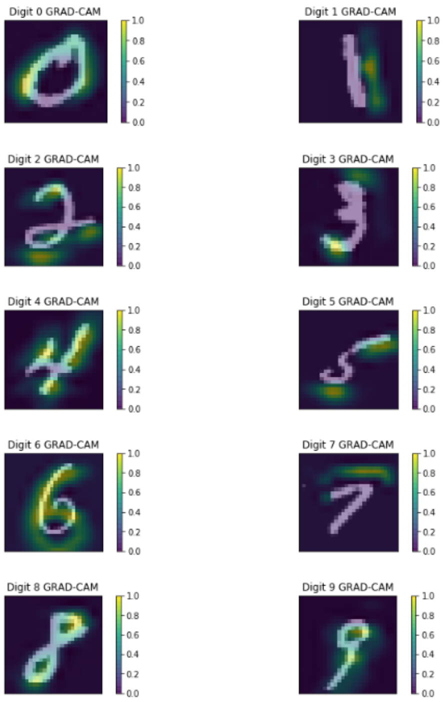
- Grad-CAMs are generated by **multiplying the activation maps of the chosen CONV layer with the global-average-pooled incoming gradient.**

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

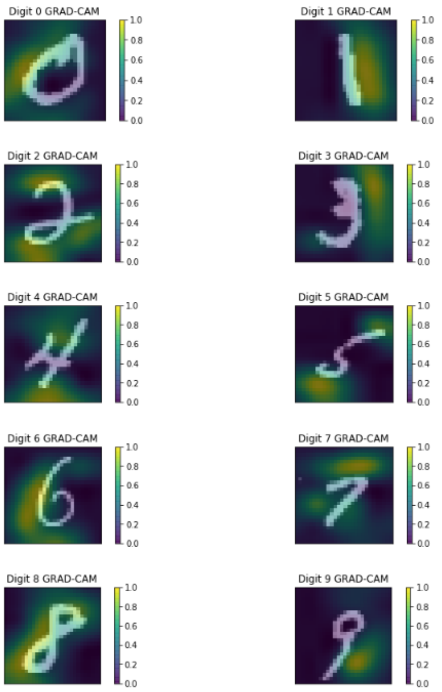
Grad-CAM Results



Grad-CAM Results: CONV 1



Grad-CAM Results: CONV 2



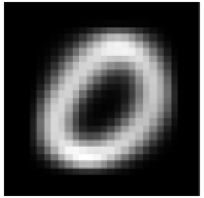
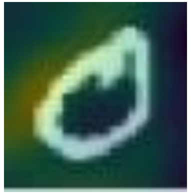
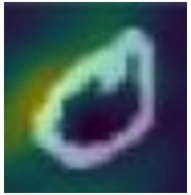
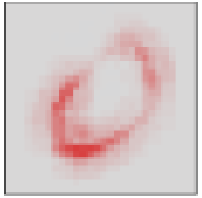
Grad-CAM Results: CONV 3

Interpretability Methods Overview

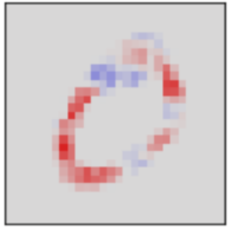
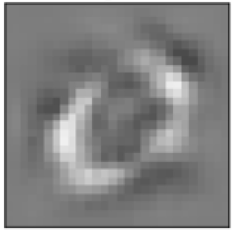
- Model-based Methods
    - Activation Maximization
        - General AM, AM in Codespace, AM with Expert

- Decision-based Methods
    - CAM
        - CAM, GradCAM
    - Deconvolution
    - Decomposition
        - Simple Taylor Decomposition, LRP
    - Gradient
        - Saliency Map, Guided Backprop, Integrated Gradients, SmoothGrad

Interpretability Overview I

	AM 	CAM 	Grad-CAM 	Deconvolution 	LRP 
Type	Model	Decision	Decision	Decision	Decision
Use case	Find the prototype of each class	Highlight important parts of the input	Highlight important parts of the input	Reconstruct output from input	Show pixel direct contributions
Complexity	**high**	**low**	**low**	**middle**	**middle**
Support	No restrictions	Subset of CNN	All CNNs	Need MAXPOOL	No restrictions
Drawback	Unstable	Strong support limitations	Interpolation issues	Need for second NN	Hard to choose between rules

Interpretability Overview II

	Taylor Expansion 	Saliency Map 	Guided Backprop 	Integrated Gradients 	SmoothGrad 
Type	Decision	Decision	Decision	Decision	Decision
Use case	Show pixel direct contributions	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
Complexity	**middle**	**low**	**low**	**low**	**low**
Support	No restrictions	No restrictions	Requires ReLU	No restrictions	No restrictions
Drawback	Hard to find root point	Noisy, Shattered gradients	Shattered gradients Easy to fail with uniform background	Shattered gradients	Shattered gradients

What comes next?

- Implementation of further methods
  - DeepDream
  - AM with an Expert

- Migration of available implementations in a package
  - Consistent class-based infrastructure for all methods
  - GPU support
  - Increase stability

- Documentation finalization

Sources

1) 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.
2) Matthew D. Zeiler and Rob Fergus (2013). Visualizing and Understanding Convolutional NetworksCoRR, abs/1311.2901.
3) Layer-Wise Relevance Propagation: An Overview
4) Olah, et al., "Feature Visualization", Distill, 2017.
5) Olah, et al., "The Building Blocks of Interpretability", Distill, 2018.
6) Karen Simonyan, Andrea Vedaldi, & Andrew Zisserman. (2013). Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps.