Improving Explainability of Integrated Gradients with Guided Non-Linearity

Hyuk Jin Kwon, Hyung Il Koo, **Nam Ik Cho**

SEOUL
NATIONAL
UNIVERSITY

Introduction

Interpretable Machine Learning

- Category of Techniques (Global vs Local)
  - Global interpretability
  - Local interpretability

Interpretable Machine Learning

- Category of Techniques (Global vs Local)
  - Global interpretability
    - Users can understand how the model works globally by inspecting the structures and parameters of a complex model.
    - Achieved by understanding the representations captured by the neurons at an intermediate layer.
  - Local interpretability

Interpretable Machine Learning

- Category of Techniques (Global vs Local)
  - Global interpretability
    - Users can understand how the model works globally by inspecting the structures and parameters of a complex model.
    - Achieved by understanding the representations captured by the neurons at an intermediate layer
  - Local interpretability
    - Locally examines an individual prediction of a model, trying to figure out why the model makes the decision it makes.
    - Identifying the contributions of each feature in a specific input to the prediction

Interpretable Machine Learning

- For Deep Neural Networks,
  - Hard to give constraint on each layer (Hard to give intrinsic properties)

    → prefer post-hoc interpretability

  - Learned deep representations are usually not human interpretable (Hard to tell meanings of features)

    → prefer local interpretability

Interpretable Machine Learning

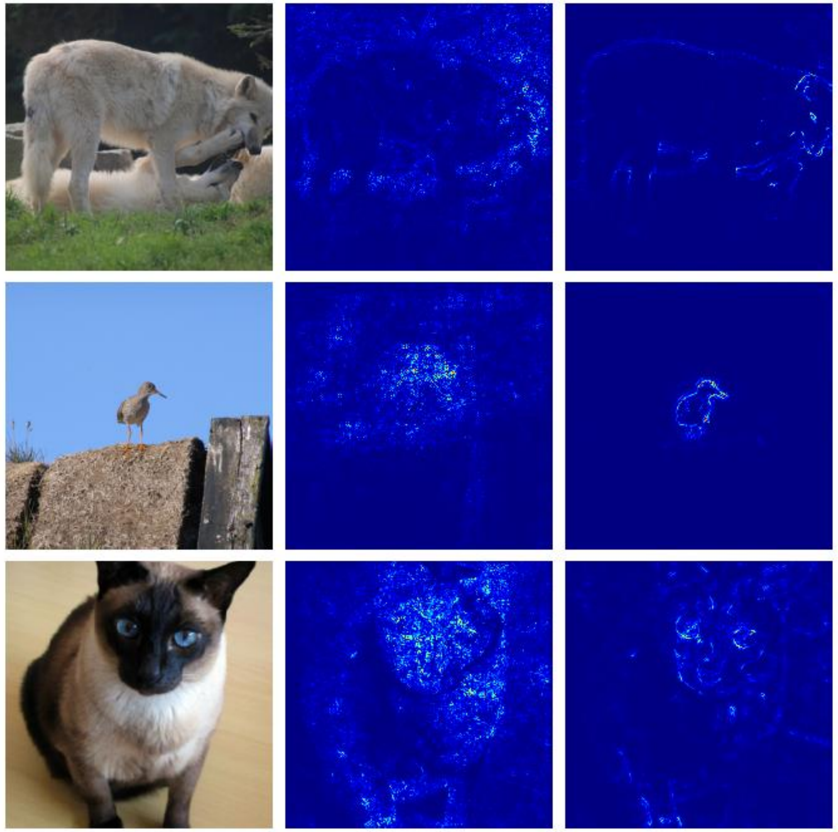
- For Deep Neural Networks,

  Post-hoc Local Explanation

  - Target to identify the contributions of each feature in the input towards a specific model prediction

  - Also called attribution methods

Interpretable Machine Learning



Label: white wolf

Label: redshank

Label: siamese cat

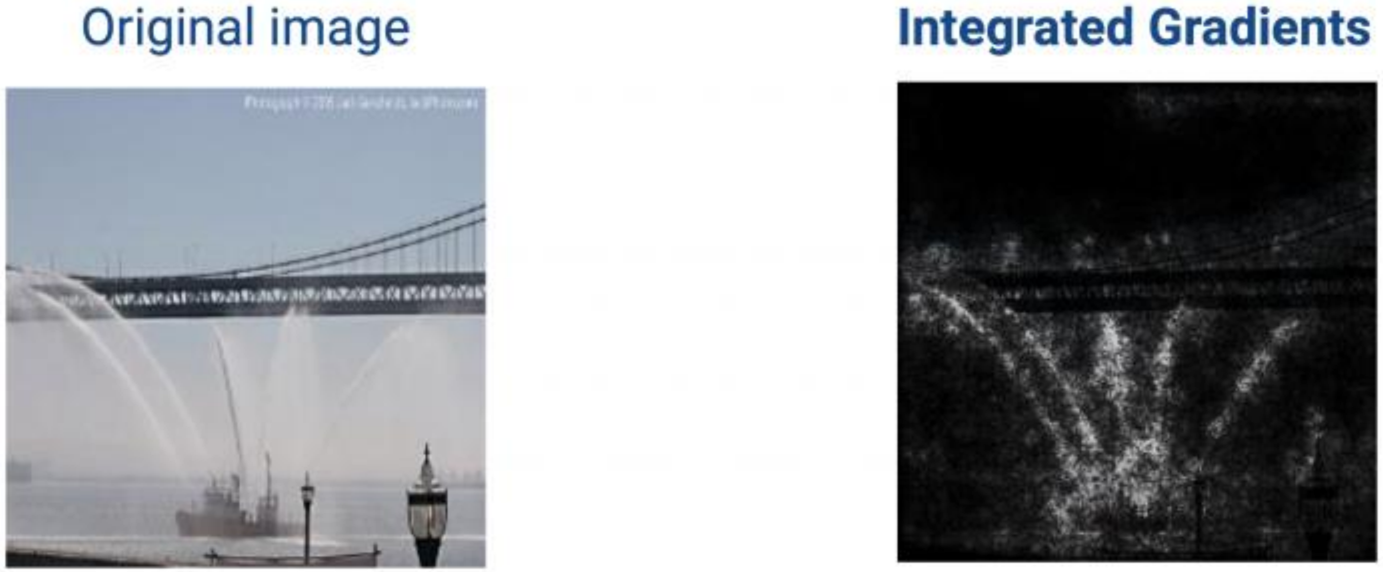
(a) Input (b) IG (c) Ours

[1] Mukund Sundararajan, Ankur Taly, and Qiqi Yan, “Axiomatic attribution for deep networks,” in *ICML*, 2017, pp. 3319–3328. (https://blog.fiddler.ai/2020/04/video-ai-explained-what-are-integrated-gradients/)

Existing Methods

- **Integrated Gradients**[1]

$$\mathrm{IG}(\text{input}, \text{base}) ::= (\text{input} - \text{base}) * \int_{0\text{-}1} \nabla F(\alpha * \text{input} + (1-\alpha) * \text{base})\, d\alpha$$



Our Approach

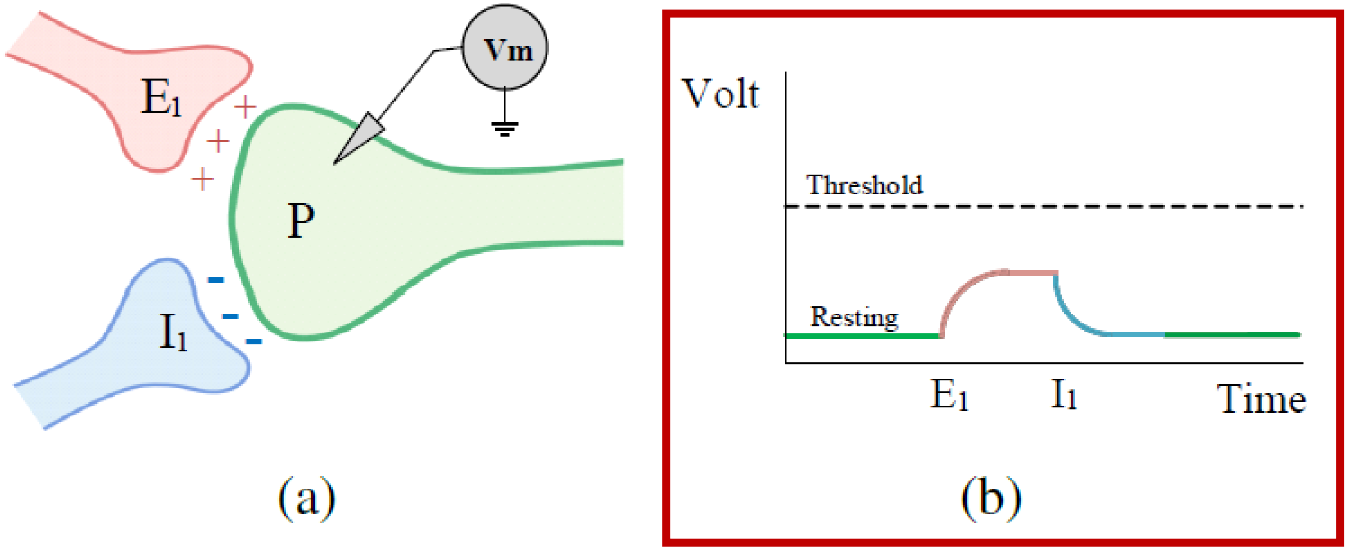
Motivation

- **Neural networks are based on the modeling of neurons** that have linear and non-linear parts.

- The **non-linear operators** in neural networks could be considered **axonal terminals** that **control the generation of action potentials** in postsynaptic cells by releasing neurotransmitters.

EPSP: Excitatory postsynaptic potential
IPSP: Inhibitory postsynaptic potential

Motivation

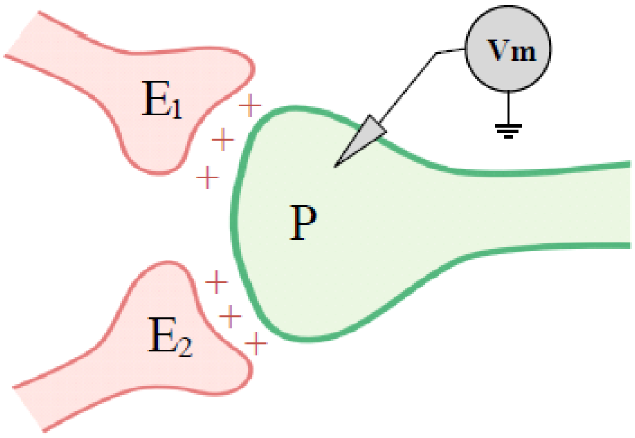


(a) (b)

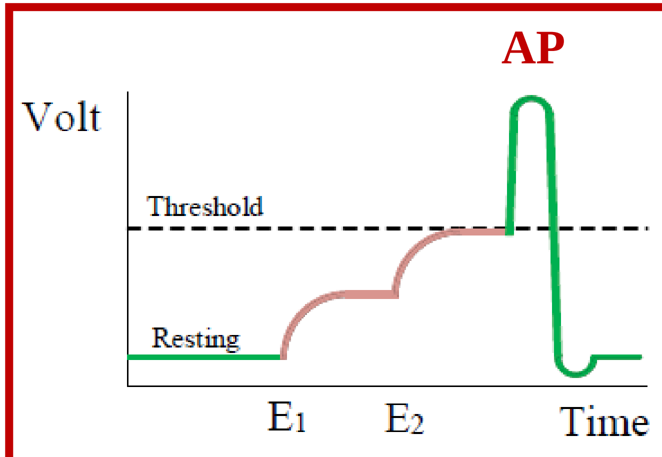
(a) Synaptic cleft consists of two presynaptic neurons that **one generates EPSP (E_1) and the other generates IPSP (I_1)**

(b) Potential in postsynaptic neuron (P) → **No action potential (AP)**

Motivation



(c)



(d)

(c) Synaptic cleft consists of **two presynaptic neurons that generates EPSPs (E_1 and E_2)**

(d) Potential in postsynaptic neuron (P) → **Action potential (AP)**

Motivation

- We think that **non-linear units (ReLU and max-pool) with positive gradients operate as EPSPs and the negative gradients as IPSPs.**

- Thus, **when we want to find the chain of fired neurons** (with the backpropagation of gradients), **we have to focus on neurons that generated EPSPs**, not IPSPs.

- In other words, **we have to focus on positive gradients to find the cause of the current prediction**

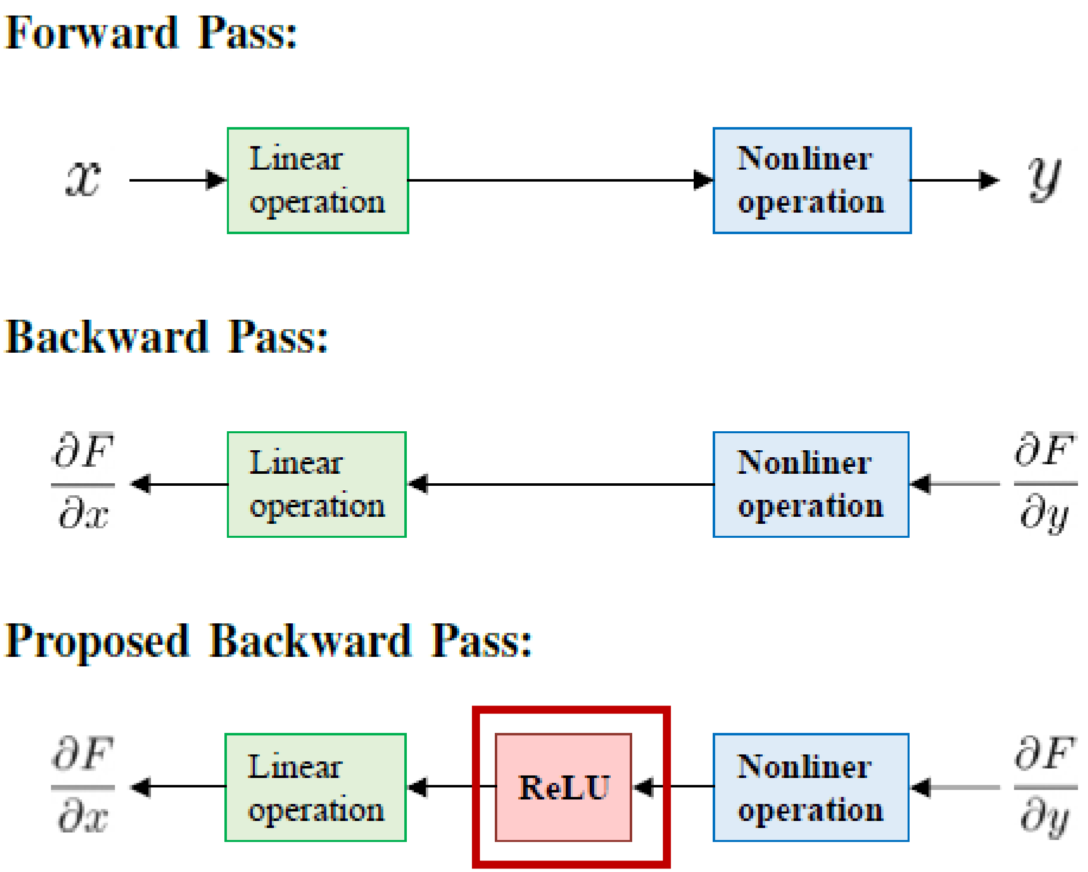
Proposed Method

- We computationally achieve this goal:

  1. **Clip negatively valued gradients in non-linear units to zero.**

  2. **Use these new gradients in the path integral of IG**

Proposed Method



Proposed Method

- **For a ReLU,**

Forward pass: $y = \mathrm{relu}(x) = x \odot I(x > 0)$

Backward pass: $\dfrac{\partial F(\cdot)}{\partial x} = \dfrac{\partial F(\cdot)}{\partial y} \odot I(x > 0)$

Proposed backward pass: $\dfrac{\partial F(\cdot)}{\partial x} = \boxed{\mathrm{relu}}\left(\dfrac{\partial F(\cdot)}{\partial y} \odot I(x > 0)\right)$

Proposed Method

- **For a max-pool,**

Forward pass: $y_i = \max_j x_{ij}$

Backward pass: $\frac{\partial F(\cdot)}{\partial x_{ij}} = \frac{\partial F(\cdot)}{\partial y_i} \odot I(x_{ij} = y_i)$

Proposed backward pass: $\frac{\partial F(\cdot)}{\partial x_{ij}} = \boxed{\mathrm{relu}} \left(\frac{\partial F(\cdot)}{\partial y_i} \odot I(x_{ij} = y_i) \right)$

Evaluation

Evaluation

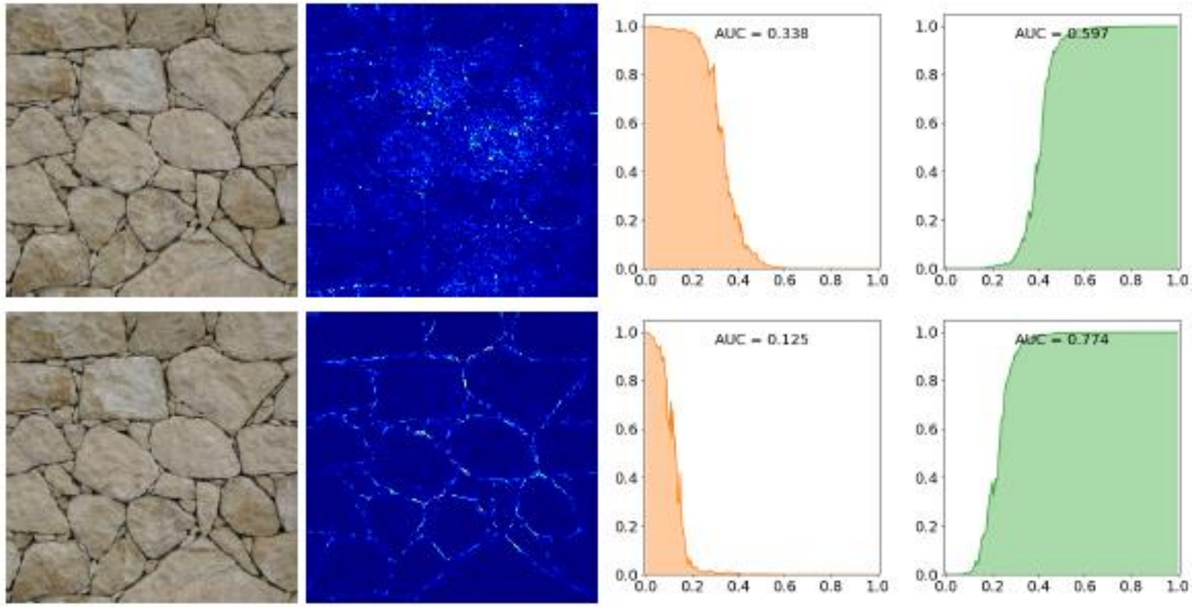
- **Networks: 5 CNN architectures**
  - VGG16, VGG19, ResNet34, ResNet50 and GoogleNet.
  - Trained for ImageNet2012 classification task.
- **Dataset:**
  - Validation split of ImageNet2012 classification database.
  - 5,000 linearly sampled (1/10) images are used for test.
- **Evaluation Metrics:**
  - Deletion/insertion metrics

Evaluation

- Qualitative Results

IG

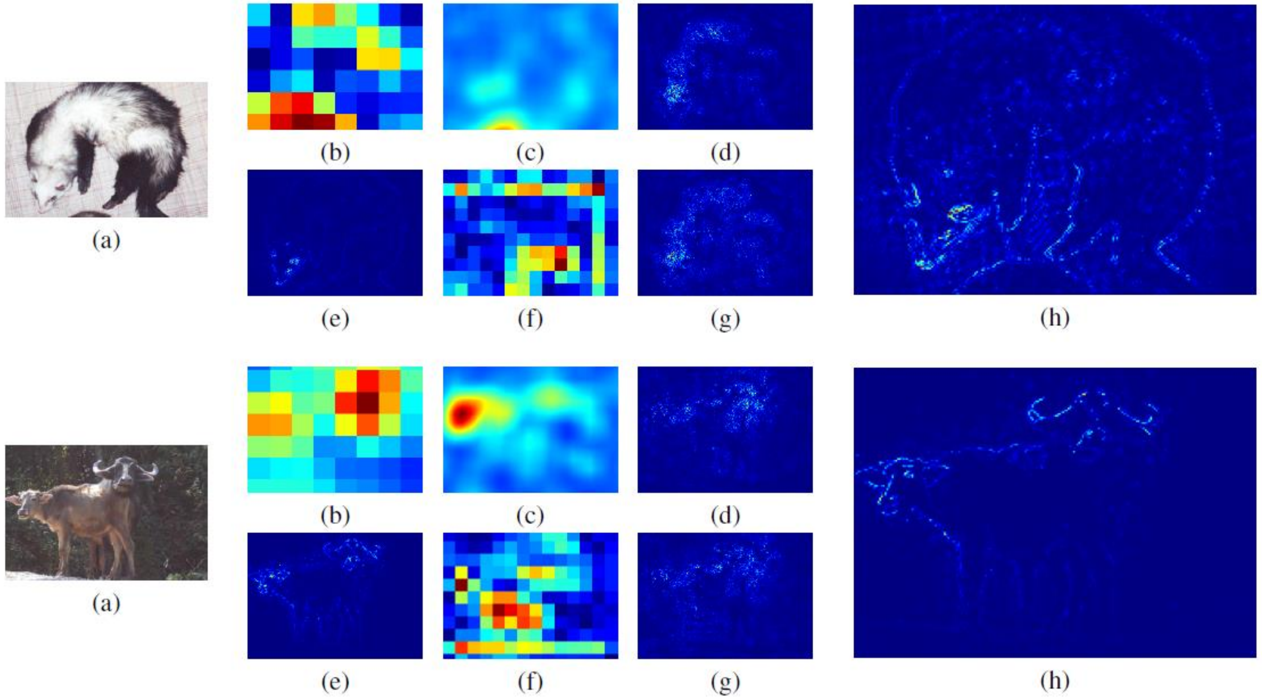
Ours



Input (*Stone Wall*) | Attribution | Del. metric | Ins. metric

Evaluation

- Qualitative Results



Evaluation

- Quantitative Results

	VGG16		VGG19		ResNet34		ResNet50		GoogleNet	
Methods	Deletion ↓	Insertion ↑	Deletion ↓	Insertion ↑	Deletion ↓	Insertion ↑	Deletion ↓	Insertion ↑	Deletion ↓	Insertion ↑
Occlusion [17]	0.1577	0.5755	0.1616	0.5770	0.1874	0.5914	0.2141	0.6309	0.1350	0.4667
LIME [28]	0.1014	0.6167	-	-	-	-	0.1217	**0.6940**	-	-
RISE [19]	0.0964	**0.6048**	0.0998	**0.6070**	0.1028	0.6308	0.1121	0.6762	0.0684	0.4995
Gradients [33]	0.0672	0.3270	0.0791	0.3423	0.1268	0.4221	0.1134	0.4234	0.0745	0.3574
GB [18]	0.0526	0.5279	0.0567	0.5445	0.0826	0.6141	0.0755	0.6460	0.0639	**0.5124**
GradCam [34]	0.1605	0.4305	0.1520	0.4578	0.1557	**0.6333**	0.1887	0.6715	0.1156	0.5086
IG [6]	0.0543	0.3621	0.0640	0.3792	0.1030	0.4575	0.0931	0.4589	0.0634	0.3936
Ours	**0.0495**	0.5151	**0.0532**	0.5295	**0.0763**	0.5932	**0.0721**	0.6295	**0.0601**	0.4912

Thank You