

Explaining CNNs: Recent Methods

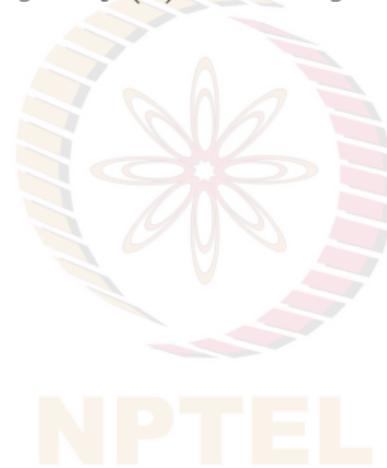
Vineeth N Balasubramanian

Department of Computer Science and Engineering
Indian Institute of Technology, Hyderabad



Recall: Explaining using Vanilla Gradients¹

- Forward pass the data \mathbf{x} , to get $y = f(\mathbf{x})$, where y is DNN's output corresponding to a given class.



¹Simonyan et al, Deep Inside Convolutional Networks:Visualising Image Classification Models and Saliency Maps, ICLRW 2014

Recall: Explaining using Vanilla Gradients¹

- Forward pass the data \mathbf{x} , to get $y = f(\mathbf{x})$, where y is DNN's output corresponding to a given class.
- Backward pass to input layer to get the gradient $\frac{\partial y}{\partial \mathbf{x}}$.



¹Simonyan et al, Deep Inside Convolutional Networks:Visualising Image Classification Models and Saliency Maps, ICLRW 2014

Recall: Explaining using Vanilla Gradients¹

- Forward pass the data \mathbf{x} , to get $y = f(\mathbf{x})$, where y is DNN's output corresponding to a given class.
- Backward pass to input layer to get the gradient $\frac{\partial y}{\partial \mathbf{x}}$.



Original image (*left*); Vanilla Gradients Attribution map (*right*)

Is this enough to explain a Deep Neural Network?

¹Simonyan et al, Deep Inside Convolutional Networks:Visualising Image Classification Models and Saliency Maps, ICLRW 2014

Recall: Explaining using Vanilla Gradients¹

- Forward pass the data \mathbf{x} , to get $y = f(\mathbf{x})$, where y is DNN's output corresponding to a given class.
- Backward pass to input layer to get the gradient $\frac{\partial y}{\partial \mathbf{x}}$.



Original image (*left*); Vanilla Gradients Attribution map (*right*)

*Is this enough to explain a Deep Neural Network?
Not always!*

¹Simonyan et al, Deep Inside Convolutional Networks:Visualising Image Classification Models and Saliency Maps, ICLRW 2014

Saturation Problem!

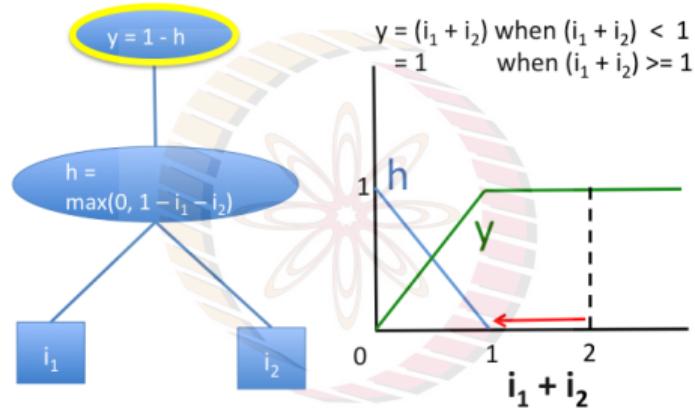


Illustration of saturation problem

Saturation Problem!

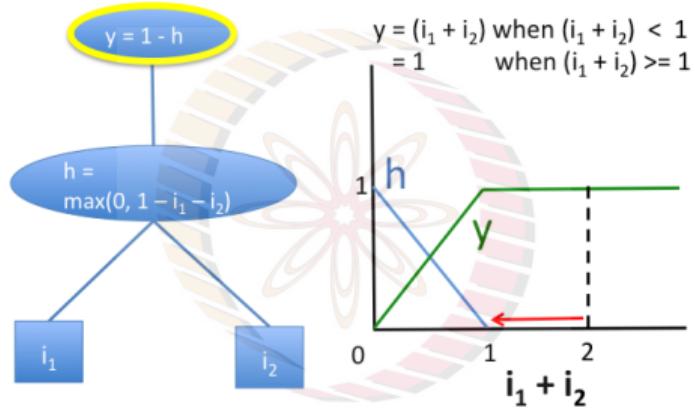


Illustration of saturation problem

- Gradient of h w.r.t both i_1 and i_2 is zero when $i_1 + i_2 > 1$ (causing both gradients and Guided Backprop to be zero)

Saturation Problem!

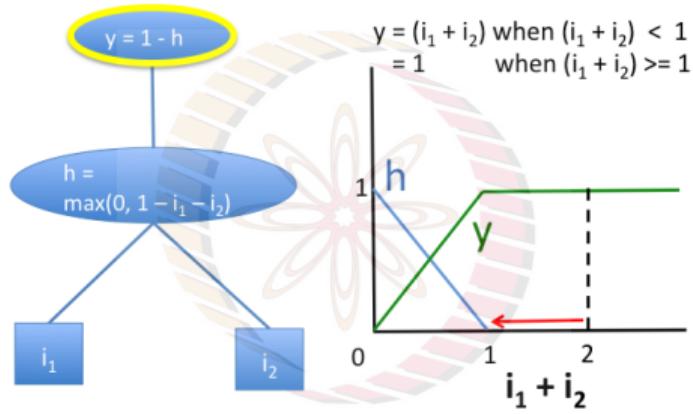


Illustration of saturation problem

- Gradient of h w.r.t both i_1 and i_2 is zero when $i_1 + i_2 > 1$ (causing both gradients and Guided Backprop to be zero)
- Gradient of y w.r.t. h is negative (causing Guided Backprop and deconvolutional networks to assign zero importance)

Deep Lift²

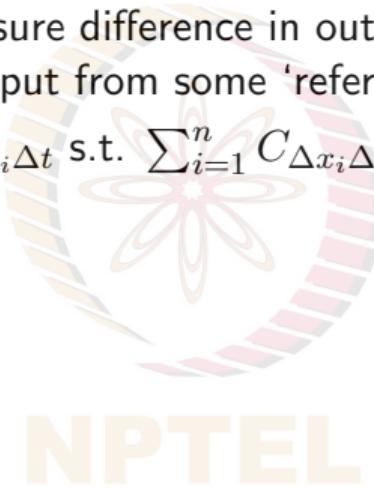
- **Idea:** Instead of gradients, measure difference in output from some ‘reference’ output (Δt) in terms of difference of input from some ‘reference’ input (Δx_i).



²Shrikumar et al, Learning important features through propagating activation differences, ICML 2017

Deep Lift²

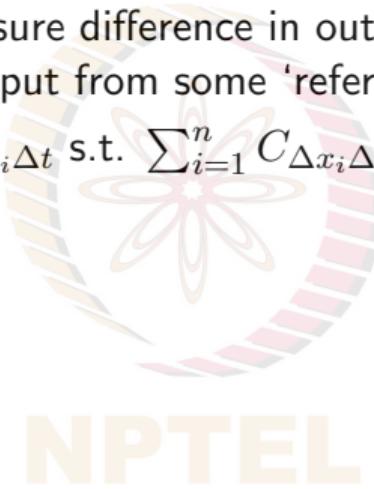
- **Idea:** Instead of gradients, measure difference in output from some ‘reference’ output (Δt) in terms of difference of input from some ‘reference’ input (Δx_i).
- Assigns contribution scores $C_{\Delta x_i \Delta t}$ s.t. $\sum_{i=1}^n C_{\Delta x_i \Delta t} = \Delta t$



²Shrikumar et al, Learning important features through propagating activation differences, ICML 2017

Deep Lift²

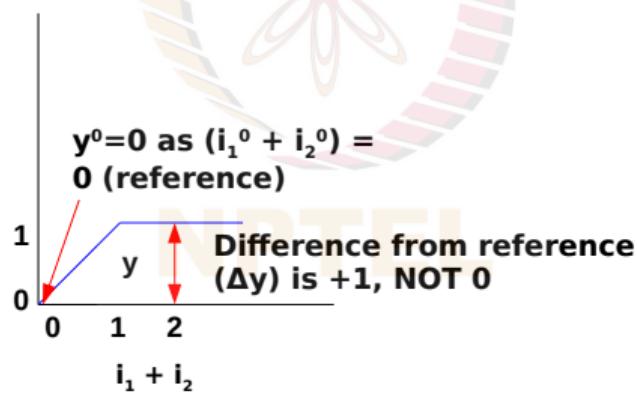
- **Idea:** Instead of gradients, measure difference in output from some ‘reference’ output (Δt) in terms of difference of input from some ‘reference’ input (Δx_i).
- Assigns contribution scores $C_{\Delta x_i \Delta t}$ s.t. $\sum_{i=1}^n C_{\Delta x_i \Delta t} = \Delta t$



²Shrikumar et al, Learning important features through propagating activation differences, ICML 2017

Deep Lift²

- **Idea:** Instead of gradients, measure difference in output from some ‘reference’ output (Δt) in terms of difference of input from some ‘reference’ input (Δx_i).
- Assigns contribution scores $C_{\Delta x_i \Delta t}$ s.t. $\sum_{i=1}^n C_{\Delta x_i \Delta t} = \Delta t$



DeepLift overcomes saturation problem

²Shrikumar et al, Learning important features through propagating activation differences, ICML 2017

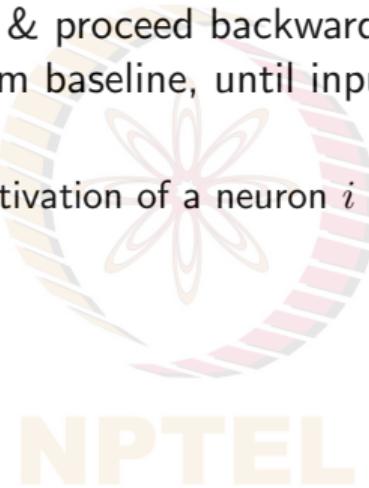
Deep Lift: Rescale Rule

- **Idea:** Start from output layer L & proceed backwards layer by layer, redistributing the difference of prediction score from baseline, until input layer is reached



Deep Lift: Rescale Rule

- **Idea:** Start from output layer L & proceed backwards layer by layer, redistributing the difference of prediction score from baseline, until input layer is reached
- **Notations:**
 - $z_{ji} = w_{ji}^{(l+1,l)} x_i^l$: weighted activation of a neuron i onto neuron j in the next layer



Deep Lift: Rescale Rule

- **Idea:** Start from output layer L & proceed backwards layer by layer, redistributing the difference of prediction score from baseline, until input layer is reached
- **Notations:**
 - $z_{ji} = w_{ji}^{(l+1,l)} x_i^l$: weighted activation of a neuron i onto neuron j in the next layer
 - $\bar{z}_{ji} = w_{ji}^{(l+1,l)} \bar{x}_i^l$: weighted activation of a neuron i onto neuron j in the next layer, when baseline \bar{x} fed as the input.

The NPTEL logo features the letters "NPTEL" in a bold, sans-serif font. The letters are colored in a gradient of orange and yellow. Behind the letters is a stylized circular emblem composed of several curved, overlapping bands in shades of orange, yellow, and red, resembling a flower or a gear.

Deep Lift: Rescale Rule

- **Idea:** Start from output layer L & proceed backwards layer by layer, redistributing the difference of prediction score from baseline, until input layer is reached
- **Notations:**
 - $z_{ji} = w_{ji}^{(l+1,l)} x_i^l$: weighted activation of a neuron i onto neuron j in the next layer
 - $\bar{z}_{ji} = w_{ji}^{(l+1,l)} \bar{x}_i^l$: weighted activation of a neuron i onto neuron j in the next layer, when baseline \bar{x} fed as the input.
 - $r_i^{(l)}$: relevance of unit i of layer l .

NPTEL

Deep Lift: Rescale Rule

- **Idea:** Start from output layer L & proceed backwards layer by layer, redistributing the difference of prediction score from baseline, until input layer is reached
- **Notations:**
 - $z_{ji} = w_{ji}^{(l+1,l)} x_i^l$: weighted activation of a neuron i onto neuron j in the next layer
 - $\bar{z}_{ji} = w_{ji}^{(l+1,l)} \bar{x}_i^l$: weighted activation of a neuron i onto neuron j in the next layer, when baseline \bar{x} fed as the input.
 - $r_i^{(l)}$: relevance of unit i of layer l .
- $r_i^{(L)}: \begin{cases} = y_i(x) - y_i(\bar{x}) \text{ if unit } i \text{ is target unit of interest} \\ = 0 \text{ otherwise} \end{cases}$

Deep Lift: Rescale Rule

- **Idea:** Start from output layer L & proceed backwards layer by layer, redistributing the difference of prediction score from baseline, until input layer is reached
- **Notations:**
 - $z_{ji} = w_{ji}^{(l+1,l)} x_i^l$: weighted activation of a neuron i onto neuron j in the next layer
 - $\bar{z}_{ji} = w_{ji}^{(l+1,l)} \bar{x}_i^l$: weighted activation of a neuron i onto neuron j in the next layer, when baseline \bar{x} fed as the input.
 - $r_i^{(l)}$: relevance of unit i of layer l .
- $r_i^{(L)}: \begin{cases} = y_i(x) - y_i(\bar{x}) \text{ if unit } i \text{ is target unit of interest} \\ = 0 \text{ otherwise} \end{cases}$
- $r_i^{(l)} = \sum_j \frac{z_{ji} - \bar{z}_{ji}}{\sum_{i'} (z_{ji'} - \bar{z}_{ji'})} r_j^{(l+1)}$

IG: Integrated Gradients³

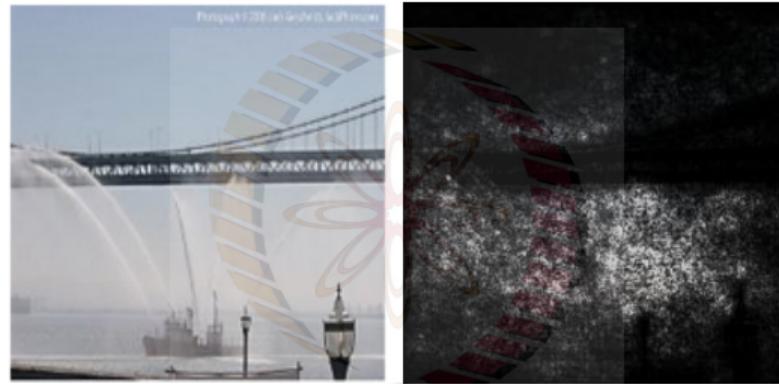


Image of Fireboat (*left*), Vanilla Gradients (*right*)

³Sundararajan et al, Axiomatic Attribution for Deep Networks, ICML 2017

IG: Integrated Gradients³

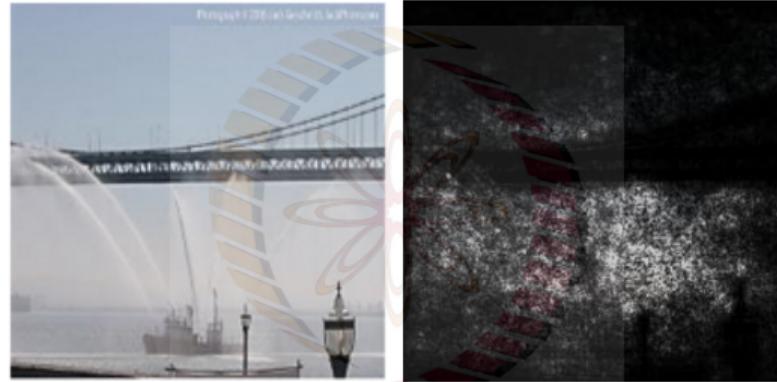


Image of Fireboat (*left*), Vanilla Gradients (*right*)

- Due to saturation problem, vanilla gradients highlight regions irrelevant to fireboat

³Sundararajan et al, Axiomatic Attribution for Deep Networks, ICML 2017

IG: Integrated Gradients³

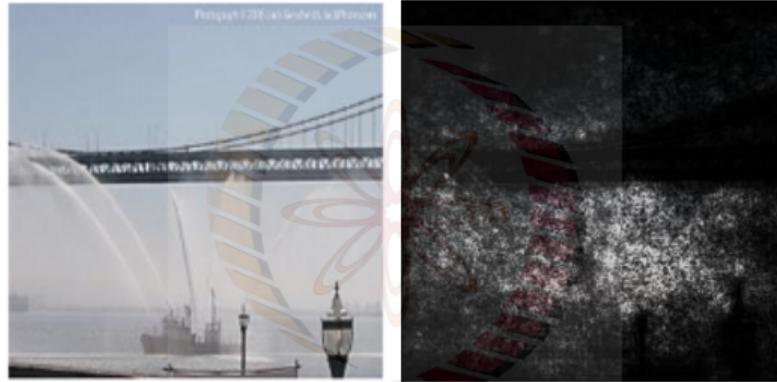
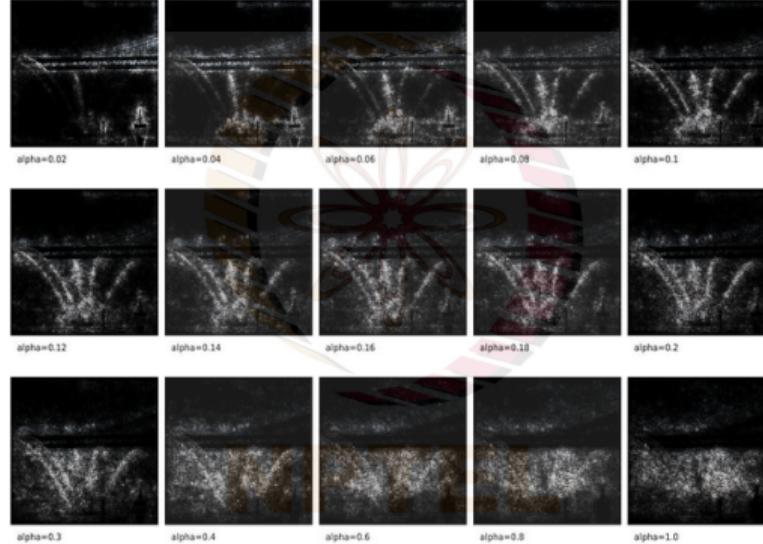


Image of Fireboat (*left*), Vanilla Gradients (*right*)

- Due to saturation problem, vanilla gradients highlight regions irrelevant to fireboat
- IG overcomes problem of saturating gradients by cumulating gradients at different pixel intensities, α 's.

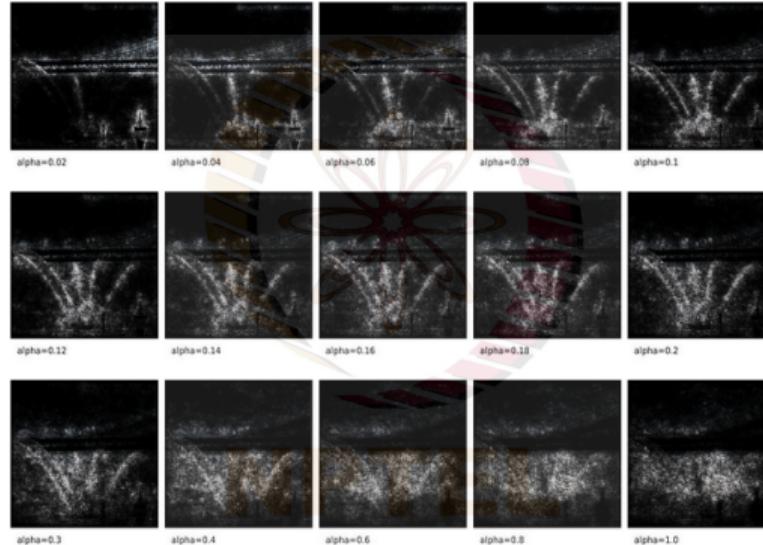
³Sundararajan et al, Axiomatic Attribution for Deep Networks, ICML 2017

IG: Integrated Gradients



Gradients at increasing α values from top-left to bottom-right

IG: Integrated Gradients



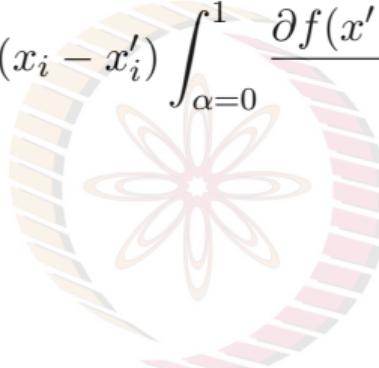
Gradients at increasing α values from top-left to bottom-right

- Region of importance is changing with increasing α . To get a more realistic picture of what is going on, cumulate these gradients using **path integral**

IG: Integrated Gradients

- **Integrated gradient** along i^{th} dimension for input x and baseline x' given by:

$$\text{IG}_i(x) := (x_i - x'_i) \int_{\alpha=0}^1 \frac{\partial f(x' + \alpha(x - x'))}{\partial x_i} d\alpha$$



NPTEL

IG: Integrated Gradients

- **Integrated gradient** along i^{th} dimension for input x and baseline x' given by:

$$\text{IG}_i(x) := (x_i - x'_i) \int_{\alpha=0}^1 \frac{\partial f(x' + \alpha(x - x'))}{\partial x_i} d\alpha$$

- $\text{IG}_i^{\text{approx}}(x) := (x_i - x'_i) \sum_{k=1}^m \frac{\partial f(x' + \frac{k}{m}(x - x'))}{\partial x_i} \frac{1}{m}$ where m is a hyperparameter.

The NPTEL logo consists of the word "NPTEL" in a bold, sans-serif font, with each letter in a different color: N is orange, P is blue, T is green, E is red, and L is yellow. Behind the letters is a circular emblem composed of several concentric arcs in shades of orange, yellow, and red, resembling a stylized flame or a rising sun.

NPTEL

IG: Integrated Gradients

- **Integrated gradient** along i^{th} dimension for input x and baseline x' given by:

$$\text{IG}_i(x) := (x_i - x'_i) \int_{\alpha=0}^1 \frac{\partial f(x' + \alpha(x - x'))}{\partial x_i} d\alpha$$

- $\text{IG}_i^{\text{approx}}(x) := (x_i - x'_i) \sum_{k=1}^m \frac{\partial f(x' + \frac{k}{m}(x - x'))}{\partial x_i} \frac{1}{m}$ where m is a hyperparameter.



IG attribution map

SmoothGrad⁴

- Add pixel-wise Gaussian noise to many copies of the image, and average resulting gradients.



⁴Smilkov et al, SmoothGrad: removing noise by adding noise, ICMLW 2017

SmoothGrad⁴

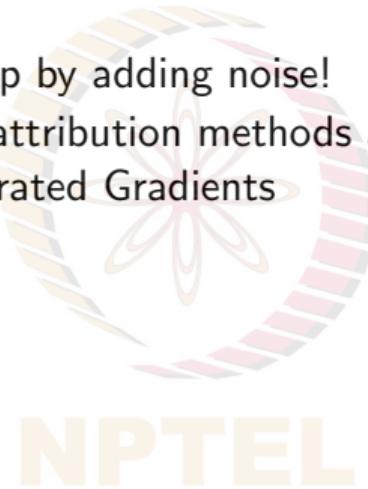
- Add pixel-wise Gaussian noise to many copies of the image, and average resulting gradients.
- Removes noise from saliency map by adding noise!



⁴Smilkov et al, SmoothGrad: removing noise by adding noise, ICMLW 2017

SmoothGrad⁴

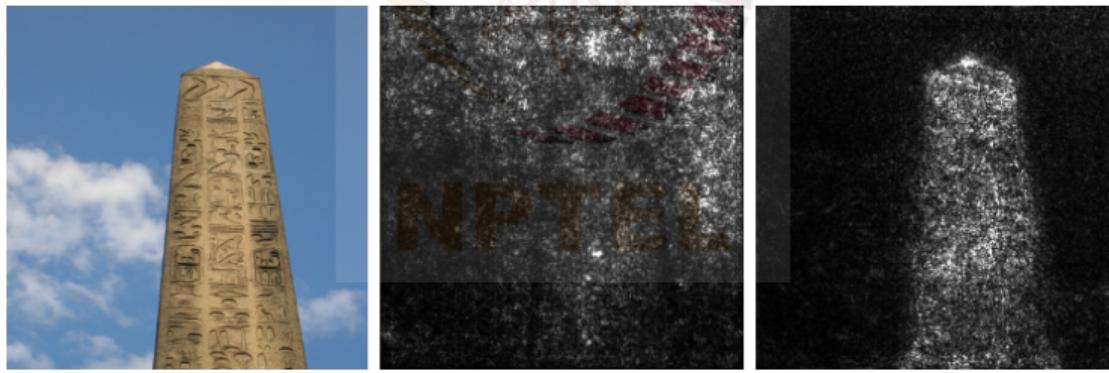
- Add pixel-wise Gaussian noise to many copies of the image, and average resulting gradients.
- Removes noise from saliency map by adding noise!
- Besides vanilla gradients, other attribution methods also have their SmoothGrad counterparts, e.g. Smooth Integrated Gradients



⁴Smilkov et al, SmoothGrad: removing noise by adding noise, ICMLW 2017

SmoothGrad⁴

- Add pixel-wise Gaussian noise to many copies of the image, and average resulting gradients.
- Removes noise from saliency map by adding noise!
- Besides vanilla gradients, other attribution methods also have their SmoothGrad counterparts, e.g. Smooth Integrated Gradients



Original Image (*left*), Vanilla Gradients (*center*), SmoothGrad (*right*)

⁴Smilkov et al, SmoothGrad: removing noise by adding noise, ICMLW 2017

Recent Variant of IG: XRAI⁵

- Get attribution map given by IG



⁵Kapishnikov et al, XRAI: Better Attributions Through Regions, ICCV 2019

Recent Variant of IG: XRAI⁵

- Get attribution map given by IG
- Over-segment the image



⁵Kapishnikov et al, XRAI: Better Attributions Through Regions, ICCV 2019

Recent Variant of IG: XRAI⁵

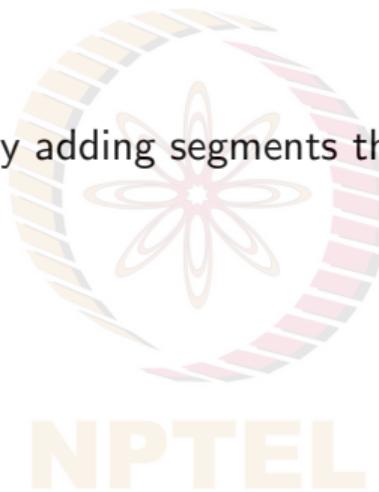
- Get attribution map given by IG
- Over-segment the image
- Start with an empty mask



⁵Kapishnikov et al, XRAI: Better Attributions Through Regions, ICCV 2019

Recent Variant of IG: XRAI⁵

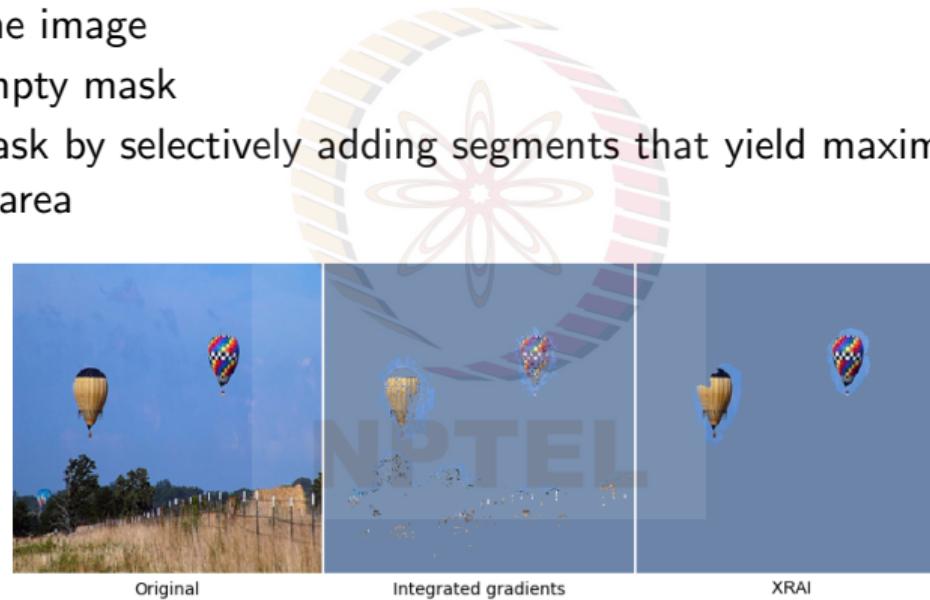
- Get attribution map given by IG
- Over-segment the image
- Start with an empty mask
- Populate this mask by selectively adding segments that yield maximum gain in total attributions per area



⁵Kapishnikov et al, XRAI: Better Attributions Through Regions, ICCV 2019

Recent Variant of IG: XRAI⁵

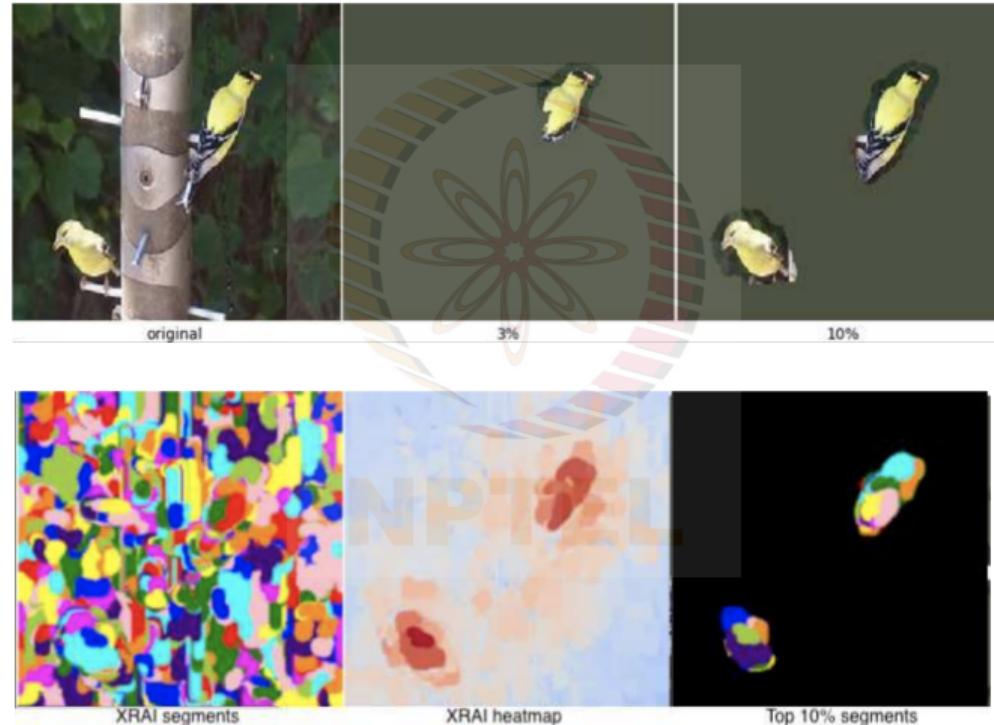
- Get attribution map given by IG
- Over-segment the image
- Start with an empty mask
- Populate this mask by selectively adding segments that yield maximum gain in total attributions per area



Original image (*left*), IG (*center*), XRAI (*right*)

⁵Kapishnikov et al, XRAI: Better Attributions Through Regions, ICCV 2019

Recent Variant of IG: XRAI⁶



⁶Kapishnikov et al, XRAI: Better Attributions Through Regions, ICCV 2019

LIME: Local Interpretable Model-agnostic Explanations⁷

- **Idea:** Approximate underlying model locally by an interpretable (typically linear) one



⁷Ribeiro et al, Why Should I Trust You?: Explaining the Predictions of Any Classifier, KDD 2016

LIME: Local Interpretable Model-agnostic Explanations⁷

- **Idea:** Approximate underlying model locally by an interpretable (typically linear) one
- Interpretable models are trained on small perturbations of original instance

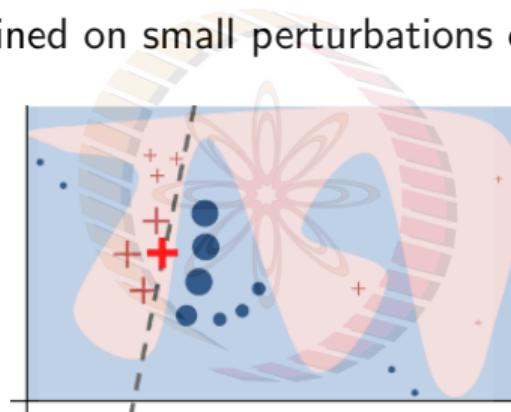


NPTEL

⁷Ribeiro et al, Why Should I Trust You?: Explaining the Predictions of Any Classifier, KDD 2016

LIME: Local Interpretable Model-agnostic Explanations⁷

- **Idea:** Approximate underlying model locally by an interpretable (typically linear) one
- Interpretable models are trained on small perturbations of original instance



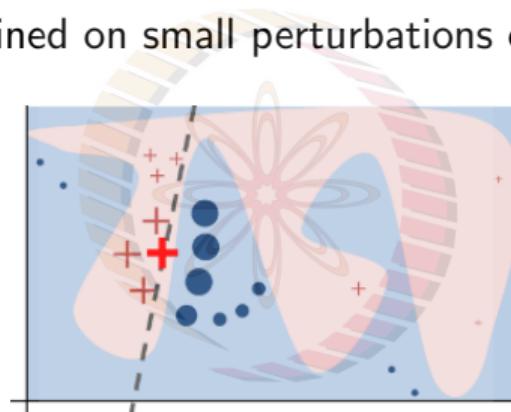
Intuition for LIME

- *Blue/Pink background:* black box model's decision function f

⁷Ribeiro et al, Why Should I Trust You?: Explaining the Predictions of Any Classifier, KDD 2016

LIME: Local Interpretable Model-agnostic Explanations⁷

- **Idea:** Approximate underlying model locally by an interpretable (typically linear) one
- Interpretable models are trained on small perturbations of original instance



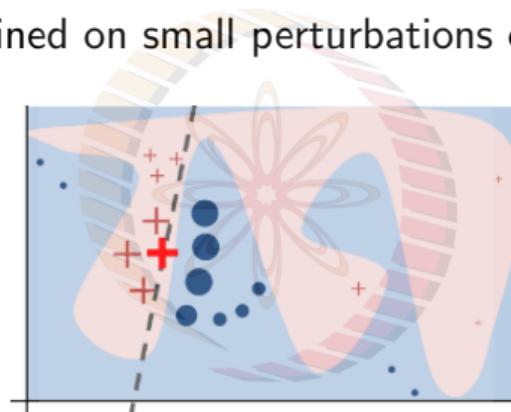
Intuition for LIME

- *Blue/Pink background:* black box model's decision function f
- *Bold red cross:* instance being explained

⁷Ribeiro et al, Why Should I Trust You?: Explaining the Predictions of Any Classifier, KDD 2016

LIME: Local Interpretable Model-agnostic Explanations⁷

- **Idea:** Approximate underlying model locally by an interpretable (typically linear) one
- Interpretable models are trained on small perturbations of original instance

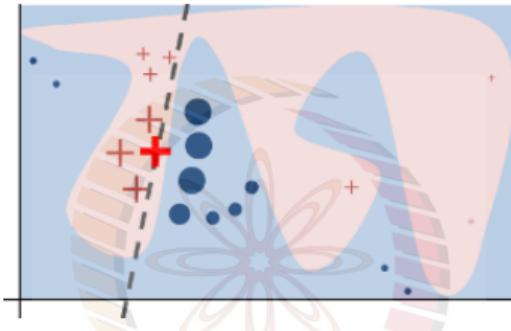


Intuition for LIME

- *Blue/Pink background:* black box model's decision function f
- *Bold red cross:* instance being explained
- *Dashed line:* learned explanation

⁷Ribeiro et al, Why Should I Trust You?: Explaining the Predictions of Any Classifier, KDD 2016

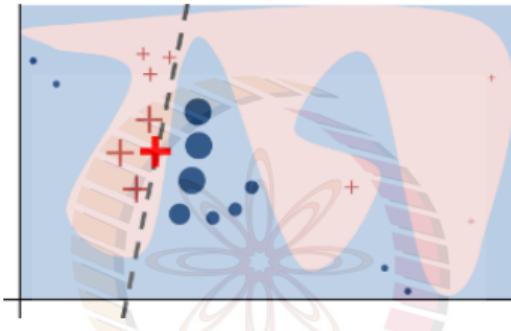
LIME: Local Interpretable Model-agnostic Explanations



- Given a point x , let $z' \in \mathcal{Z}$ be a point obtained by perturbing one dimension (or region) in x ; \mathcal{Z} is the set of perturbations

NPTEL

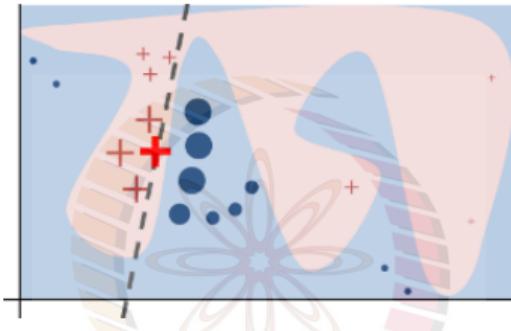
LIME: Local Interpretable Model-agnostic Explanations



- Given a point x , let $z' \in \mathcal{Z}$ be a point obtained by perturbing one dimension (or region) in x ; \mathcal{Z} is the set of perturbations
- $\pi_x(z')$: proximity measure between instances z' and x , e.g. $\pi_x(z') = \exp(-D(x, z')^2/\sigma^2)$

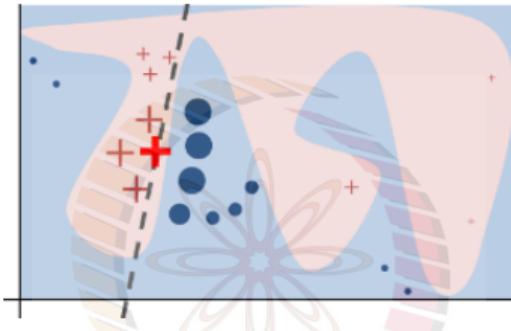
NPTEL

LIME: Local Interpretable Model-agnostic Explanations



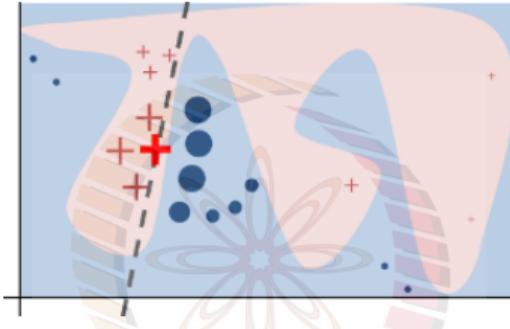
- Given a point x , let $z' \in \mathcal{Z}$ be a point obtained by perturbing one dimension (or region) in x ; \mathcal{Z} is the set of perturbations
- $\pi_x(z')$: proximity measure between instances z' and x , e.g. $\pi_x(z') = \exp(-D(x, z')^2/\sigma^2)$
- $f : \mathbb{R}^d \rightarrow \mathbb{R}$: model being explained

LIME: Local Interpretable Model-agnostic Explanations



- Given a point x , let $z' \in \mathcal{Z}$ be a point obtained by perturbing one dimension (or region) in x ; \mathcal{Z} is the set of perturbations
- $\pi_x(z')$: proximity measure between instances z' and x , e.g. $\pi_x(z') = \exp(-D(x, z')^2/\sigma^2)$
- $f : \mathbb{R}^d \rightarrow \mathbb{R}$: model being explained
- Build a sparse linear model, $g(z') = w_g \cdot z'$

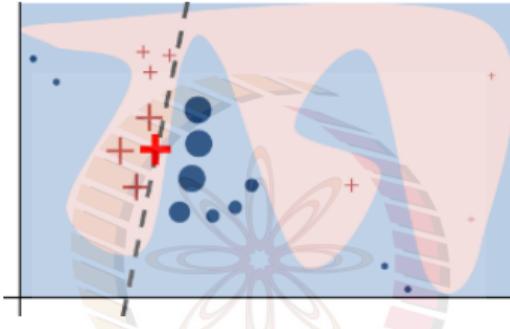
LIME: Local Interpretable Model-agnostic Explanations



- Given a point x , let $z' \in \mathcal{Z}$ be a point obtained by perturbing one dimension (or region) in x ; \mathcal{Z} is the set of perturbations
- $\pi_x(z')$: proximity measure between instances z' and x , e.g. $\pi_x(z') = \exp(-D(x, z')^2/\sigma^2)$
- $f : \mathbb{R}^d \rightarrow \mathbb{R}$: model being explained
- Build a sparse linear model, $g(z') = w_g \cdot z'$
- Learn w_g to minimize:

$$\mathcal{L}(f, g, \pi_x) = \sum_{z, z' \in \mathcal{Z}} \pi_x(z)(f(z) - g(z'))^2$$

LIME: Local Interpretable Model-agnostic Explanations



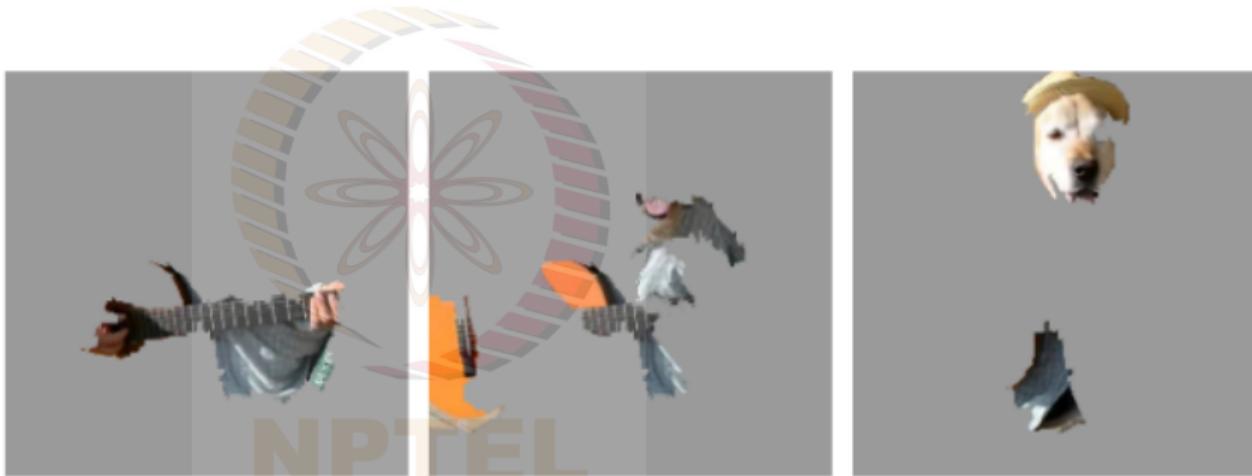
- Given a point x , let $z' \in \mathcal{Z}$ be a point obtained by perturbing one dimension (or region) in x ; \mathcal{Z} is the set of perturbations
- $\pi_x(z')$: proximity measure between instances z' and x , e.g. $\pi_x(z') = \exp(-D(x, z')^2/\sigma^2)$
- $f : \mathbb{R}^d \rightarrow \mathbb{R}$: model being explained
- Build a sparse linear model, $g(z') = w_g \cdot z'$
- Learn w_g to minimize:

$$\mathcal{L}(f, g, \pi_x) = \sum_{z, z' \in \mathcal{Z}} \pi_x(z)(f(z) - g(z'))^2$$

LIME: Local Interpretable Model-agnostic Explanations



(a) Original Image



(b) Explaining *Electric guitar*

(c) Explaining *Acoustic guitar*

(d) Explaining *Labrador*

LIME: Fidelity-Interpretability Trade-off

Notations:

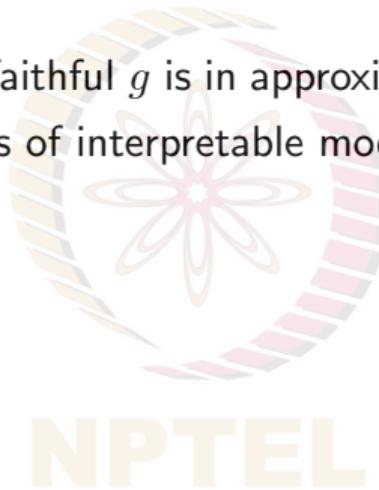
- $\mathcal{L}(f, g, \pi_x)$: measure of how unfaithful g is in approximating f in the locality defined by π_x



LIME: Fidelity-Interpretability Trade-off

Notations:

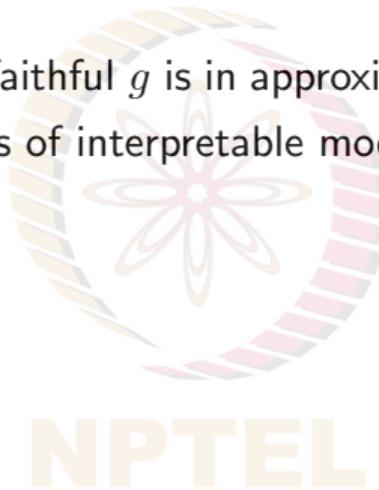
- $\mathcal{L}(f, g, \pi_x)$: measure of how unfaithful g is in approximating f in the locality defined by π_x
- $g \in G$: model belonging to class of interpretable models, e.g. linear model $g(z') = w_g.z'$



LIME: Fidelity-Interpretability Trade-off

Notations:

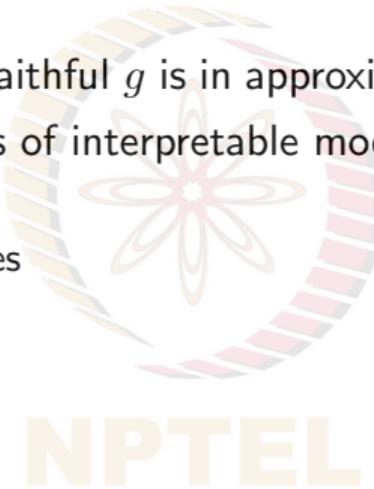
- $\mathcal{L}(f, g, \pi_x)$: measure of how unfaithful g is in approximating f in the locality defined by π_x
- $g \in G$: model belonging to class of interpretable models, e.g. linear model $g(z') = w_g.z'$
- $\Omega(g)$: Complexity of g model



LIME: Fidelity-Interpretability Trade-off

Notations:

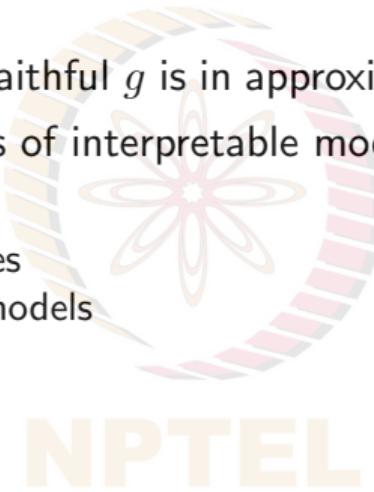
- $\mathcal{L}(f, g, \pi_x)$: measure of how unfaithful g is in approximating f in the locality defined by π_x
- $g \in G$: model belonging to class of interpretable models, e.g. linear model $g(z') = w_g.z'$
- $\Omega(g)$: Complexity of g model
 - Depth of trees in decision trees



LIME: Fidelity-Interpretability Trade-off

Notations:

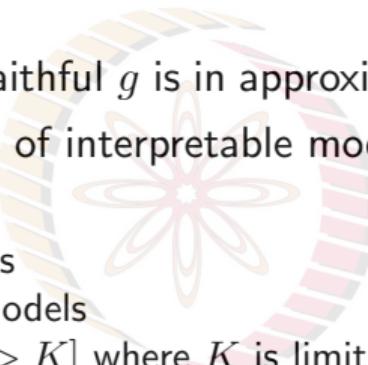
- $\mathcal{L}(f, g, \pi_x)$: measure of how unfaithful g is in approximating f in the locality defined by π_x
- $g \in G$: model belonging to class of interpretable models, e.g. linear model $g(z') = w_g \cdot z'$
- $\Omega(g)$: Complexity of g model
 - Depth of trees in decision trees
 - Number of weights in linear models



LIME: Fidelity-Interpretability Trade-off

Notations:

- $\mathcal{L}(f, g, \pi_x)$: measure of how unfaithful g is in approximating f in the locality defined by π_x
- $g \in G$: model belonging to class of interpretable models, e.g. linear model $g(z') = w_g \cdot z'$
- $\Omega(g)$: Complexity of g model
 - Depth of trees in decision trees
 - Number of weights in linear models
 - For images, $\Omega(g) = \mathbb{1}[||w_g||_0 > K]$ where K is limit on number of super-pixels



NPTEL

LIME: Fidelity-Interpretability Trade-off

Notations:

- $\mathcal{L}(f, g, \pi_x)$: measure of how unfaithful g is in approximating f in the locality defined by π_x
- $g \in G$: model belonging to class of interpretable models, e.g. linear model $g(z') = w_g \cdot z'$
- $\Omega(g)$: Complexity of g model
 - Depth of trees in decision trees
 - Number of weights in linear models
 - For images, $\Omega(g) = \mathbb{1}[||w_g||_0 > K]$ where K is limit on number of super-pixels
- **LIME explanation** obtained as a trade-off:

$$\varepsilon(x) = \arg \min_g \mathcal{L}(f, g, \pi_x) + \Omega(g)$$

SHAP⁸

- Inspired from Shapley values in game theory
- let N : Total number of features; v : Value function that assigns a real number to any coalition $S \subseteq N$; and $\phi_v(i)$: Attribution score for feature i



⁸Lundberg et al, A Unified Approach to Interpreting Model Predictions; NeurIPS 2017

SHAP⁸

- Inspired from Shapley values in game theory
- let N : Total number of features; v : Value function that assigns a real number to any coalition $S \subseteq N$; and $\phi_v(i)$: Attribution score for feature i
- **Attribution score:** Marginal contribution that player (in our case, feature) i makes upon joining the team, averaged over all orders in which team can be formed

$$\phi_v(i) = \sum_{S \subseteq \{1, 2, \dots, N\} \setminus \{i\}} \frac{1}{N!} |S|!(N - |S| - 1)! \underbrace{(v(S \cup i) - v(S))}_{\text{Value of adding player } i \text{ to a coalition}}$$

NPTEL

⁸Lundberg et al, A Unified Approach to Interpreting Model Predictions; NeurIPS 2017

SHAP⁸

- Inspired from Shapley values in game theory
- let N : Total number of features; v : Value function that assigns a real number to any coalition $S \subseteq N$; and $\phi_v(i)$: Attribution score for feature i
- **Attribution score:** Marginal contribution that player (in our case, feature) i makes upon joining the team, averaged over all orders in which team can be formed

$$\phi_v(i) = \sum_{S \subseteq \{1, 2, \dots, N\} \setminus \{i\}} \frac{1}{N!} |S|!(N - |S| - 1)! \underbrace{(v(S \cup i) - v(S))}_{\text{Value of adding player } i \text{ to a coalition}}$$

- With $f(x)$ as model prediction, we marginalize over out-of-coalition features $x_{\bar{S}}$ where $\bar{S} = \{1, 2, \dots, N\} \setminus S$ to get:

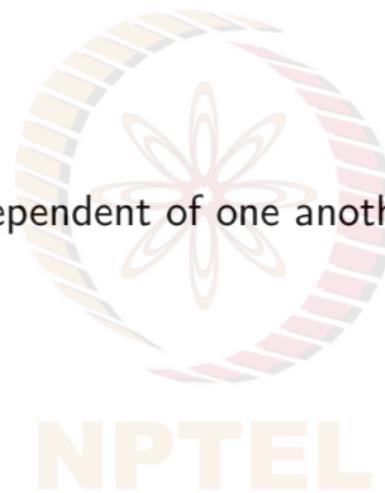
$$v(S) = \mathbb{E}_{p(x' | x_S)} [f(x_S \cup x'_{\bar{S}})]$$

- SHAP assumes features to be independent $\implies v(S) = \mathbb{E}_{p(x')} [f(x_S \cup x'_{\bar{S}})]$

⁸Lundberg et al, A Unified Approach to Interpreting Model Predictions; NeurIPS 2017

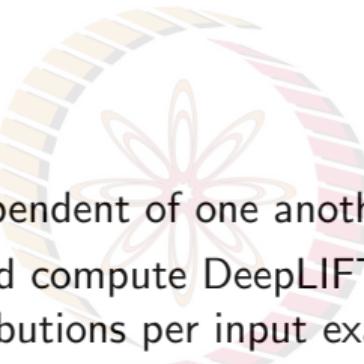
DeepSHAP

- Assumes input features are independent of one another and explanation model is linear



DeepSHAP

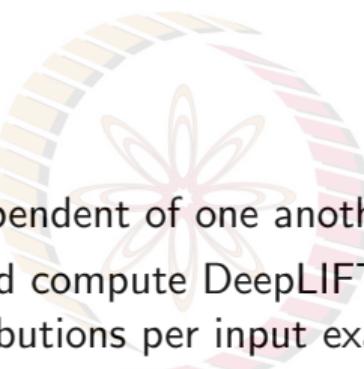
- Assumes input features are independent of one another and explanation model is linear
- Take distribution of baselines and compute DeepLIFT attribution for each input-baseline pair, then average resulting attributions per input example



NPTEL

DeepSHAP

- Assumes input features are independent of one another and explanation model is linear
- Take distribution of baselines and compute DeepLIFT attribution for each input-baseline pair, then average resulting attributions per input example



NPTEL

How to Evaluate Explanations?



⁹Melis et al, Towards Robust Interpretability with Self-Explaining Neural Networks, NeurIPS 2018

¹⁰Petsiuk et al, RISE: Randomized Input Sampling for Explanation of Black-box Models, BMVC 2018

How to Evaluate Explanations?

- IoU of thresholded salient region with ground truth bounding box (if available)



⁹Melis et al, Towards Robust Interpretability with Self-Explaining Neural Networks, NeurIPS 2018

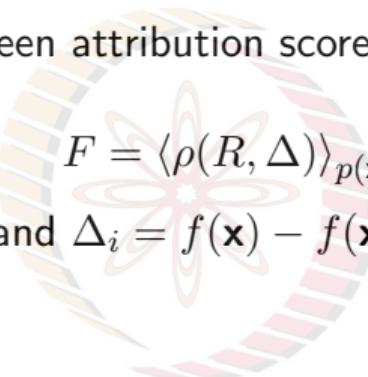
¹⁰Petsiuk et al, RISE: Randomized Input Sampling for Explanation of Black-box Models, BMVC 2018

How to Evaluate Explanations?

- IoU of thresholded salient region with ground truth bounding box (if available)
- **Faithfulness**⁹: Correlation between attribution scores and output differences on perturbation:

$$F = \langle \rho(R, \Delta) \rangle_{p(\mathbf{x})}$$

where R_i is relevance of pixel i and $\Delta_i = f(\mathbf{x}) - f(\mathbf{x}_i)$ where \mathbf{x}_i is image obtained after perturbing pixel i



⁹Melis et al, Towards Robust Interpretability with Self-Explaining Neural Networks, NeurIPS 2018

¹⁰Petsiuk et al, RISE: Randomized Input Sampling for Explanation of Black-box Models, BMVC 2018

How to Evaluate Explanations?

- IoU of thresholded salient region with ground truth bounding box (if available)
- **Faithfulness**⁹: Correlation between attribution scores and output differences on perturbation:

$$F = \langle \rho(R, \Delta) \rangle_{p(\mathbf{x})}$$

where R_i is relevance of pixel i and $\Delta_i = f(\mathbf{x}) - f(\mathbf{x}_i)$ where \mathbf{x}_i is image obtained after perturbing pixel i

- **Causal Metric (Deletion Metric)**¹⁰:

- ① Delete pixels sequentially, most relevant first



⁹Melis et al, Towards Robust Interpretability with Self-Explaining Neural Networks, NeurIPS 2018

¹⁰Petsiuk et al, RISE: Randomized Input Sampling for Explanation of Black-box Models, BMVC 2018

How to Evaluate Explanations?

- IoU of thresholded salient region with ground truth bounding box (if available)
- **Faithfulness**⁹: Correlation between attribution scores and output differences on perturbation:

$$F = \langle \rho(R, \Delta) \rangle_{p(\mathbf{x})}$$

where R_i is relevance of pixel i and $\Delta_i = f(\mathbf{x}) - f(\mathbf{x}_i)$ where \mathbf{x}_i is image obtained after perturbing pixel i

- **Causal Metric (Deletion Metric)**¹⁰:

- ① Delete pixels sequentially, most relevant first
- ② Compute AUC of network's output as function of perturbed inputs vs amount of perturbation; lesser AUC better

⁹Melis et al, Towards Robust Interpretability with Self-Explaining Neural Networks, NeurIPS 2018

¹⁰Petsiuk et al, RISE: Randomized Input Sampling for Explanation of Black-box Models, BMVC 2018

How to Evaluate Explanations?

- IoU of thresholded salient region with ground truth bounding box (if available)
- **Faithfulness**⁹: Correlation between attribution scores and output differences on perturbation:

$$F = \langle \rho(R, \Delta) \rangle_{p(\mathbf{x})}$$

where R_i is relevance of pixel i and $\Delta_i = f(\mathbf{x}) - f(\mathbf{x}_i)$ where \mathbf{x}_i is image obtained after perturbing pixel i

- **Causal Metric (Deletion Metric)**¹⁰:

- ① Delete pixels sequentially, most relevant first
- ② Compute AUC of network's output as function of perturbed inputs vs amount of perturbation; lesser AUC better

Similarly, **Insertion Metric** inserts pixels sequentially, least relevant first; higher AUC better

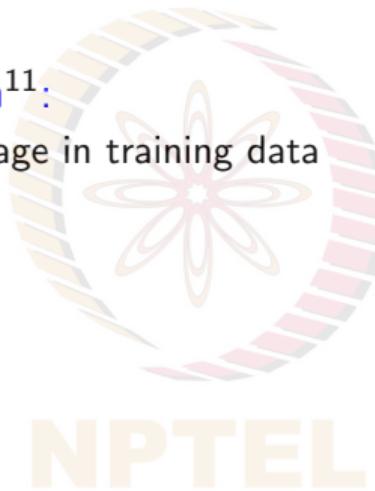
⁹Melis et al, Towards Robust Interpretability with Self-Explaining Neural Networks, NeurIPS 2018

¹⁰Petsiuk et al, RISE: Randomized Input Sampling for Explanation of Black-box Models, BMVC 2018

How to Evaluate Explanations?

- ROAR: RemOve And Retrain¹¹:

- ① Get saliency map for each image in training data



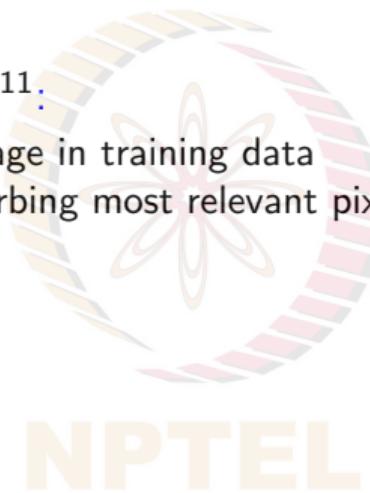
¹¹Hooker et al, A Benchmark for Interpretability Methods in Deep Neural Networks, NeurIPS 2019

¹²Adebayo et al, Sanity Checks for Saliency Maps, NeurIPS 2018

¹³Sundararajan et al, Axiomatic Attribution for Deep Networks, ICML 2017

How to Evaluate Explanations?

- ROAR: RemOve And Retrain¹¹:
 - ① Get saliency map for each image in training data
 - ② Retrain the model after perturbing most relevant pixels



¹¹Hooker et al, A Benchmark for Interpretability Methods in Deep Neural Networks, NeurIPS 2019

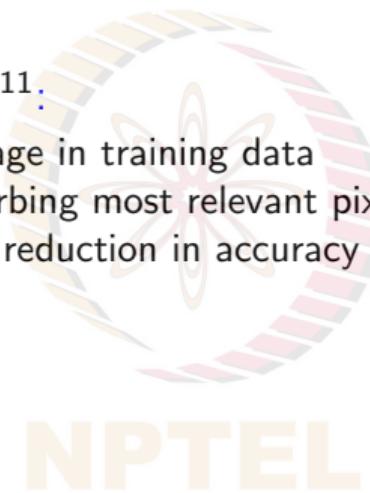
¹²Adebayo et al, Sanity Checks for Saliency Maps, NeurIPS 2018

¹³Sundararajan et al, Axiomatic Attribution for Deep Networks, ICML 2017

How to Evaluate Explanations?

- **ROAR: RemOve And Retrain¹¹:**

- ① Get saliency map for each image in training data
- ② Retrain the model after perturbing most relevant pixels
- ③ New model should have large reduction in accuracy



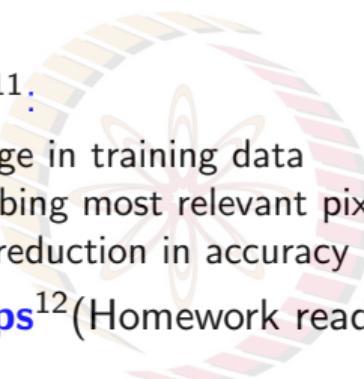
¹¹Hooker et al, A Benchmark for Interpretability Methods in Deep Neural Networks, NeurIPS 2019

¹²Adebayo et al, Sanity Checks for Saliency Maps, NeurIPS 2018

¹³Sundararajan et al, Axiomatic Attribution for Deep Networks, ICML 2017

How to Evaluate Explanations?

- ROAR: RemOve And Retrain¹¹:
 - ① Get saliency map for each image in training data
 - ② Retrain the model after perturbing most relevant pixels
 - ③ New model should have large reduction in accuracy
- Sanity checks for saliency maps¹²(Homework reading!)



NPTEL

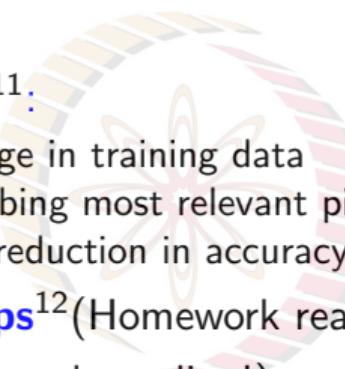
¹¹Hooker et al, A Benchmark for Interpretability Methods in Deep Neural Networks, NeurIPS 2019

¹²Adebayo et al, Sanity Checks for Saliency Maps, NeurIPS 2018

¹³Sundararajan et al, Axiomatic Attribution for Deep Networks, ICML 2017

How to Evaluate Explanations?

- ROAR: RemOve And Retrain¹¹:
 - ① Get saliency map for each image in training data
 - ② Retrain the model after perturbing most relevant pixels
 - ③ New model should have large reduction in accuracy
- Sanity checks for saliency maps¹²(Homework reading!)
- Axioms for attribution¹³(Homework reading!)



¹¹Hooker et al, A Benchmark for Interpretability Methods in Deep Neural Networks, NeurIPS 2019

¹²Adebayo et al, Sanity Checks for Saliency Maps, NeurIPS 2018

¹³Sundararajan et al, Axiomatic Attribution for Deep Networks, ICML 2017

Summary

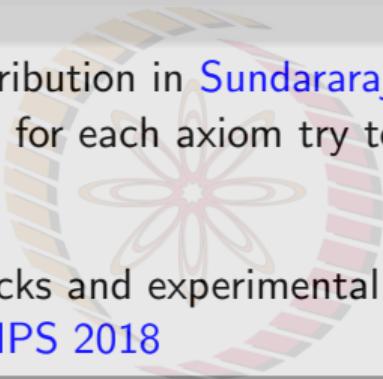
- Both DeepLIFT and Integrated Gradients overcome saturating gradients problem; although DeepLIFT is usually faster, it violates Implementation Invariance axiom¹⁴ (one of the axioms for homework reading!) due to use of discrete gradients
- Smooth Integrated Gradients may be preferred over Integrated Gradients when sparsity is desired
- For better interpretability in terms of visual coherence, XRAI is good choice whose mask is composed of relevant segments rather than pixels
- LIME is model-agnostic and can be used for image, text as well as tabular data but is slow and appears inconsistent between runs
- SHAP has strong game-theoretic background but needs approximations for real world experiments

¹⁴Sundararajan et al, Axiomatic Attribution for Deep Networks, ICML 2017

Homework

Reading

- Go through list of axioms of attribution in [Sundararajan et al, Axiomatic Attribution for Deep Networks, ICML 2017](#) and for each axiom try to identify the attribution algorithms that satisfy that
- Go through proposed sanity checks and experimental findings in [Adebayo et al, Sanity Checks for Saliency Maps, NeurIPS 2018](#)

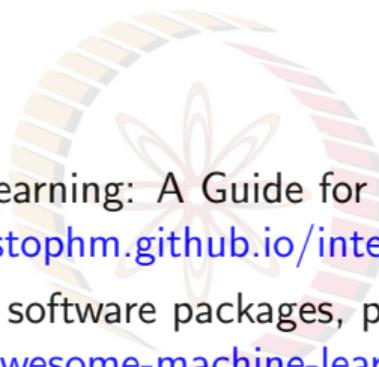


Programming

NPTEL

- Play with [Captum](#): A popular library for model interpretation by Facebook Open Source
- Try visualizing your models through the lens of [OpenAI Microscope](#)

Extra Resources



- Molnar, Interpretable machine learning: A Guide for Making Black Box Models Explainable, 2019: <https://christophm.github.io/interpretable-ml-book/>.
- For a collection of tutorials and software packages, please refer:
<https://github.com/jphall663/awesome-machine-learning-interpretability>

NPTEL

References I



Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. "Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps". In: *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Workshop Track Proceedings*. Ed. by Yoshua Bengio and Yann LeCun. 2014.



Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "'Why Should I Trust You?': Explaining the Predictions of Any Classifier". In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*. 2016, pp. 1135–1144.



Scott M Lundberg and Su-In Lee. "A Unified Approach to Interpreting Model Predictions". In: *Advances in Neural Information Processing Systems 30*. Ed. by I. Guyon et al. Curran Associates, Inc., 2017, pp. 4765–4774.

References II



Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. "Learning Important Features Through Propagating Activation Differences". In: *Proceedings of the 34th International Conference on Machine Learning*. Ed. by Doina Precup and Yee Whye Teh. Vol. 70. Proceedings of Machine Learning Research. International Convention Centre, Sydney, Australia: PMLR, 2017, pp. 3145–3153.



D. Smilkov et al. "SmoothGrad: removing noise by adding noise". In: *ICML workshop on visualization for deep learning* (June 2017). arXiv: [1706.03825 \[cs.LG\]](https://arxiv.org/abs/1706.03825).



Mukund Sundararajan, Ankur Taly, and Qiqi Yan. "Axiomatic Attribution for Deep Networks". In: *Proceedings of the 34th International Conference on Machine Learning*. Ed. by Doina Precup and Yee Whye Teh. Vol. 70. Proceedings of Machine Learning Research. International Convention Centre, Sydney, Australia: PMLR, 2017, pp. 3319–3328.



Julius Adebayo et al. "Sanity Checks for Saliency Maps". In: *Advances in Neural Information Processing Systems 31*. Ed. by S. Bengio et al. Curran Associates, Inc., 2018, pp. 9505–9515.

References III

-  David Alvarez Melis and Tommi Jaakkola. "Towards Robust Interpretability with Self-Explaining Neural Networks". In: *Advances in Neural Information Processing Systems 31*. Ed. by S. Bengio et al. Curran Associates, Inc., 2018, pp. 7775–7784.
-  Vitali Petsiuk, Abir Das, and Kate Saenko. "RISE: Randomized Input Sampling for Explanation of Black-box Models". In: *Proceedings of the British Machine Vision Conference (BMVC)*. 2018.
-  Sara Hooker et al. "A Benchmark for Interpretability Methods in Deep Neural Networks". In: *Advances in Neural Information Processing Systems 32*. Ed. by H. Wallach et al. Curran Associates, Inc., 2019, pp. 9737–9748.
-  Andrei Kapishnikov et al. "XRAI: Better Attributions Through Regions". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 2019.
- <http://www.unofficialgoogledatascience.com/2017/03/attributing-deep-networks-prediction-to.html>.