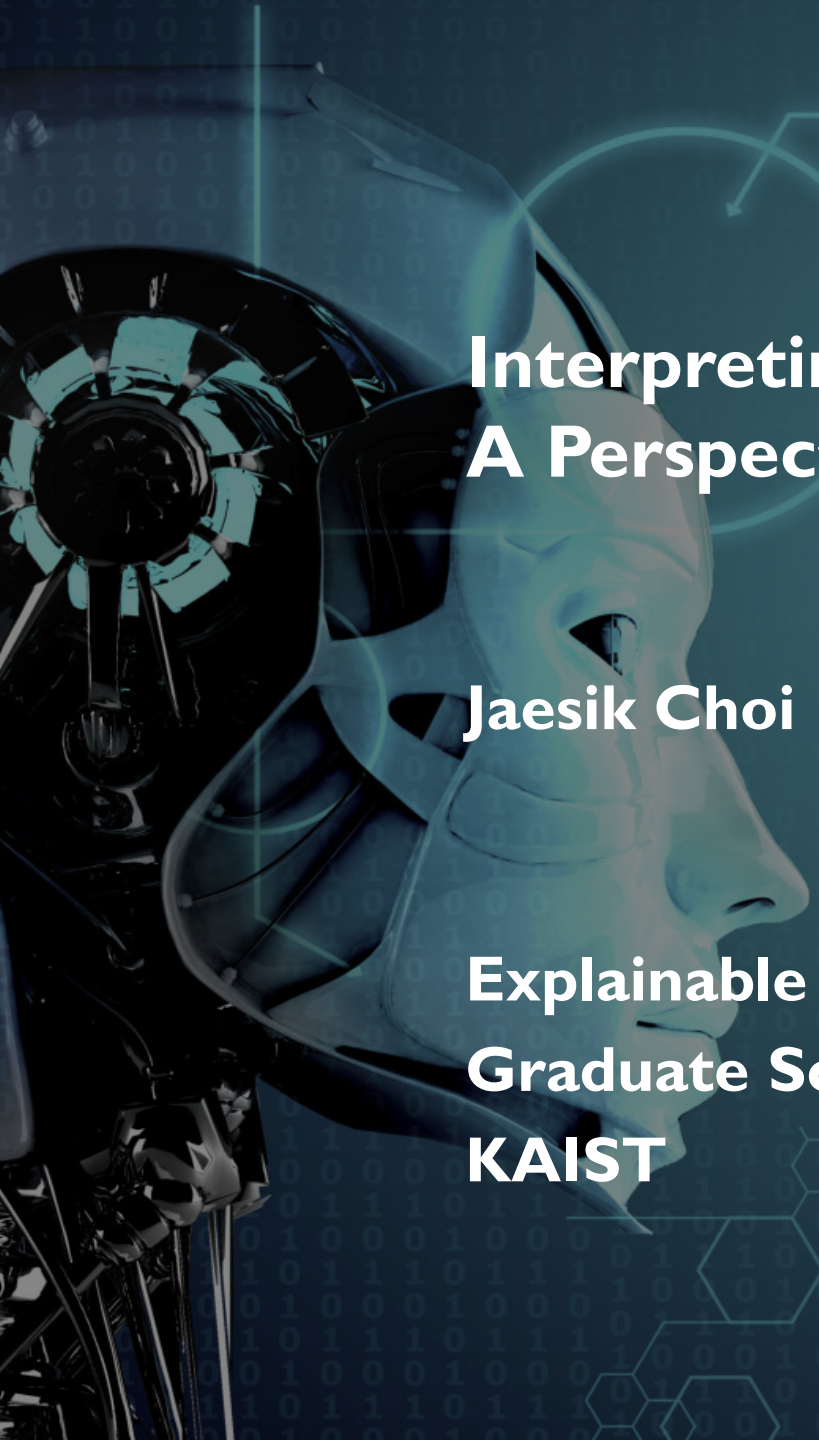


Interpreting and Explaining Deep Neural Networks: A Perspective on Time Series Data – Part 1/3

Jaesik Choi

Explainable Artificial Intelligence Center
Graduate School of Artificial Intelligence
KAIST

Interpreting and Explaining Deep Neural Networks: A Perspective on Time Series Data

Agenda (150 min)

Overview to Explainable Artificial Intelligence (XAI) – 15 min

- Biases in AI systems
- General Data Protection Regulation (GDPR)
- Categories of XAI algorithms

Input Attributions Methods for Deep Neural Networks – 35 min

[10 min break]

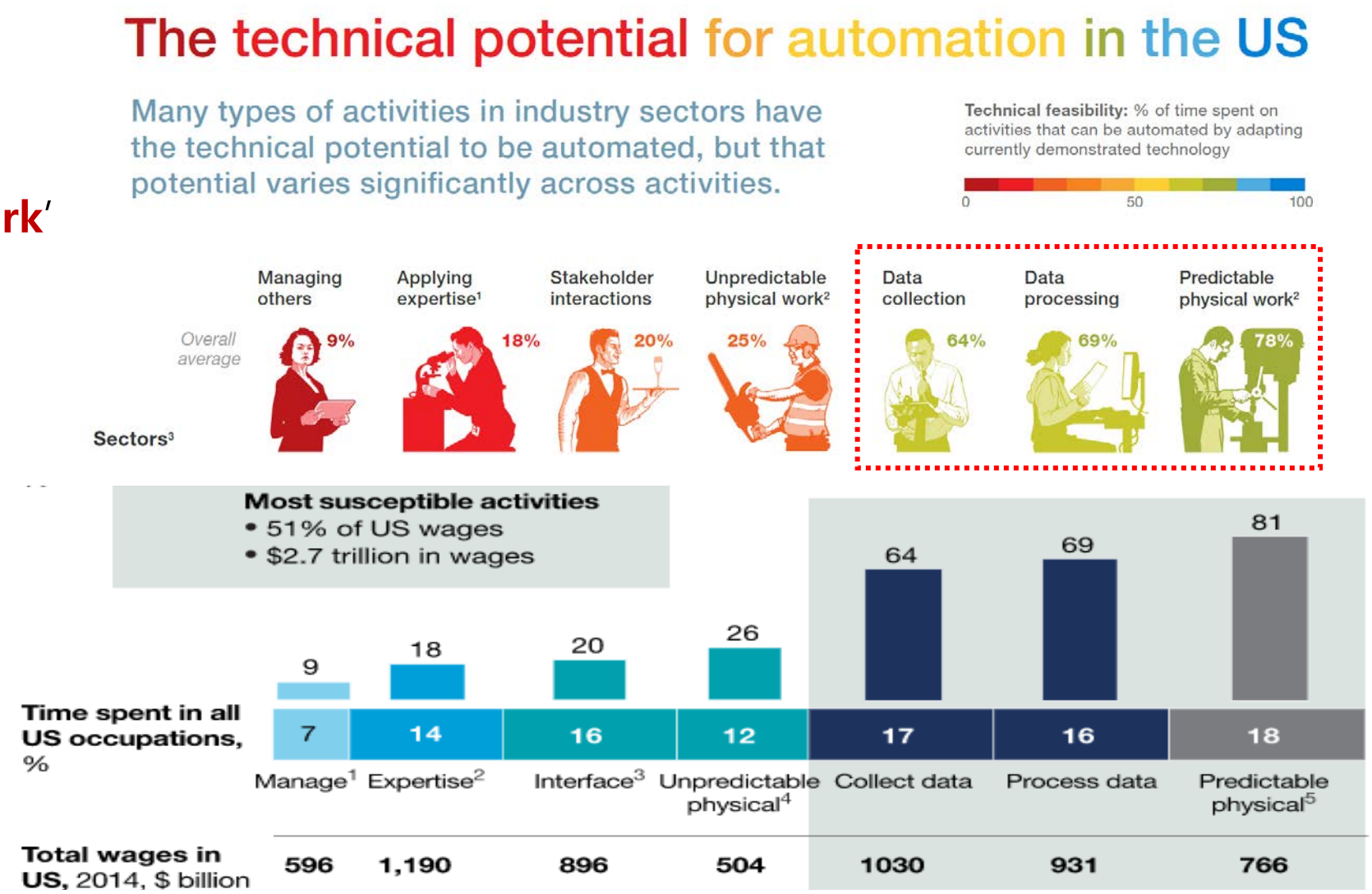
Interpreting Inside of Deep Neural Networks – 50 min

[10 min break]

Explainable Models for Time Series Data – 50 min

In 2025,
estimated economic impact of
'**Automation of Knowledge work**'
may reach up to
6.7 trillion US dollar.

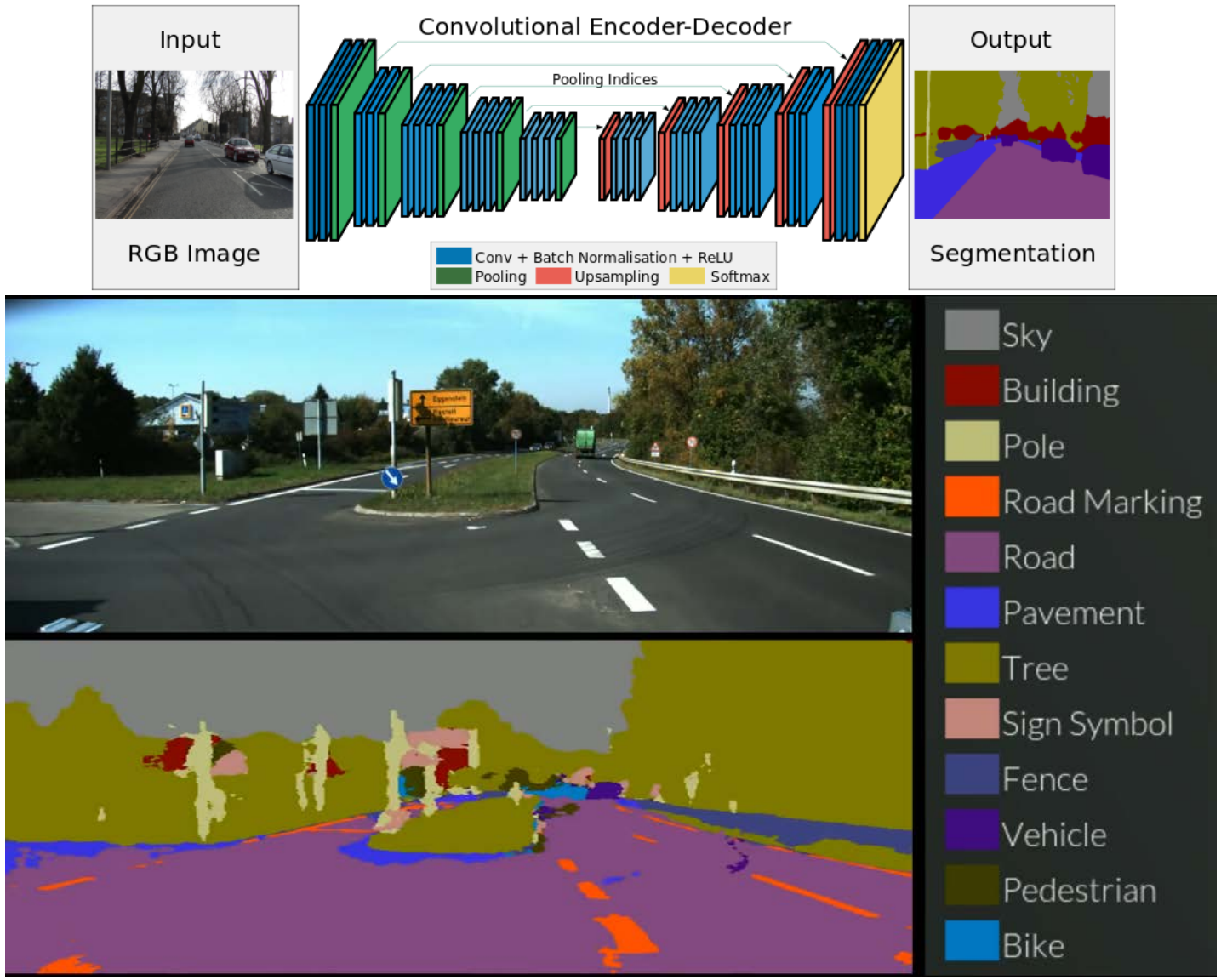
In US,
51% of US wages or
$2.7 trillion in wages
could be automated.



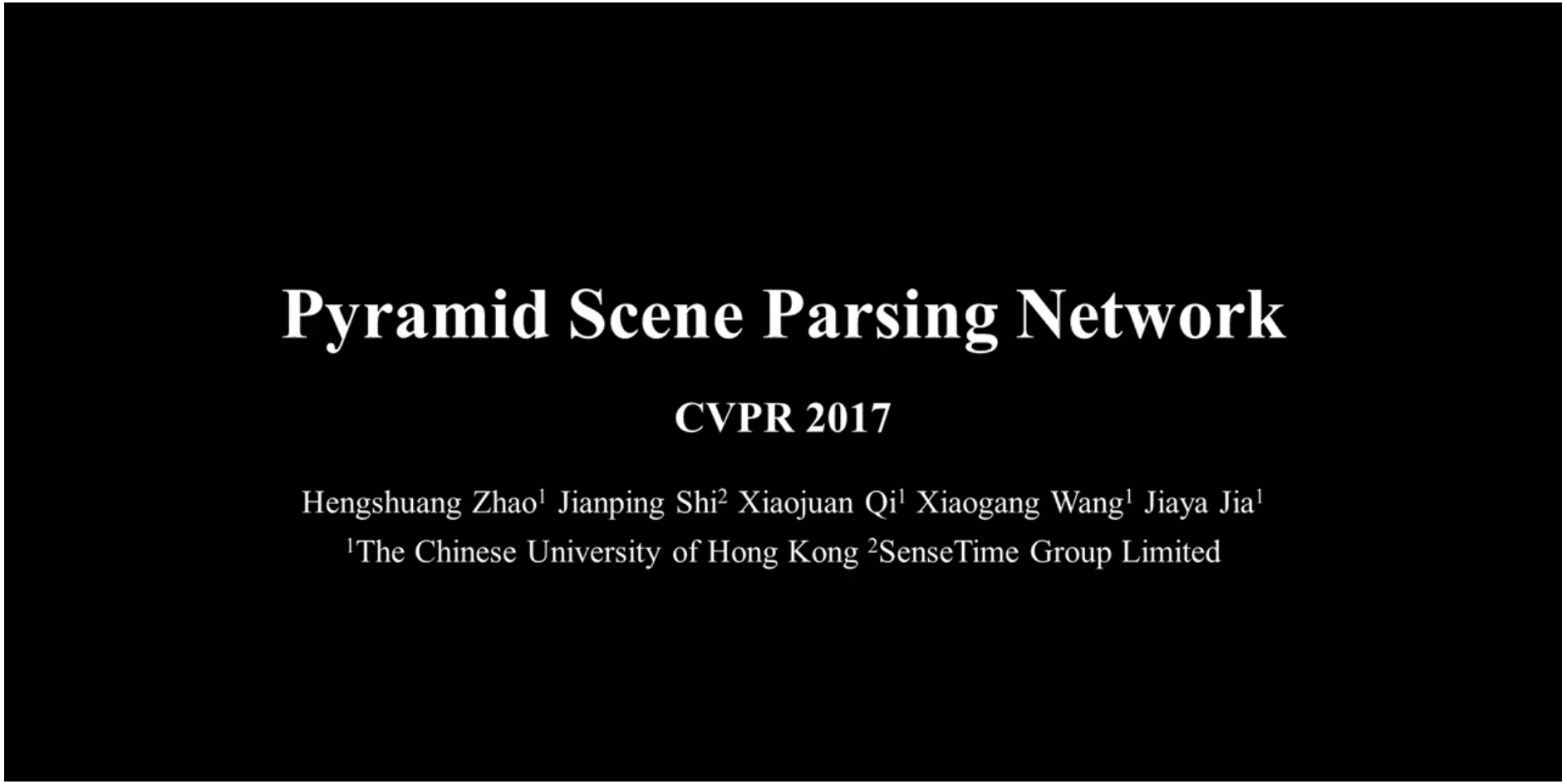
Automation of Knowledge Work [McKinsey 2013]

DARPA Grand Challenge 2005

Say Hello to Waymo 2016



Semantic Segmentation by SegNet 2015



Semantic Segmentation by Pyramid Scene Parsing Network 2017

Uber’s first car accident - Death of Elaine Herzberg

Uber's self-driving car killed a pedestrian (Marc 18th, 2018)
The ‘safety driver’ was watching a TV show (June 22th, 2018)

Many, complex AI systems are not transparent to see the mechanisms inside!

Do We Understand AI Systems Enough?

COMPAS: Prediction of Crime

Prior Offense	1 attempted burglary	1 resisting arrest without violence
COMPAS' decision		
Subsequent Offenses	3 drug possessions	None

AI algorithms are exposed to

(1) data bias,
(2) model bias, and
(3) algorithmic bias

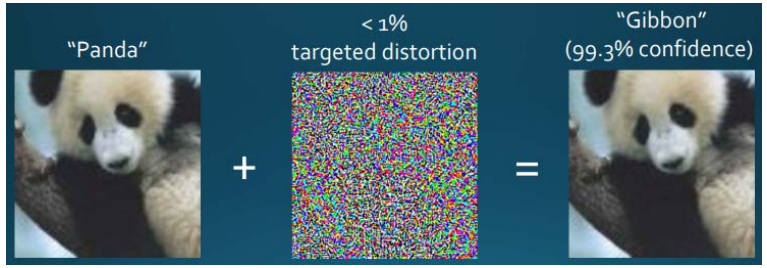
Do We Understand AI Systems Enough?

Article	Contents
13-14. **Right to explanation**	A data subject has the right to "**meaningful information about the logic involved**" when decision is made automatically.
EU administration	When violated **4% of global revenue** will be fined.
Enact	**May 28th, 2018**

EU General Data Protection Regulation (GDPR)

DESCRIBE	CATEGORIZE	EXPLAIN
Handcrafted Knowledge	*Statistical Learning*	*Contextual Adaptation*

Statistically impressive, but individually unreliable

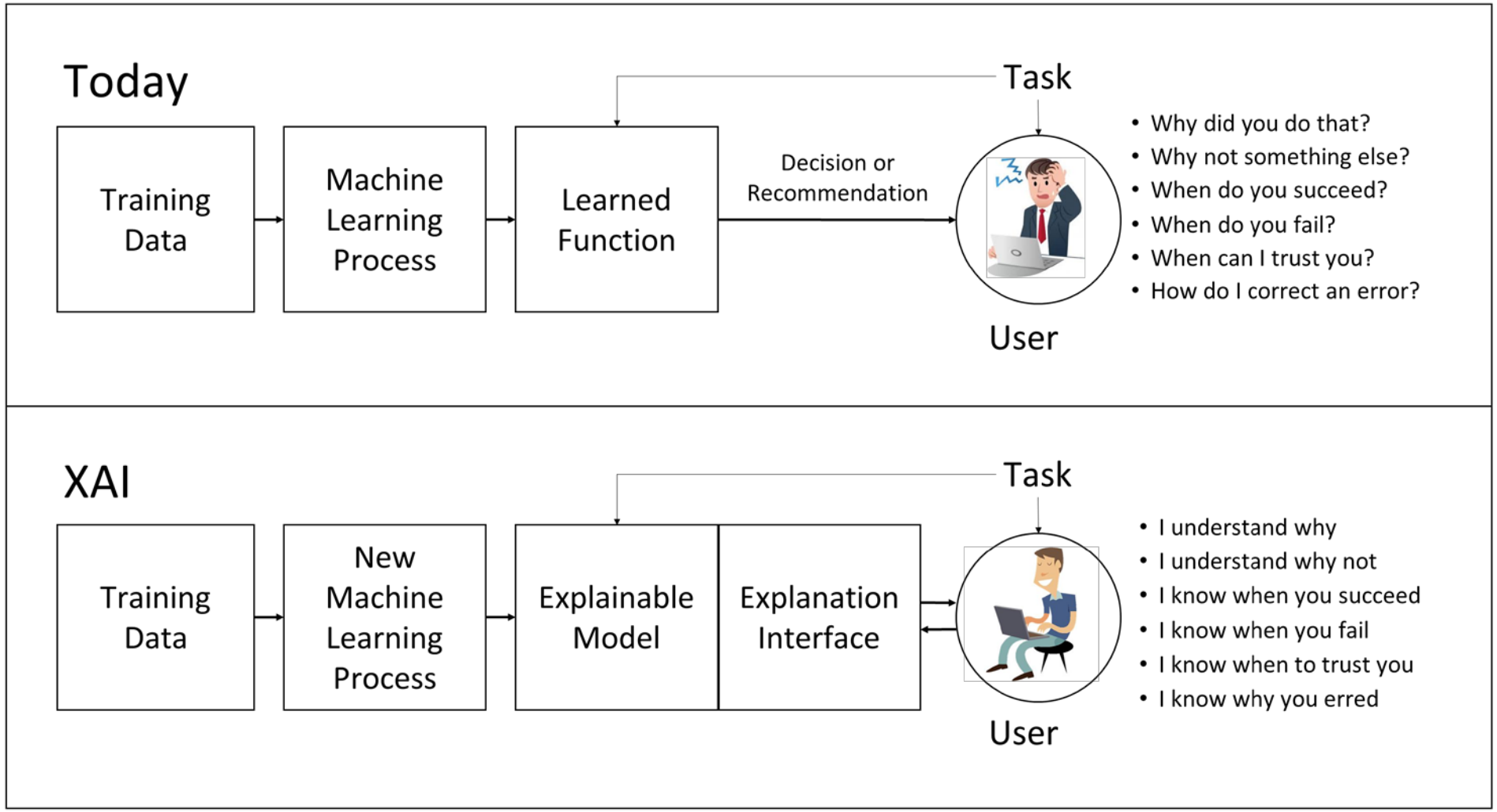


Inherent flaws can be exploited

Skewed training data creates Maladaptation

DARPA

A DARPA Perspective on AI – Three Waves of AI

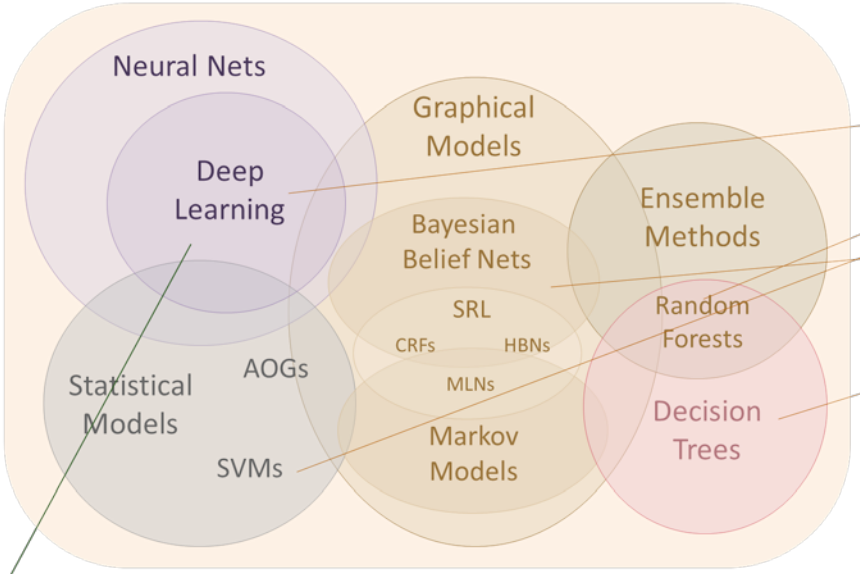


Explainable AI – Performance vs. Explainability

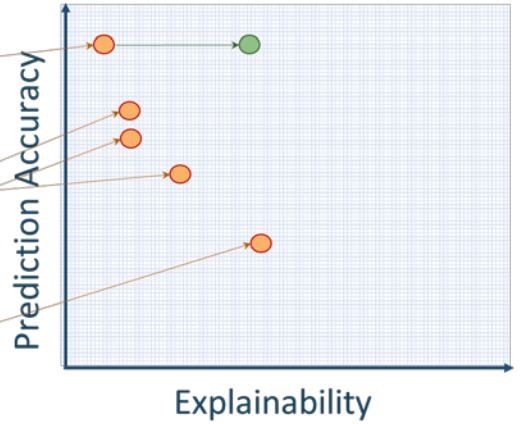
New Approach

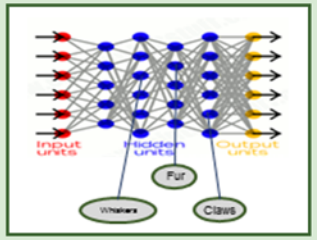
Create a suite of machine learning techniques that produce more explainable models, while maintaining a high level of learning performance

Learning Techniques (today)



Explainability (notional)





Deep Explanation

Modified deep learning techniques to learn explainable features

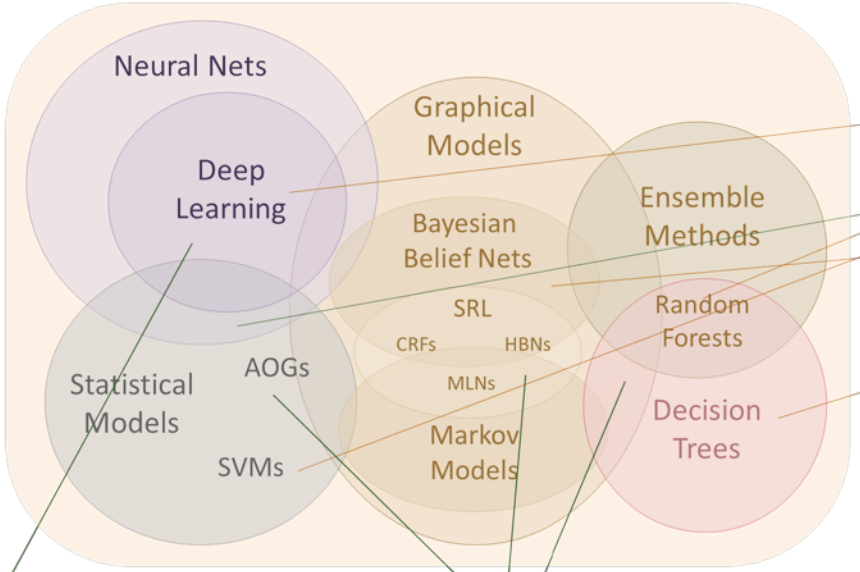
DARPA

Explainable AI – Performance vs. Explainability

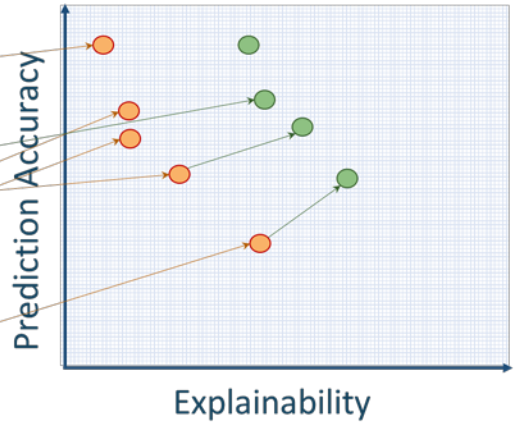
New Approach

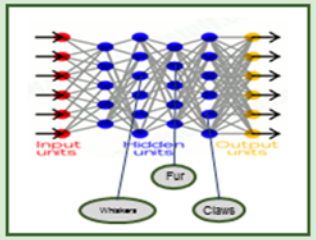
Create a suite of machine learning techniques that produce more explainable models, while maintaining a high level of learning performance

Learning Techniques (today)



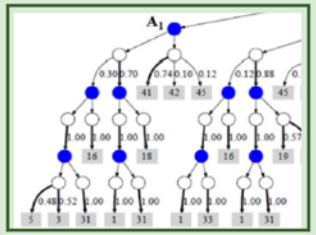
Explainability (notional)





Deep Explanation

Modified deep learning techniques to learn explainable features



Interpretable Models

Techniques to learn more structured, interpretable, causal models

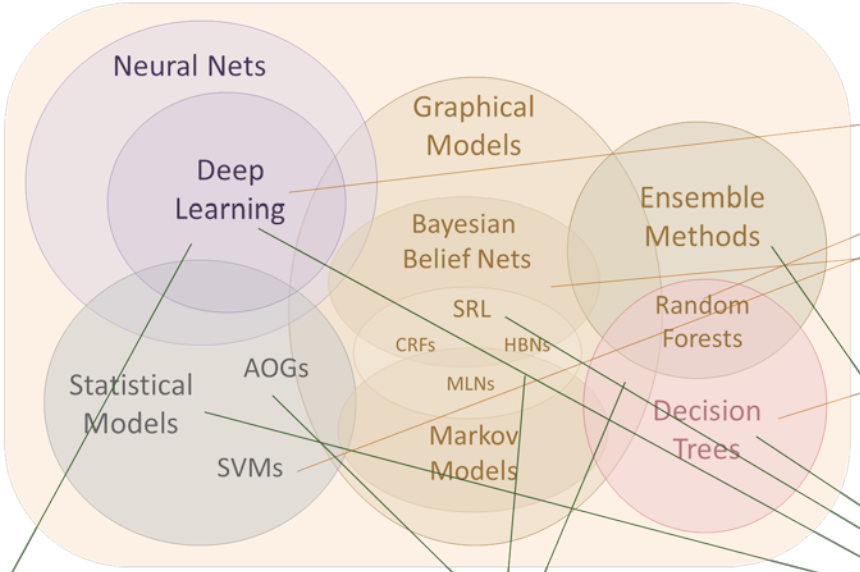
DARPA

Explainable AI – Performance vs. Explainability

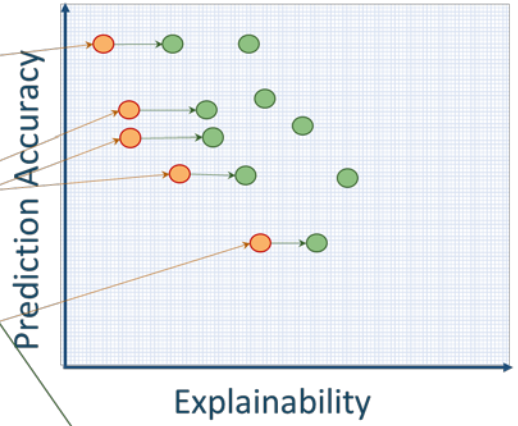
New Approach

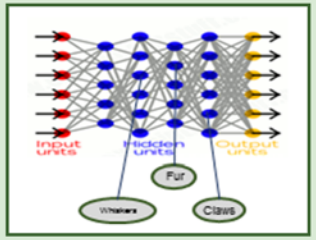
Create a suite of machine learning techniques that produce more explainable models, while maintaining a high level of learning performance

Learning Techniques (today)



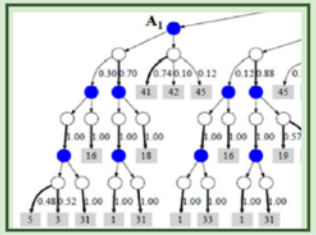
Explainability (notional)





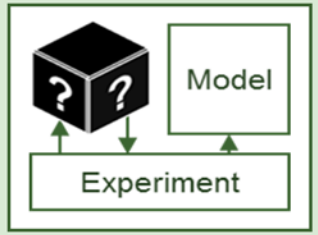
Deep Explanation

Modified deep learning techniques to learn explainable features



Interpretable Models

Techniques to learn more structured, interpretable, causal models



Model Induction

Techniques to infer an explainable model from any model as a black box

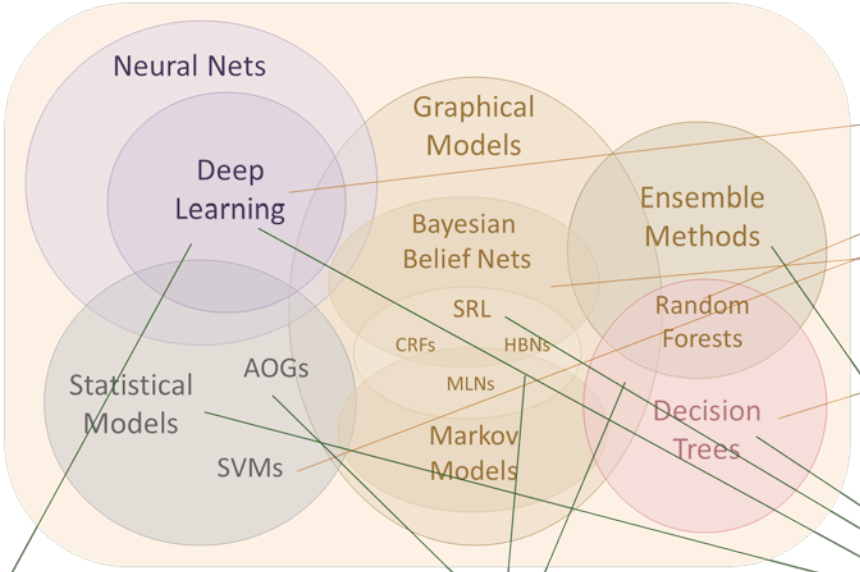
DARPA

Explainable AI – Performance vs. Explainability

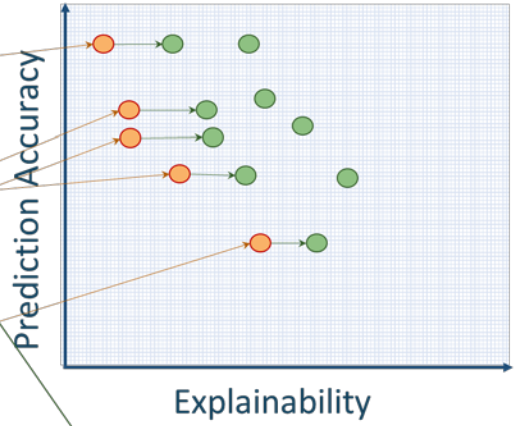
New Approach

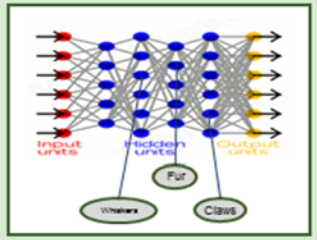
Create a suite of machine learning techniques that produce more explainable models, while maintaining a high level of learning performance

Learning Techniques (today)



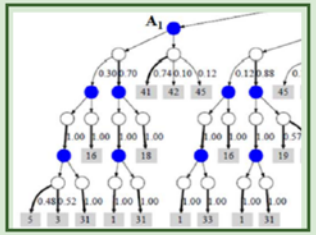
Explainability (notional)





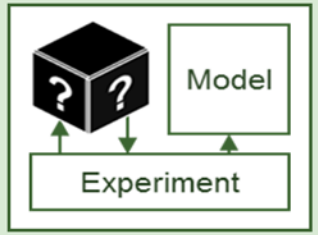
Deep Explanation

Modified deep learning techniques to learn explainable features



Interpretable Models

Techniques to learn more structured, interpretable, causal models

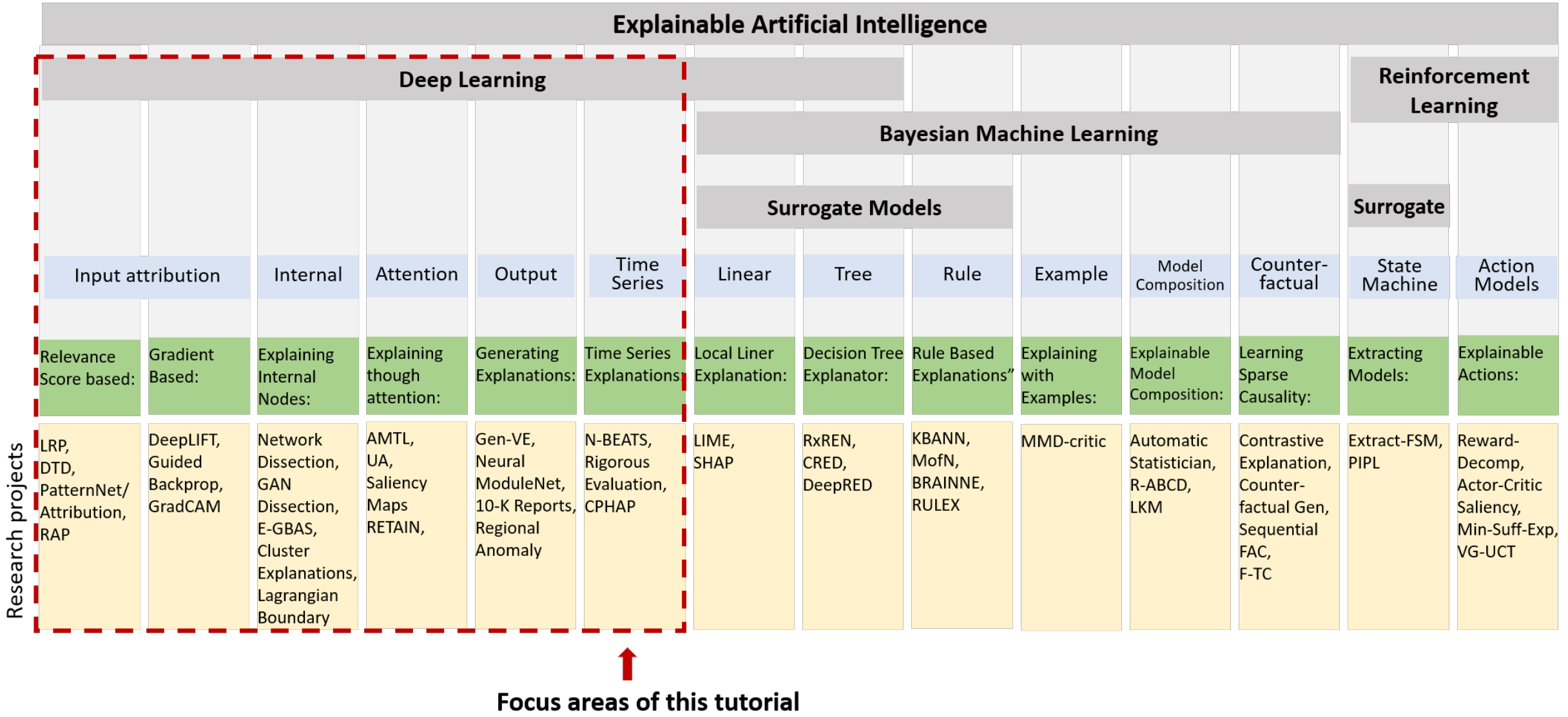


Model Induction

Techniques to infer an explainable model from any model as a black box

DARPA

Explainable AI – Performance vs. Explainability



A roadmap of Explainable Artificial Intelligence

Interpreting and Explaining Deep Neural Networks: A Perspective on Time Series Data

Agenda (150 min)

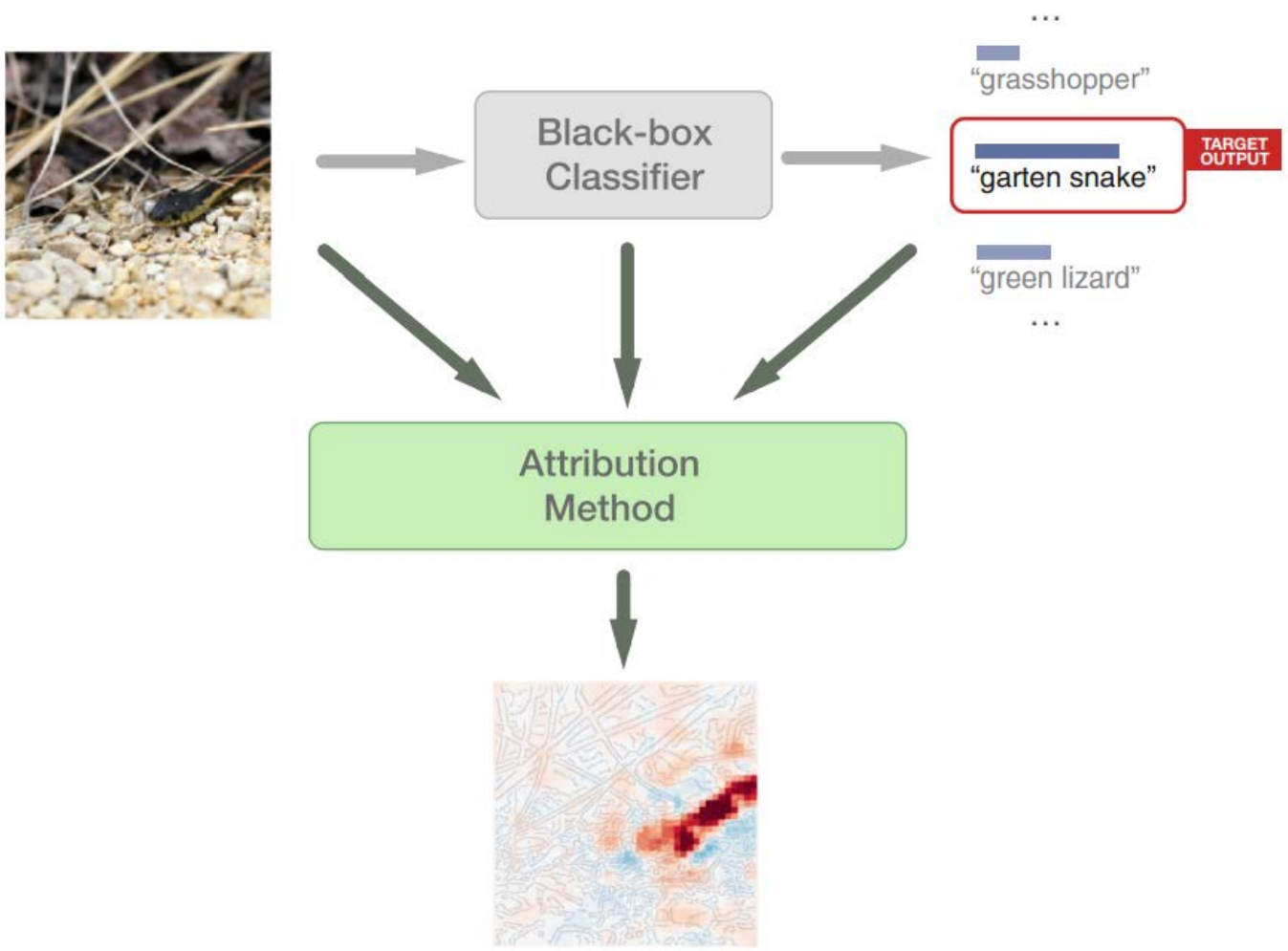
Overview to Explainable Artificial Intelligence (XAI) – 15 min

Input Attributions Methods for Deep Neural Networks – 35 min

- **Properties of Good Attribution Methods**
- **Relevance Score Based Methods: Layer-wise Relevance Propagation (LRP),**
- **Gradient Based Methods: DeepLIFT,**
- **Equivalence of LRP and DeepLIFT**
- **Handling Negative Relevance Scores**
- **Relative Attributing Propagation**

Interpreting Inside of Deep Neural Networks – 50 min

Explainable Models for Time Series Data – 50 min



[Image courtesy of Ancona Marco]

An Example: General Setting for Attribution Methods

☐ Model

- Input: N-dimensional one $x = [x_1, ..., x_N] \in \mathbb{R}^N$
- Output: C-dimensional one $S(x) = [S_1(x), ..., S_C(x)] \in \mathbb{R}^C$
- An attribution value (or relevance/contribution) of each input feature for a class c

$$R^c = [R_1^c, ..., R_N^c] \in \mathbb{R}^N$$

Definition: Input Attribution Toward an Output

☐ Linear Regression

$$y = w_0 + w_1 x_1 + ... + w_N x_N + \epsilon$$

☐ Example

- y_c: the future capital asset
- x_1 and x_2: two investments

$$\mathbb{E}[y_c | x_1, x_2] = 1.05 x_1 + 1.50 x_2$$

- The influence of the independent variables of the target

$$R_1(x) = 1.05 \quad R_2(x) = 1.50$$

- In fact, the attribution is the model gradient:

$$R_i(x) = \frac{\partial y_c}{\partial x_i}(x)$$

Input Attribution of Linear Models

- The influence of the independent variables of the target

$$\mathbb{E}[y_c|x_1, x_2] = 1.05x_1 + 1.50x_2$$

- However, when there are two different inputs:

x_1 = \$100,000, x_2 = \$10,000

y_c = 1.05 * \$100,000 + 1.50 * \$10,000

= \$105,000 + \$15,000

$$R_1(x) = 105'000 \quad R_2(x) = 15'000$$

- We can compute the attributions as the gradient multiplied element-wise by the input:

$$R_i(x) = x_i \cdot \frac{\partial y_c}{\partial x_i}(x)$$

Attribution of Linear Models

☐ Explanation Continuity

- An attribution methods satisfies explanation continuity if
  - Given a continuous prediction function $S_c(x)$, it produces continuous attributions $R^c(x)$.
- That is, for two nearly identical data points, the model responses are nearly identical, then its explanations are.

Properties for Good Attribution Methods: Explanation Continuity

☐ Implementation Invariance

- m_1 and m_2: two functionally equivalent models
- For any x, the models produce the same output

$$\forall x : S_{m_1}(x) = S_{m_2}(x)$$

- An attribution methods is implementation invariant if it always produces identical attributions for m_1 and m_2.

$$\forall (m_1, m_2, x, c) : R^{c,m_1}(x) = R^{c,m_2}(x)$$

Properties for Good Attribution Methods: Implementation Invariance

☐ Sensitivity-n

- An attribution methods satisfies sensitivity-n when the sum of the attributions for any subset of n features is equal to the variation of the output S_c caused removing the features.
- When n features are selected $x_S = [x_1, ...x_n] \subseteq x$

$$\sum_{i=1}^{n} R_i^c(x) = S_c(x) - S_c(x \setminus x_S)$$

- When n= N, this property is the efficiency property in the Shapley value.

$$\sum_{i=0}^{N} R_i^c(x) = S_c(x) - S_c(\bar{x})$$

- That is,

$$\forall x, c : \sum_{i=1}^{N} R_i^c(x) = S_c(x)$$

Properties for Good Attribution Methods: Sensitivity-n

☐ Attribution methods in a linear model

$$R_i^c(x) = \frac{\partial S_c(x)}{\partial x_i} \qquad R_i^c(x) = x_i \cdot \frac{\partial S_c(x)}{\partial x_i}$$

☐ Sensitivity analysis

- Compute the absolute value of the partial derivative

$$R_i^c(x) = \left| \frac{\partial S_c(x)}{\partial x_i} \right|$$

☐ Gradient * Input

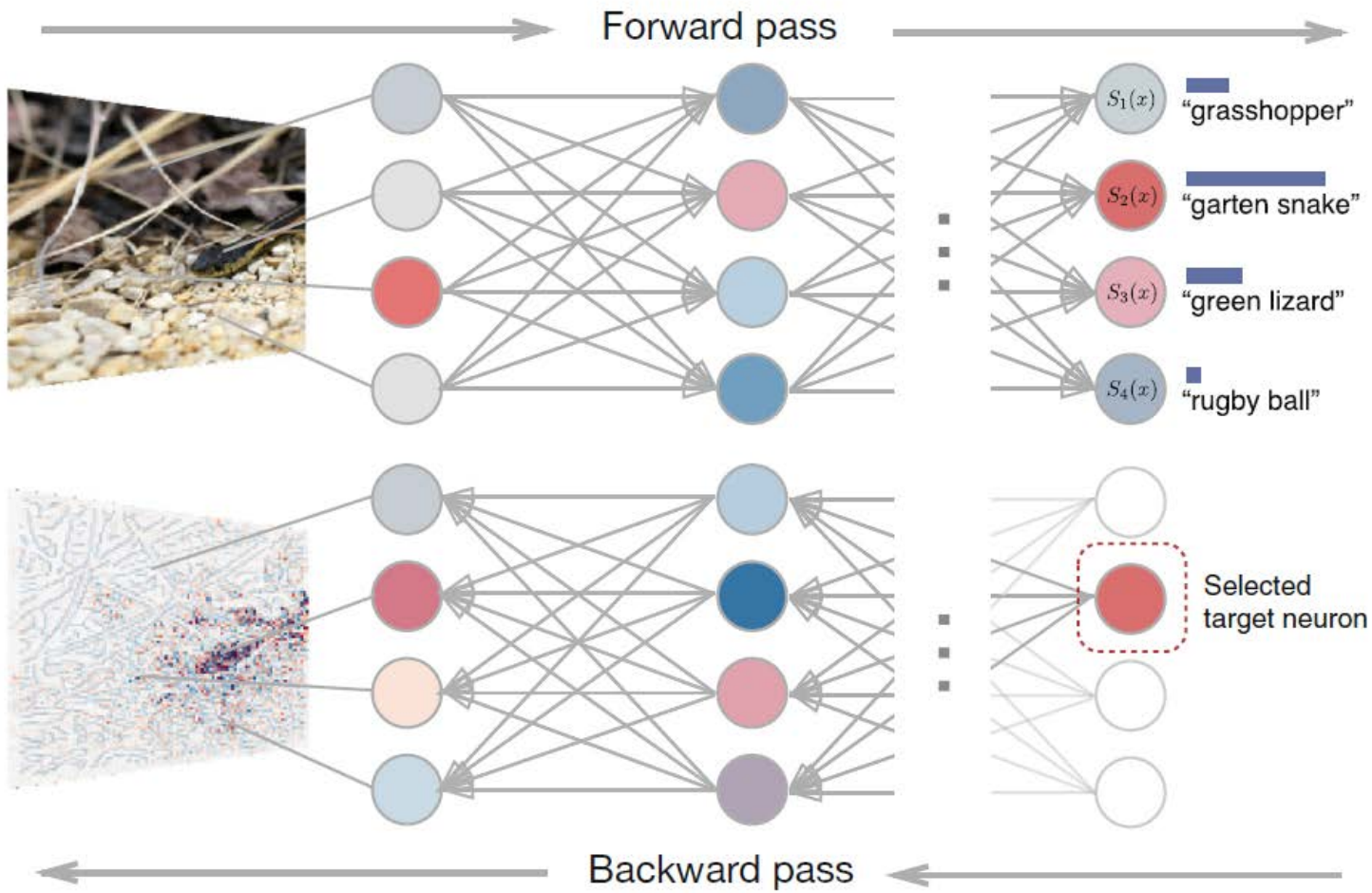
- Multiply the partial derivatives feature-wise by the input

$$R_i^c(x) = \frac{\partial S_c(x)}{\partial x_i} \cdot x_i$$

Or written as following
When f is prediction function

$$R_i = \left. \frac{\partial f}{\partial x_i} \right|_{\boldsymbol{x}} \cdot x_i$$

Attribution Methods for Non-Linear Models



[Image courtesy of Ancona Marco]

Goal of Input Attribution Methods

☐ Definition of ϵ-LRP

- $r_i^{(l)}$: relevance of unit I of layer l
- The relevance of the target neuron c is the activation of the neuron
- z_{ij}:the weighted activation of a neuron i onto neuron j

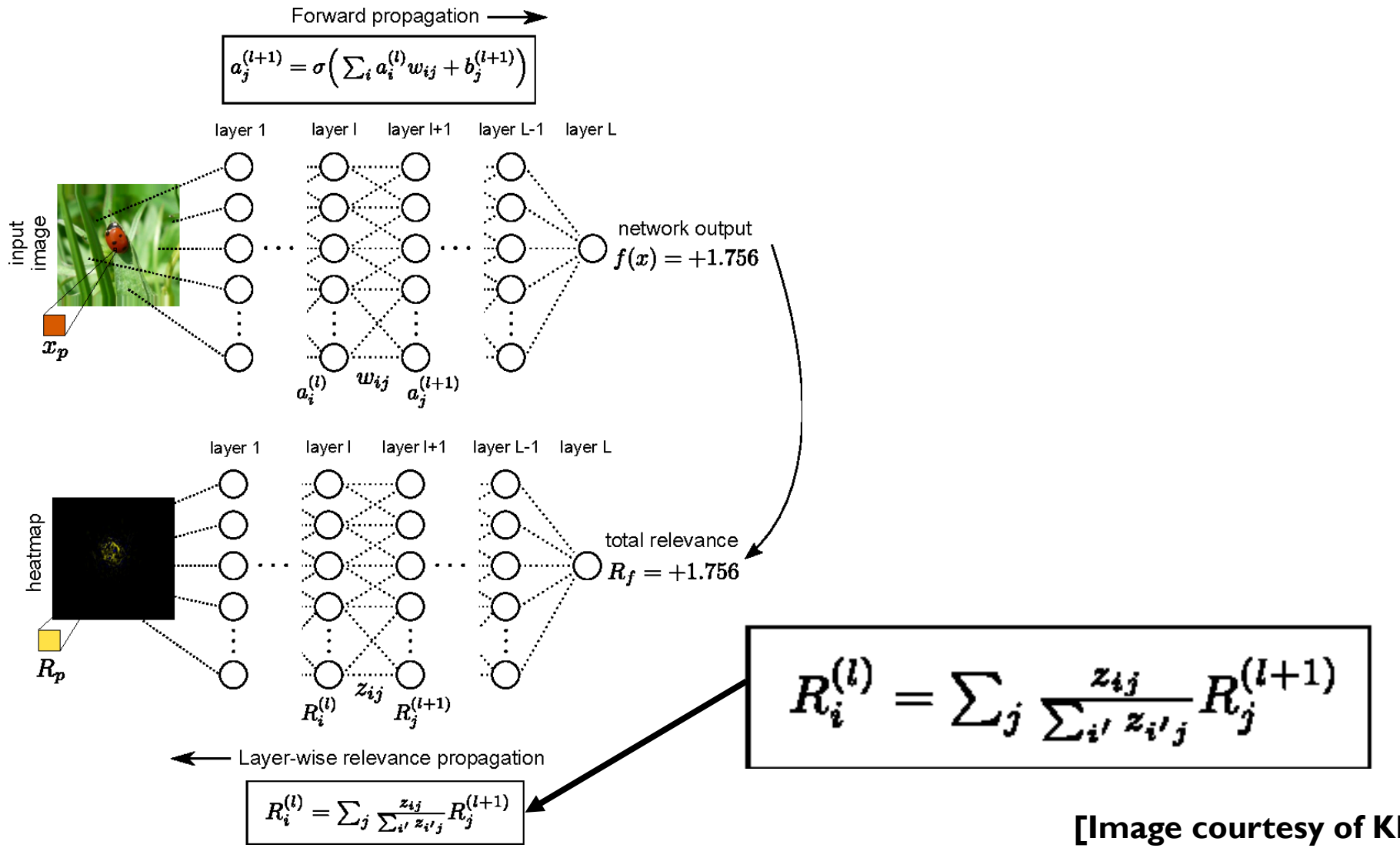
- b_j: the additive $z_{ij} = x_i^{(l)} w_{ij}^{(l,l+1)}$

$$r_i^{(L)} = \begin{cases} S_i(x) & \text{if unit } i \text{ is the target unit of interest} \\ 0 & \text{otherwise} \end{cases}$$

$$r_i^{(l)} = \sum_j \frac{z_{ij}}{\sum_{i'} z_{i'j} + b_j + \epsilon \cdot sign(\sum_{i'} z_{i'j} + b_j)} r_j^{(l+1)}$$

- In the input layer, the final attributions are $R_i^c(x) = r_i^{(1)}$

Definition: ϵ-Layer-wise Relevance Propagation (LRP)



[Image courtesy of Klaus Muller]

An Example of LRP

- The chain rule along a single path is the produce of the partial derivatives of all linear and nonlinear transformations along the path.
- For two units i and j in subsequent layers

$$\partial x_j / \partial x_i = w_{ji} \cdot f'(z_j)$$

- P_{ic}: a set of paths connect units i and c

$$\frac{\partial^g x_c}{\partial x_i} = \sum_{p \in P_{ic}} \left(\prod w_p \prod g(z)_p \right)$$

- When $g() = f'()$

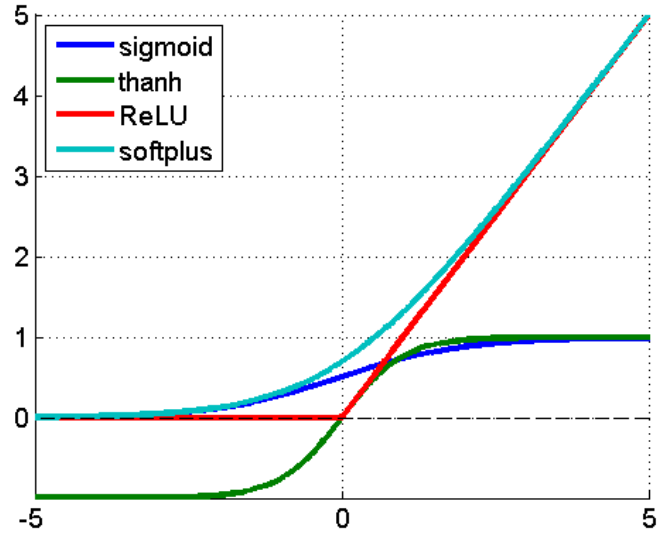
- This does work for fully-connected, convolutional, recurrent layers without multiplicative units, pooling operations

Some Notes on LRP

Proposition 1: $\epsilon - LRP$ is equivalent to the feature-wise product of the input and the modified partial derivative $\partial^g S_c(x)/\partial x_i$, with $g = g^{LRP} = f_i(z_i)/z_i$, i.e. the ratio between the output and the input at each nonlinearity.

- In ReLU or Tanh activations, $g^{LRP}(z)$ is the average gradient of the nonlinearity in [0, z].

$$g^{LRP}(z) = (f(z) - 0)/(z - 0)$$



Correctness of LRP on Computing Average Gradient

- Proof by induction. By definition, the ϵ-LRP relevance of the target neuron c on the top layer L to be equal to the output of the neuron, S_c:

$$r_c^{(L)} = S_c(x) = f\left(\sum_j w_{cj}^{(L,L-1)} x_j^{(L-1)} + b_c\right)$$

Correctness of LRP: Proof of proposition 1

- The relevance of the parent layer is:

$$
\begin{aligned}
r_j^{(L-1)} &= r_c^L \frac{w_{cj}^{(L,L-1)} x_j^{(L-1)}}{\sum_{j'} w_{cj'}^{(L,L-1)} x_{j'}^{(L-1)} + b_c} && \text{LRP propagation rule} \\
&= f\left(\sum_{j'} w_{cj'}^{(L,L-1)} x_{j'}^{(L-1)} + b_c\right) \frac{w_{cj}^{(L,L-1)} x_j^{(L-1)}}{\sum_{j'} w_{cj'}^{(L,L-1)} x_{j'}^{(L-1)} + b_c} && r_c^L \text{ is substituted} \\
&= g^{LRP}\left(\sum_{j'} w_{cj'}^{(L,L-1)} x_{j'}^{(L-1)} + b_c\right) w_{cj}^{(L,L-1)} x_j^{(L-1)} && r_c^L \text{ is substituted} \\
&= \frac{\partial g^{LRP} S_c(x)}{\partial x_j^{(L-1)}} x_j^{(L-1)} && \text{by definition of}
\end{aligned}
$$

$$
\frac{\partial^g x_c}{\partial x_i} = \sum_{p \in P_{ic}} \left(\prod w_p \prod g(z)_p\right)
$$

Correctness of LRP: Proof continued

- For the inductive step from the hypothesis that on a layer l the LRP explanation is:

$$r_i^{(l)} = \frac{\partial^{g^{LRP}} S_c(x)}{\partial x_i^{(l)}} x_i^{(l)}$$

$x_i^{(l)} = f\left(\sum_{j'} w_{ij'}^{(l,l-1)} x_{j'}^{(l-1)} + b_i\right)$

- Then for layer l-1 it holds:

$$r_j^{(l-1)} = \sum_i r_i^{(l)} \frac{w_{ij}^{(l,l-1)} x_j^{(l-1)}}{\sum_{j'} w_{ij'}^{(l,l-1)} x_{j'}^{(l-1)} + b_i}$$

LRP propagation rule

$$= \sum_i \frac{\partial^{g^{LRP}} S_c(x)}{\partial x_i^{(l)}} \underbrace{\frac{x_i^{(l)}}{\sum_{j'} w_{ij'}^{(l,l-1)} x_{j'}^{(l-1)} + b_i}}_{g^{LRP}(x_i^l)} w_{ij}^{(l,l-1)} x_j^{(l-1)}$$

$$= \frac{\partial^{g^{LRP}} S_c(x)}{\partial x_j^{(l-1)}} x_j^{(l-1)}$$

By definition of $\frac{\partial^g x_c}{\partial x_i} = \sum_{p \in P_{ic}} \left(\prod w_p \prod g(z)_p \right)$

■

Correctness of LRP: Proof continued

☐ DeepLIFT Rescale

- $\bar{x}$: baseline input

$$r_i^{(L)} = \begin{cases} S_i(x) - S_i(\bar{x}) & \text{if unit } i \text{ is the target unit of interest} \\ 0 & \text{otherwise} \end{cases}$$

$$r_i^{(l)} = \sum_j \frac{z_{ij} - \bar{z}_{ij}}{\sum_{i'} z_{i'j} - \sum_{i'} \bar{z}_{i'j}} r_j^{(l+1)}$$

Definition: DeepLIFT Rescale

☐ ϵ-LRP equivalence

Proposition: $\epsilon - LRP$ is equivalent to

*(i) Gradient * Input if only ReLUs are used as nonlinearities:*

(ii) DeepLIFT (computed with a zero baseline) if applied to a network with no additive biases and with nonlinearities f such that f(0)=0 (e.g., RELU or Tanh).

Equivalence of ϵ-LRP, Gradient * Input and DeepLIFT

Implementation Invariance Method

☐ Integrated Gradients

- LRP and DeepLIFT replace each instant gradient by an average gradient at each nonlinearity does not necessarily result in the average gradient of the function as a whole.
- Thus the attribution method fails to satisfy implementation invariance.
- It computes attributions multiplying the input variable element-wise with the average partial derivative as the input varies from a baseline $\bar{x}$ to its final value x.

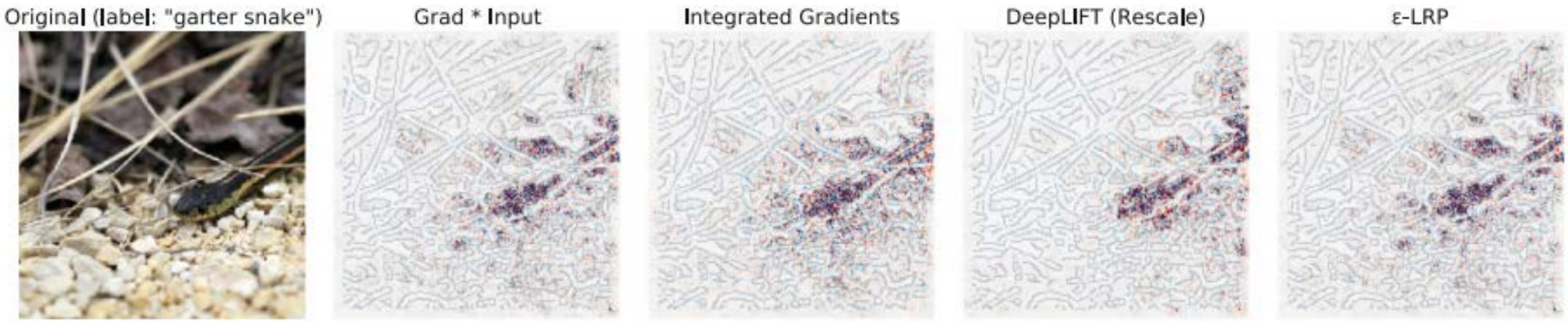
$$R_i^c(x) = x_i \cdot \int_{\alpha=0}^{1} \frac{\partial S_c(\tilde{x})}{\partial(\tilde{x}_i)}\Big|_{\tilde{x}=\bar{x}+\alpha(x-\bar{x})} d\alpha$$

- It satisfies sensitivity-N.

Method	Attribution $R_i^c(x)$
Sensitivity analysis	$\left\lvert \frac{\partial S_c(x)}{\partial x_i} \right\rvert$
Gradient * Input	$x_i \cdot \frac{\partial S_c(x)}{\partial x_i}$
ϵ-LRP	$x_i \cdot \frac{\partial^g S_c(x)}{\partial x_i}, \quad g = \frac{f(z)}{z}$
DeepLIFT (Rescale)	$(x_i - \bar{x_i}) \cdot \frac{\partial^g S_c(x)}{\partial x_i}, \quad g = \frac{f(z) - f(\bar{z})}{z - \bar{z}}$
Integrated Gradients	$(x_i - \bar{x_i}) \cdot \int_{\alpha=0}^{1} \frac{\partial S_c(\tilde{x})}{\partial(\tilde{x_i})} \Big\rvert_{\tilde{x}=\bar{x}+\alpha(x-\bar{x})} d\alpha$

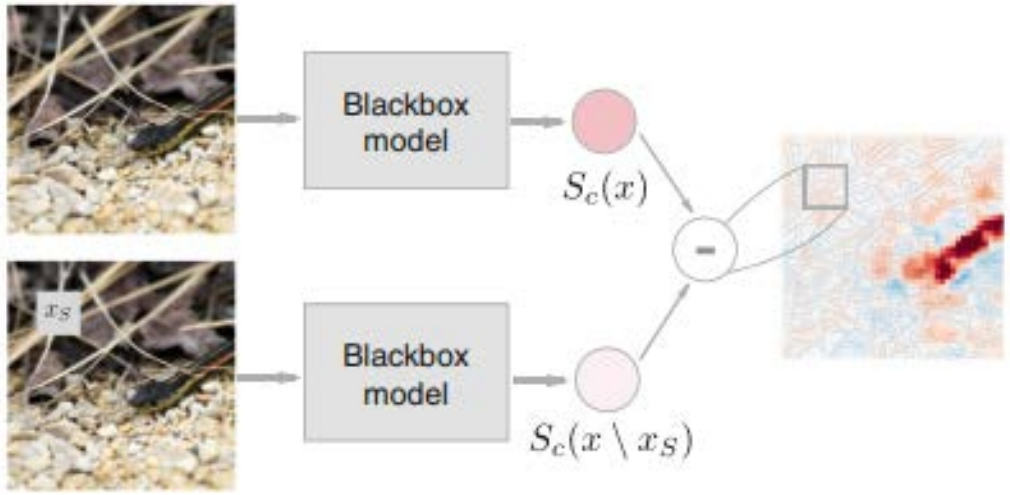
Comparisons of Attribution Methods

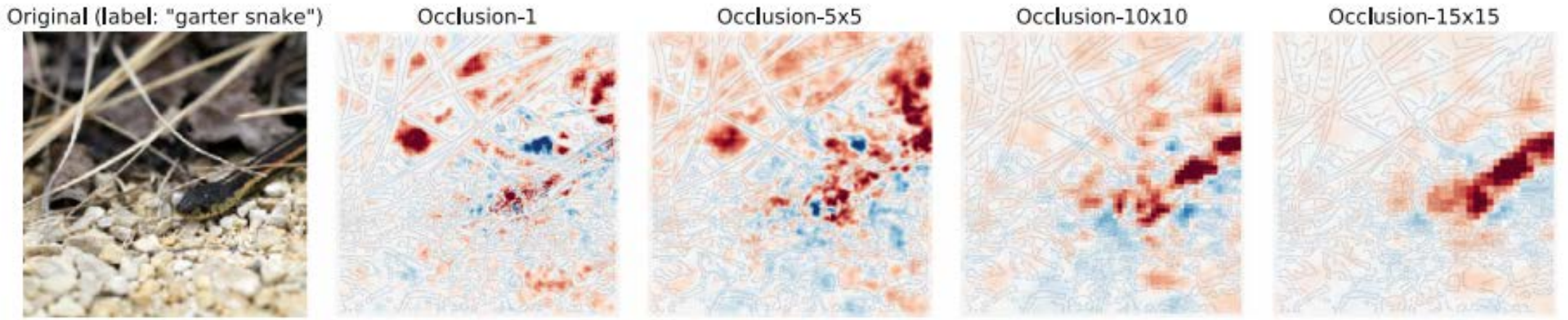
☐ Comparisons of Attribution Methods



[Image courtesy of Ancona Marco]

Comparisons of Attribution Methods





[Image courtesy of Ancona Marco]

Results with Perturbation Methods

Saliency Maps
Simonyan et al. 2015

Integrated Gradients
Sundararajan et al. 2017

DeepLIFT
Shrikumar et al. 2017

LIME
Ribeiro et al. 2016

Gradient * Input
Shrikumar et al. 2016

Layer-wise Relevance Propagation (LRP)
Bach et al. 2015

Guided Backpropagation
Springenberg et al. 2014

Grad-CAM
Selvaraju et al. 2016

Simple occlusion
Zeiler et al. 2014

Meaningful Perturbation
Fong et al. 2017

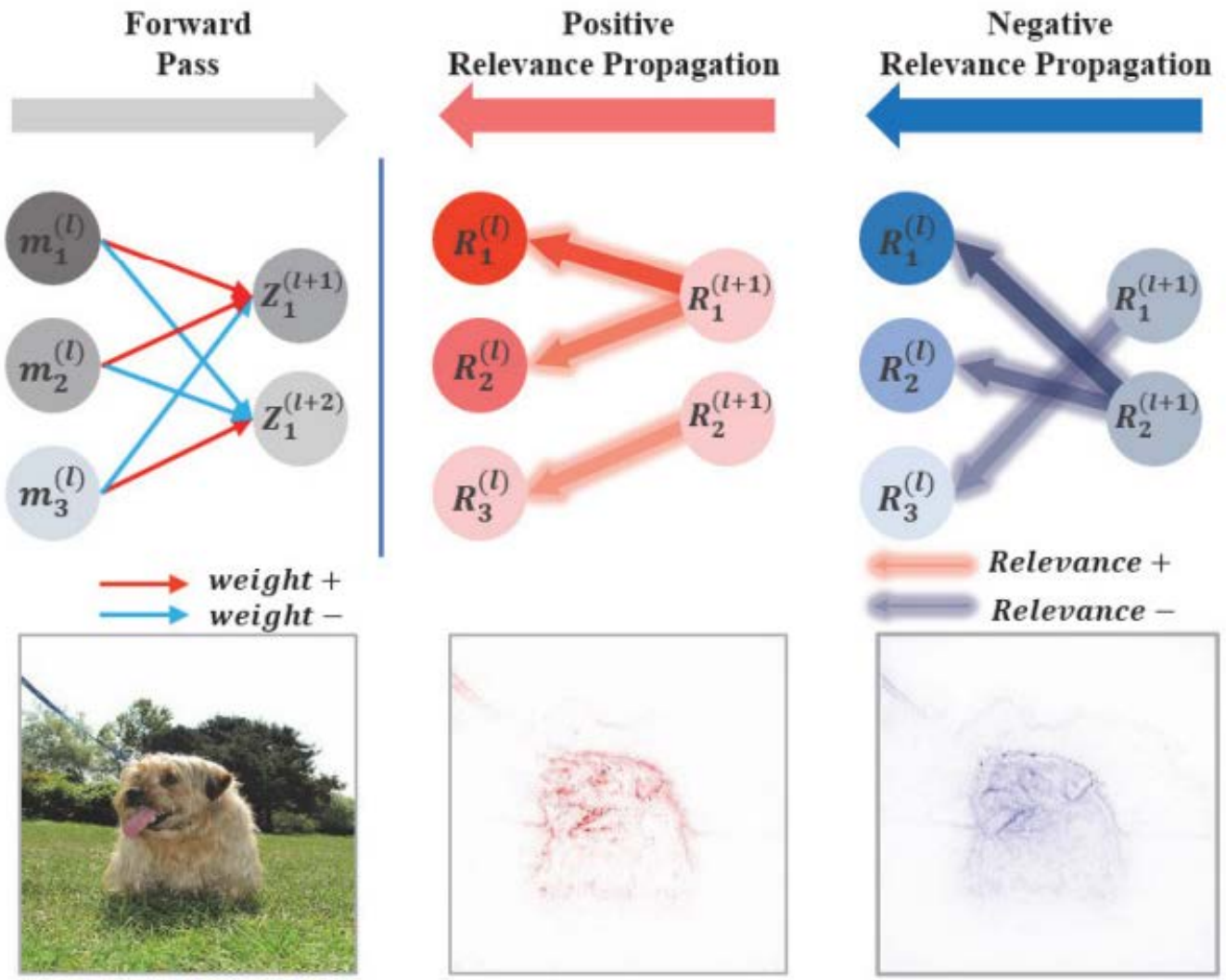
Prediction Difference Analysis
Zintgraf et al. 2017

KernelSHAP/DeepSHAP
Lundberg et al., 2017

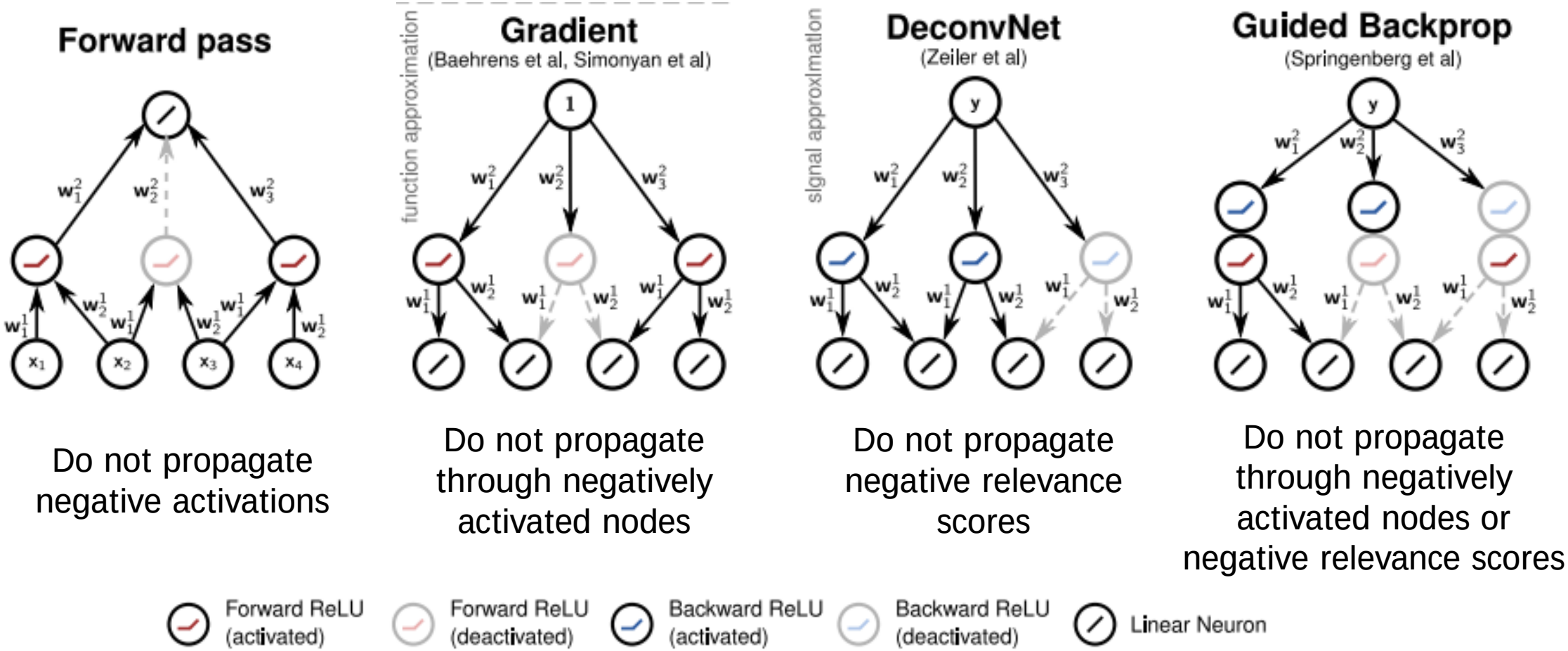
Slides courtesy of [Marco Ancona, et. al., Explaining Deep Neural Networks with a Polynomial Time Algorithm for Shapley Values Approximation, ICML 2019]

[Image courtesy of Ancona Marco]

Some References



Issues with Positive/Negative Relevance Propagation



Do not propagate negative activations

Do not propagate through negatively activated nodes

Do not propagate negative relevance scores

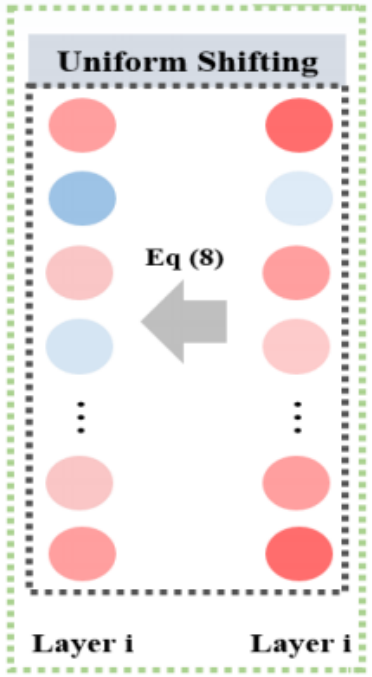
Do not propagate through negatively activated nodes or negative relevance scores

[Image courtesy of Klaus Muller]

Handling Negative Relevance Scores During the Propagation

III) Uniform Shifting

Switch low priority neuron to negative

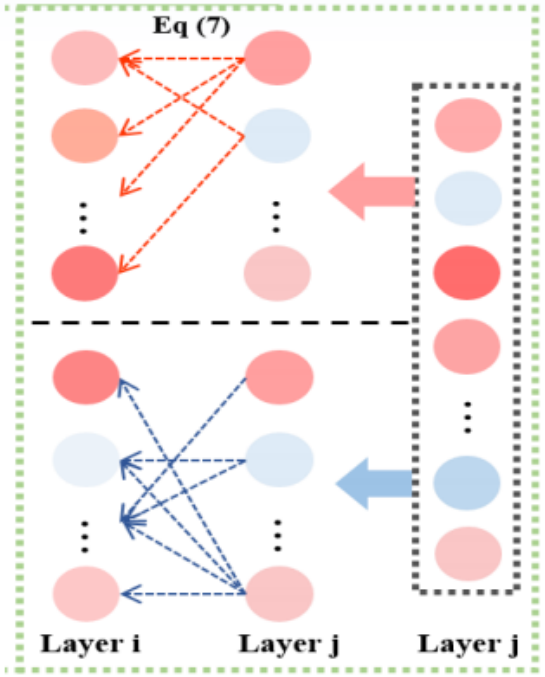


$$\Psi_i^l = \begin{cases} \sum_i \left(\bar{R}_{i\epsilon N}^{(l)}\right) * \frac{1}{\Gamma}, & m_i^l \text{ is activated} \\ 0, & \text{otherwise} \end{cases}$$

$$R_j^{(l)} = \bar{R}_{i\epsilon P \cup N}^{(l)} - \Psi_i^l$$

II) Criterion of Relevant Neuron

Deciding more relevant neuron to baseline

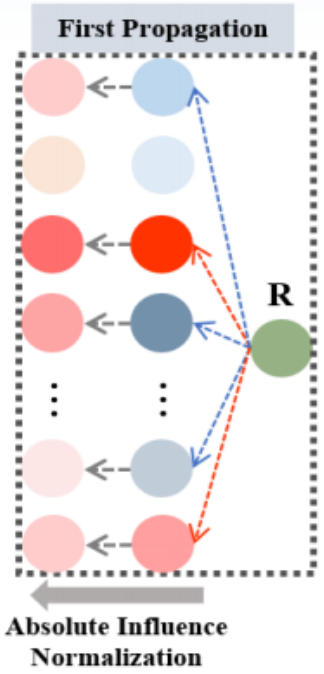


$$v_j^{(l+1)} = R_j^{(l+1)} * \frac{\sum_i |z_{ij}^-|}{\sum_i (|z_{ij}^+| + |z_{ij}^-|)}$$

$$\bar{R}_{i\epsilon P,N}^{\prime(l)} = \sum_j \left(\frac{z_{ij}^+}{\sum_i (z_{ij}^+)} R_j^{(l+1)} + \frac{z_{ij}^-}{\sum_i (z_{ij}^-)} v_j^{(l+1)} \right)$$

I) Absolute Influence Normalization

Changing perspective: From value to influence

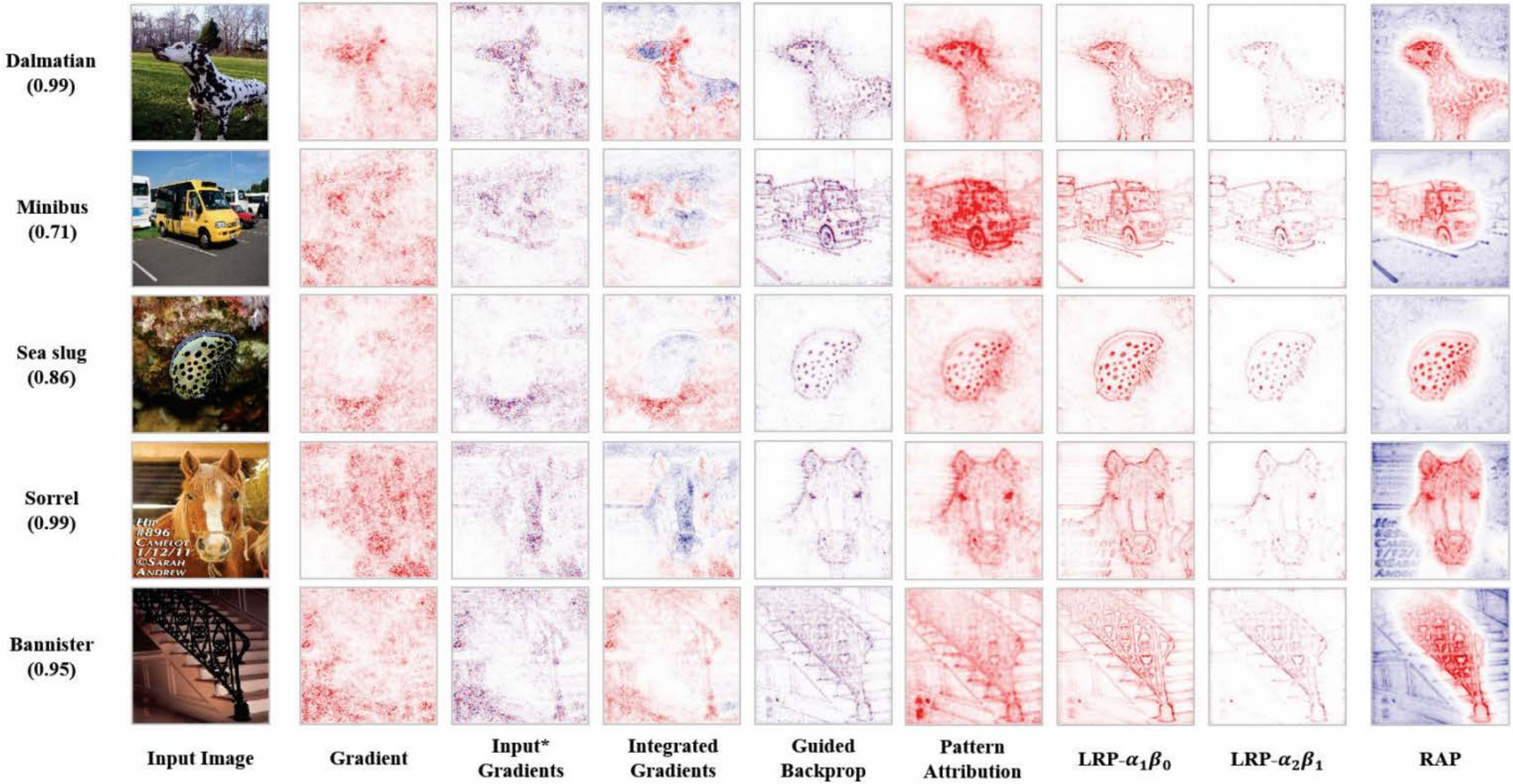


$$R_i^{(p)} = \left(\sum_i z_{ij}^+ + \sum_i z_{ij}^- \right) * R_j^{(q)}$$

$$R_i^{\prime(p)} = |R_i^{(p)}| * \frac{\sum_i R_i^{(p)}}{\sum_i |R_i^{(p)}|}$$

Relative Attributing Propagation

Woo-Jeong Nam, et. al., "Relative Attributing Propagation: **Interpreting the Comparative Contributions of Individual Units in Deep Neural Networks"**, *AAAI, 2020*

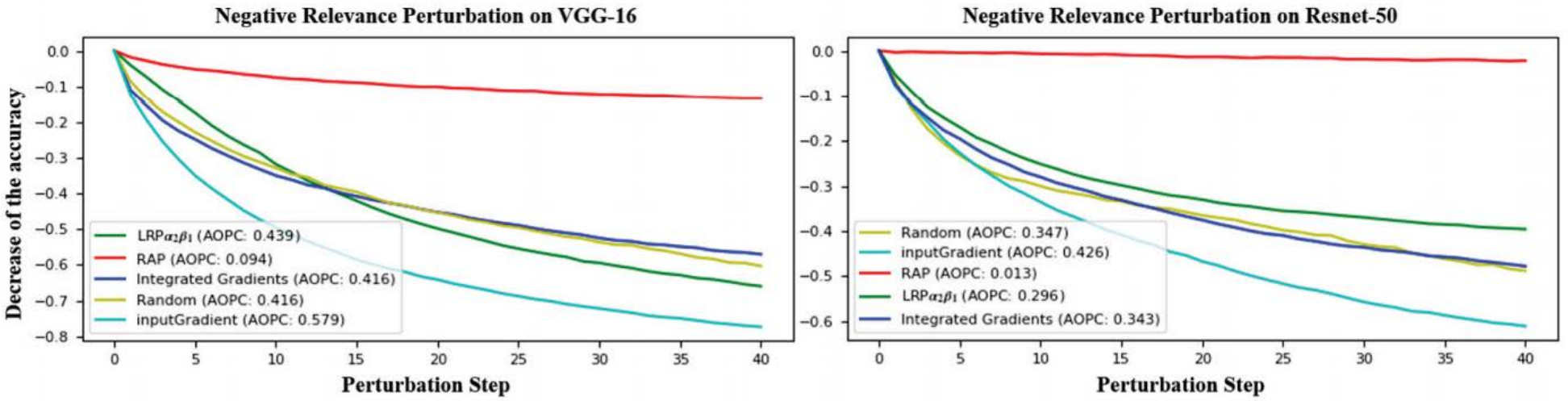


Relative Attributing Propagation: Quantitative Evaluations

Quantitative Performance

Outside-Inside Ratio		RAP	$LRP_{\alpha_1\beta_0}$	$LRP_{\alpha_2\beta_1}$	Gradient	Input* Gradient	Integrated Gradients	Pattern Attribution	Guided Backprop
VGG-16	ALL	**0.252**	-	0.616	-	0.989	1.230	-	1.069
	POS	**0.341**	0.474	0.524	0.619	0.691	0.827	0.415	0.427
Res-50	ALL	**0.164**	-	0.302	-	0.996	1.195	-	1.035
	POS	**0.166**	0.429	0.299	0.597	0.689	0.698	-	0.296
Segmentation Mask		**RAP**	$LRP_{\alpha_1\beta_0}$	$LRP_{\alpha_2\beta_1}$	Gradient	Input* Gradient	Integrated Gradients	Pattern Attribution	Guided Backprop
Imagenet	PIX ACC	**79.23**	75.40	72.95	70.01	66.38	66,52	76.84	71.98
	mIOU	**62.23**	55.78	50.86	49.30	44.01	45.90	58.05	49.87
Pascal VOC	PIX ACC	**73.91**	70.86	69.43	68.14	50.01	52.38	-	66.92
	mIOU	**55.60**	49.82	46.85	46.07	31.69	34.39	-	43.63

When Perturbating pixels with negative attributions



Relative Attributing Propagation: Quantitative Evaluations

- Input attribution methods can compute the contributions of individual inputs.

- Under some assumptions, results of different input attribution methods are equivalent.

- Handling negative attributions are also important.

Conclusions of Part I

References

1. [XAI] Gunning, D. (2017). Explainable artificial intelligence. Defense Advanced Research Projects Agency (DARPA).

2. [XAI-Perspective] Gunning, D., Stefik, D., Choi, J., Miller, T., Stumpf, S. and Yang G.-Z.(2019), XAI—Explainable artificial intelligence, Science Robotics, 4(37).

3. [LRP] Bach, S., Binder, A., Montavon, G., Klauschen, F., Müller, K. R., & Samek, W. (2015). On pixel-wise explanationsfor non-linear classifier decisions by layer-wise relevance propagation. PloS one, 10(7), e0130140.

4. [DTD] Montavon, G., Lapuschkin, S., Binder, A., Samek, W., & Müller, K. R. (2017). Explaining nonlinear classification decisions with deep taylor decomposition. Pattern Recognition, 65, 211-222.

5. [PatternNet] Li, H., Ellis, J. G., Zhang, L., & Chang, S. F. (2018). Patternnet: Visual pattern mining with deep neural network. In Proceedings of the ACM on International Conference on Multimedia Retrieval (pp. 291-299).

6. [Clever Hans] Lapuschkin, S., Wäldchen, S., Binder, A., Montavon, G., Samek, W., Muller, K.-R. (2019). Unmasking Clever Hans predictors and assessing what machines really learn. Nature Communication 10, 1096.

7. [RAP] Nam, W. J., Choi, J., & Lee, S. W. (2020). Relative Attributing Propagation: Interpreting the Comparative Contributions of Individual Units in Deep Neural Networks. AAAI Conference on Artificial Intelligence. Gradient Based

8. [DeConvNet] Zeiler, Matthew D., and Rob Fergus. (2014). Visualizing and understanding convolutional networks. European conference on computer vision. Springer.

9. [DeepLIFT] Shrikumar, A., Greenside, P., & Kundaje, A. (2017). Learning important features through propagating activation differences. In Proceedings of the 34th International Conference on Machine Learning-Volume 70 (pp. 3145-3153). JMLR. org.

10. [Guided Backprop] Springenberg, J. T., Dosovitskiy, A., Brox, T., & Riedmiller, M. (2014). Striving for simplicity: The all convolutional net. arXiv preprint arXiv:1412.6806.

11. [GradCAM] Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., & Batra, D. (2017). Grad-cam: Visual explanations from deep networks via gradient-based localization. In Proceedings of the IEEE International Conference on Computer Vision(pp. 618-626). Explaining Internal Nodes

12. Ancona, M., Ceolini, E., Öztireli, C., Gross, M. (2018) Towards better understanding of gradient-based attribution methods for Deep Neural Networks, International Conference on Learning Representations.

13. Ancona, M., Öztireli, C., Gross, M. (2019) Explaining Deep Neural Networks with a Polynomial Time Algorithm for Shapley Values Approximation, International Conference on Machine Learning.