Advanced Deep Learning: Explainability

Anders Søgaard

coastal

Course outline

Goal 1: Quick tour of recent developments in deep learning

Goal 2: Inspiration for thesis/research projects

Week	Lecturer	Subject	Literature	Assignment
1	Stefan	Introduction to Neural Networks.	d2l 2.1-2.5, 2.7, 11.5.1, **slides**	
2	Stefan	CNNs; FCNs; U-Nets. Data augmentation; invariance; regularization e.g. dropout	d2l 6, 7, 13.9-13.11, **slides**	Assignment 1 (May 10)
3	Anders/Phillip	May 9 (A): RNNs May 11 (P): Transformers	d2l 8 Transformers: d2l 10.5-10.7 + Vaswani et al. (2017)	Assignment 2 (May 20)
4	Phillip/Anders	May 16 (P): Representation and Adversarial Learning May 18 (A): A Learning Framework + Self-supervised Learning + Contrastive Learning	Autoencoders: blog post GANs: Goodfellow (2016) Self-supervised learning: blog post Contrastive learning: Dor et al. (2018) Adversarial examples: Goodfellow et al. (2015)	
5	Anders	May 23: General Properties, e.g., Scaling Laws, Lottery Tickets, Bottleneck Phenomena May 25: Applications of Representation, Adversarial and Contrastive Learning	GANs: Lample et al. (2018) Autoencoders: Chandar et al. (2011) Contrastive learning: Yu et al. (2018) DynaBench: Talk by Douwe Kiela (Facebook, now HuggingFace) Scaling laws: Kaplan et al. (2020)	Assignment 3 *[MC on Representation Learning/1p Report on Lottery Ticket extraction]* (June 3)
6	Anders	May 30: Interpretability, Transparency, and Trustworthiness & Deep Learning for Scientific Discovery June 1: Interpretability (Feature Attribution), including Guest Lecture by Stephanie Brandl	DL for Scientific Discovery: Sullivan (2022) Interpretability/Background: Søgaard (2022)	
7	Anders	June 6: *Off (no teaching)* June 8: Interpretability (Training Data Influence)	Literature: Feng and Boyd-Graber (2018) ; Jiang and Senge (2021)	
8	Anders	June 13-15: Best Practices	Literature: Dodge et al. (2019) and Raji et al. (2021)	Assignment 4 *[MC on Interpretability; 1p Report on Best Practices]* (June 21)

Course outline

Goal 1: Quick tour of recent developments in deep learning

Goal 2: Inspiration for thesis/research projects

Architectures

Framework

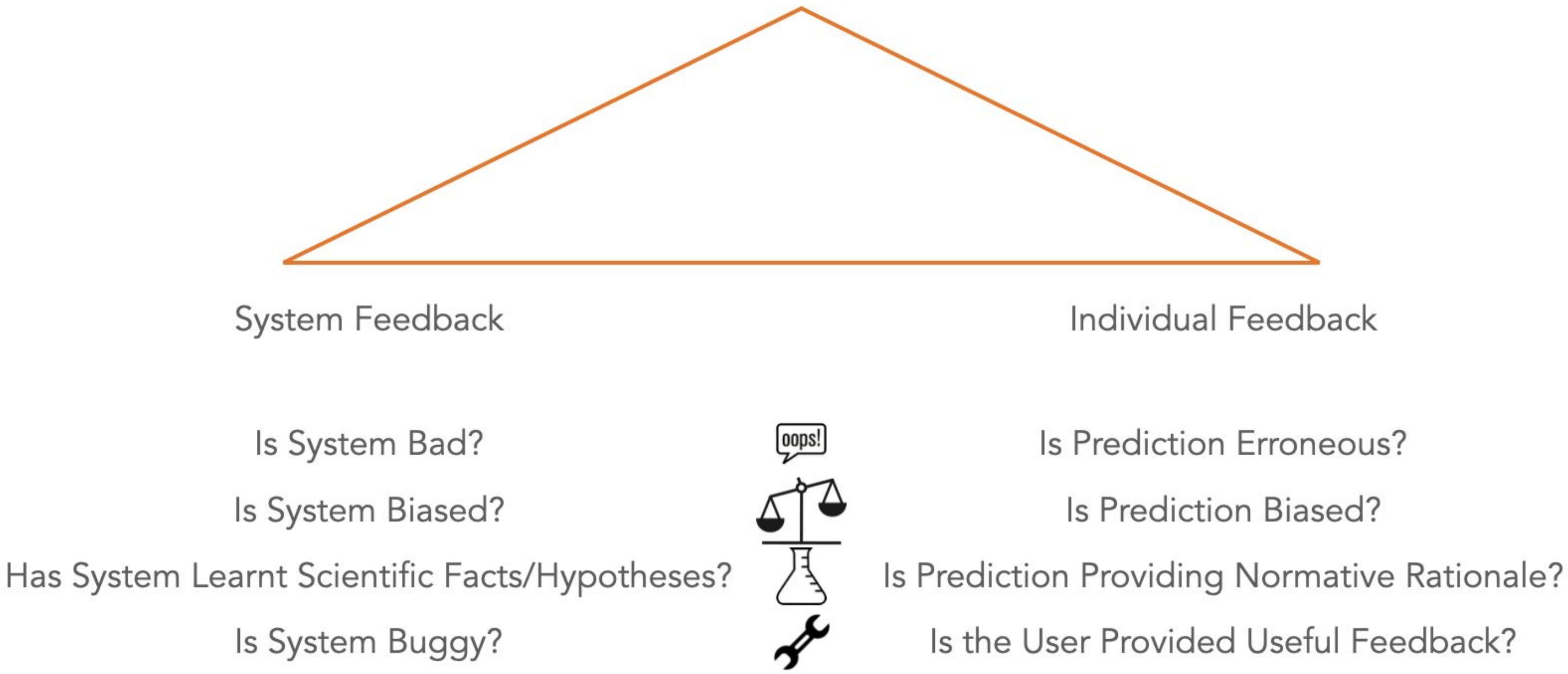
Fairness / Explainable AI

Methodology

Week	Lecturer	Subject	Literature	Assignment
1	Stefan	Introduction to Neural Networks.	d2l 2.1-2.5, 2.7, 11.5.1, **slides**	
2	Stefan	CNNs; FCNs; U-Nets. Data augmentation; invariance; regularization e.g. dropout	d2l 6, 7, 13.9-13.11, **slides**	Assignment 1 (May 10)
3	Anders/Phillip	May 9 (A): RNNs May 11 (P): Transformers	d2l 8 Transformers: d2l 10.5-10.7 + Vaswani et al. (2017)	Assignment 2 (May 20)
		May 16 (P): Representation and Adversarial Learning	Autoencoders: blog post GANs: Goodfellow (2016) Self-supervised learning: blog post Contrastive learning: Dor et al. (2018) Adversarial examples: Goodfellow et al. (2015)	
4	Phillip/Anders	May 18 (A): A Learning Framework + Self-supervised Learning + Contrastive Learning		
5	Anders	May 23: General Properties, e.g., Scaling Laws, Lottery Tickets, Bottleneck Phenomena May 25: Applications of Representation, Adversarial and Contrastive Learning	GANs: Lample et al. (2018) Autoencoders: Chandar et al. (2011) Contrastive learning: Yu et al. (2018) DynaBench: Talk by Douwe Kiela (Facebook, now HuggingFace) Scaling laws: Kaplan et al. (2020)	Assignment 3 *[MC on Representation Learning/1p Report on Lottery Ticket extraction]* (June 3)
6	Anders	May 30: Interpretability, Transparency, and Trustworthiness & Deep Learning for Scientific Discovery June 1: Interpretability (Feature Attribution), including Guest Lecture by Stephanie Brandl	DL for Scientific Discovery: Sullivan (2022) Interpretability/Background: Søgaard (2022)	
7	Anders	June 6: *Off (no teaching)* June 8: Interpretability (Training Data Influence)	Literature: Feng and Boyd-Graber (2018); Jiang and Senge (2021)	
8	Anders	June 13-15: Best Practices	Literature: Dodge et al. (2019) and Raji et al. (2021)	Assignment 4 *[MC on Interpretability; 1p Report on Best Practices]* (June 21)

Today

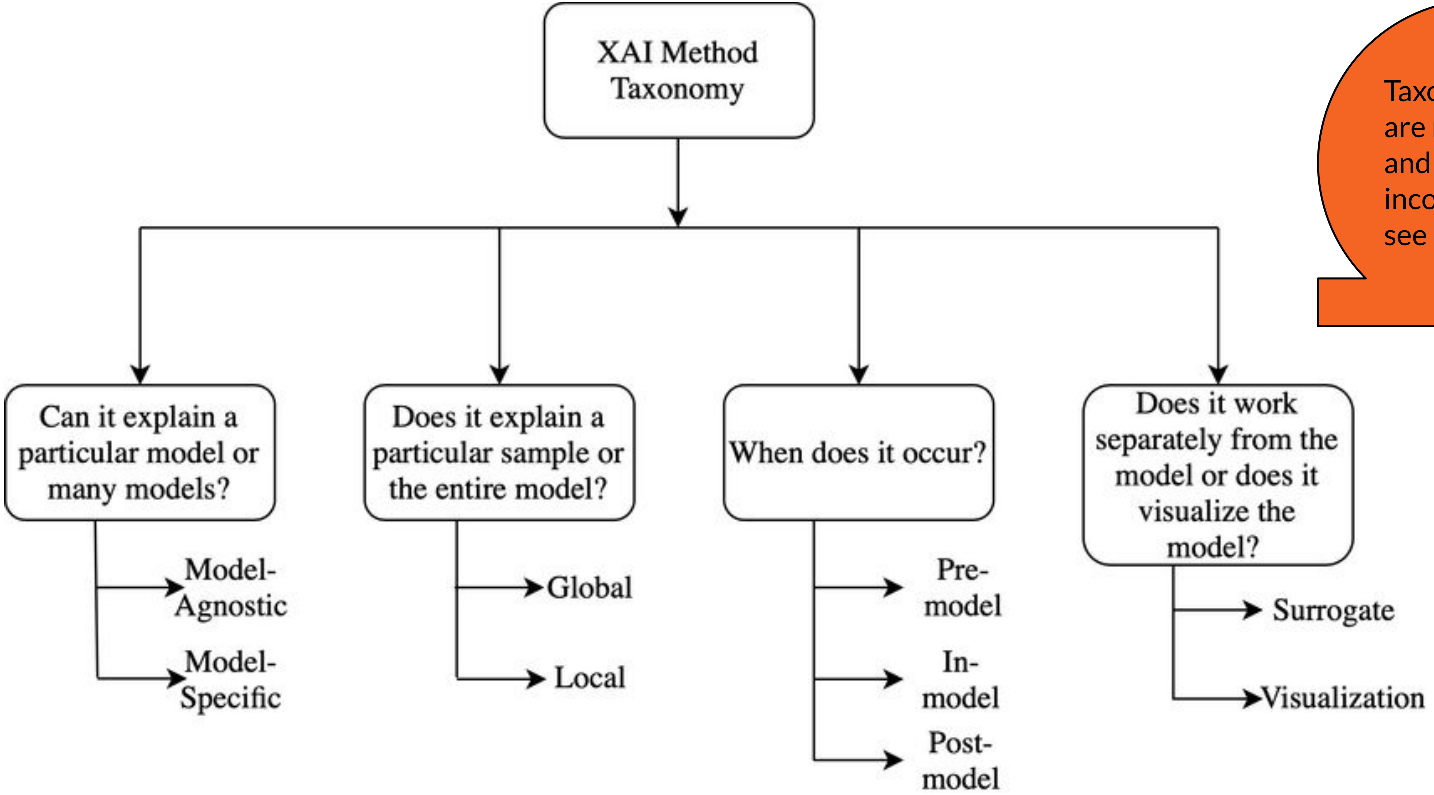
a) Taxonomies of XAI methods
b) Interpretable approximations
c) Feature attribution methods
d) Training data influence
e) Probing and error analyses


System Feedback
Individual Feedback
Is System Bad?
oops!
Is Prediction Erroneous?
Is System Biased?
Is Prediction Biased?
Has System Learnt Scientific Facts/Hypotheses?
Is Prediction Providing Normative Rationale?
Is System Buggy?
Is the User Provided Useful Feedback?

Motivation

Taxonomies

Standard taxonomy: loc-glob + intr-posthoc



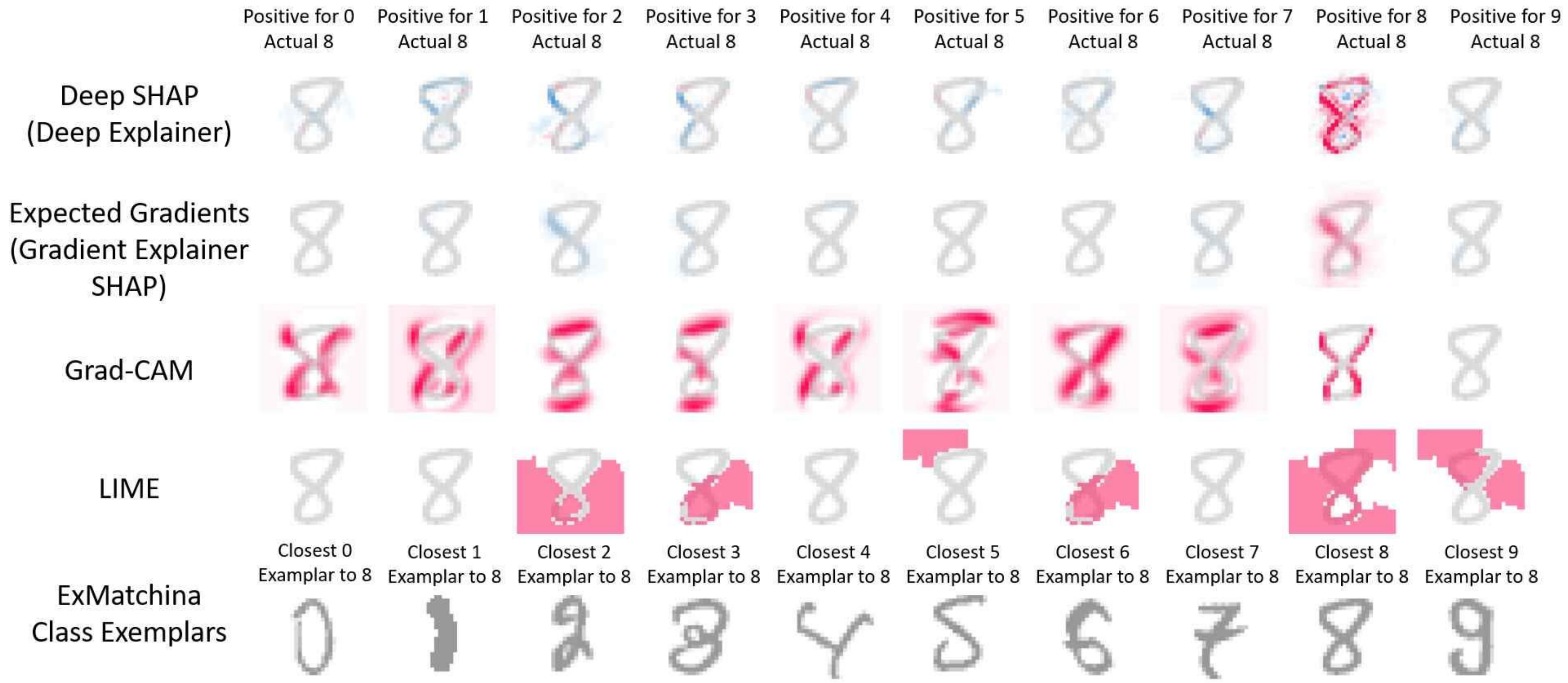
Taxonomies are redundant and inconsistent; see book.

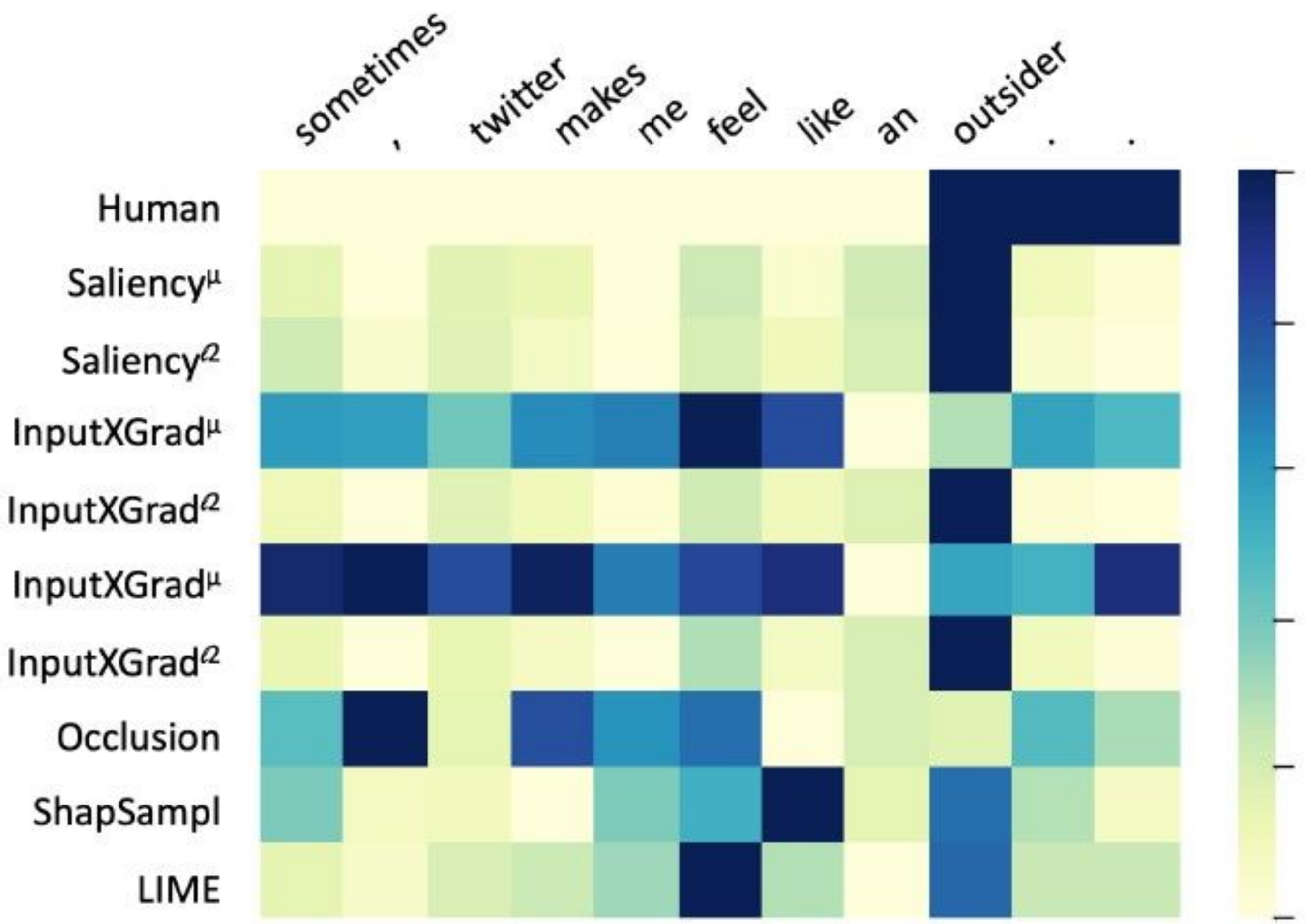
Feature attribution

Feature attribution

Using feature attribution methods to 'explain' deep neural networks took off in 2014-15.

Method	Year	Reference
Vanilla gradients	2014	Denil et al. (2014)
Guided back-propagation	2015	Springenberg et al. (2015)
Layer-wise relevance propagation	2015	Bach et al. (2015)
Deep Taylor decomposition	2017	Montavon et al. (2017)
Integrated gradients	2017	Sundararajan et al. (2017)
DeepLift	2017	Shrikumar et al. (2017)


Positive for 0 Actual 8
Positive for 1 Actual 8
Positive for 2 Actual 8
Positive for 3 Actual 8
Positive for 4 Actual 8
Positive for 5 Actual 8
Positive for 6 Actual 8
Positive for 7 Actual 8
Positive for 8 Actual 8
Positive for 9 Actual 8
Deep SHAP (Deep Explainer)
Expected Gradients (Gradient Explainer SHAP)
Grad-CAM
LIME
ExMatchina Class Exemplars
Closest 0 Examplar to 8
Closest 1 Examplar to 8
Closest 2 Examplar to 8
Closest 3 Examplar to 8
Closest 4 Examplar to 8
Closest 5 Examplar to 8
Closest 6 Examplar to 8
Closest 7 Examplar to 8
Closest 8 Examplar to 8
Closest 9 Examplar to 8



sometimes
,
twitter
makes
me
feel
like
an
outsider
.
.
Human
Saliency^μ
Saliency^ℓ2
InputXGrad^μ
InputXGrad^ℓ2
InputXGrad^μ
InputXGrad^ℓ2
Occlusion
ShapSampl
LIME

—

Vanilla gradients

- Compute the gradient of the loss function or the logit of the predicted class with respect to the input embeddings given model parameters.
- We cannot compute the gradient with respect to tokens, only with respect to their embeddings. We therefore must reduce the *d*-dimensional gradients to a scalar value, e.g., by computing their L_n norm.

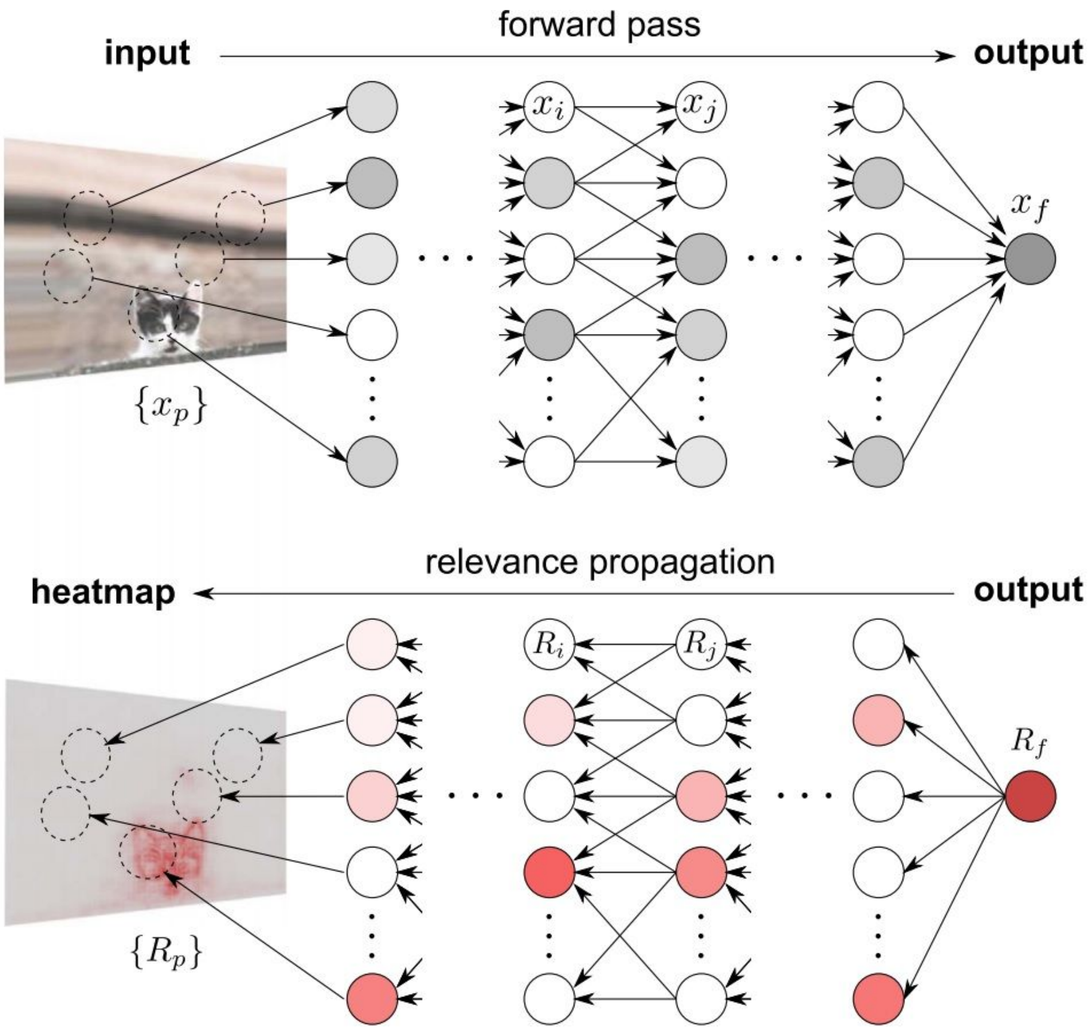
Vanilla gradients + Guided BP

- Compute the gradient of the loss function or the logit of the predicted class with respect to the input embeddings given model parameters.
- We cannot compute the gradient with respect to tokens, only with respect to their embeddings. We therefore must reduce the *d*-dimensional gradients to a scalar value, e.g., by computing their L_n norm.
- While using vanilla gradients relies on actual gradients, guided back-propagation only back-propagates positive error signals, setting negative gradients to zero, reflecting the intuition that positive gradients provide more direct explanations of model decisions.

Layerwise Relevance Propagation

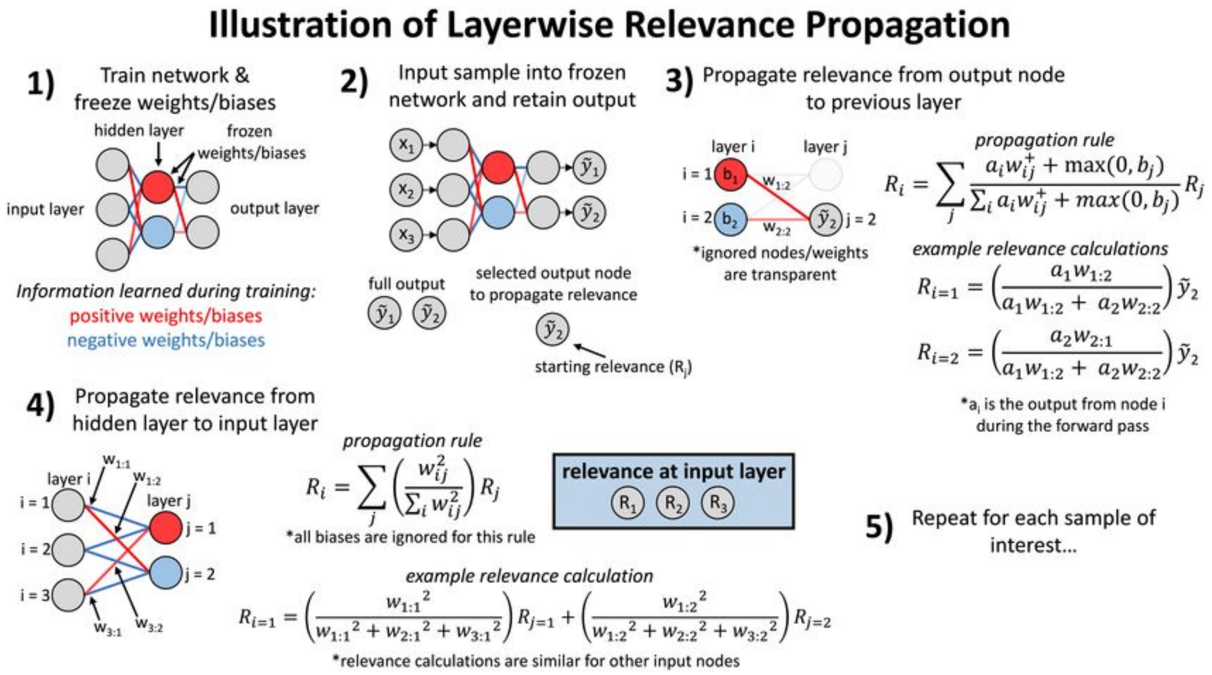
LRP can be seen as an instance of Deep Taylor Expansion, and equivalent to a restricted version of DeepLift.

Works as follows: Associate the actual prediction with a relevance score R_f. Going backwards you compute subsequent relevance scores as the sum of the incoming relevance scores multiplied by (normalized) activation times weight.



—

- LRP back-propagates relevance recursively from the output layer to the input layer.
- Deep Taylor Decomposition (see video on Absalon) is its theoretical motivation.



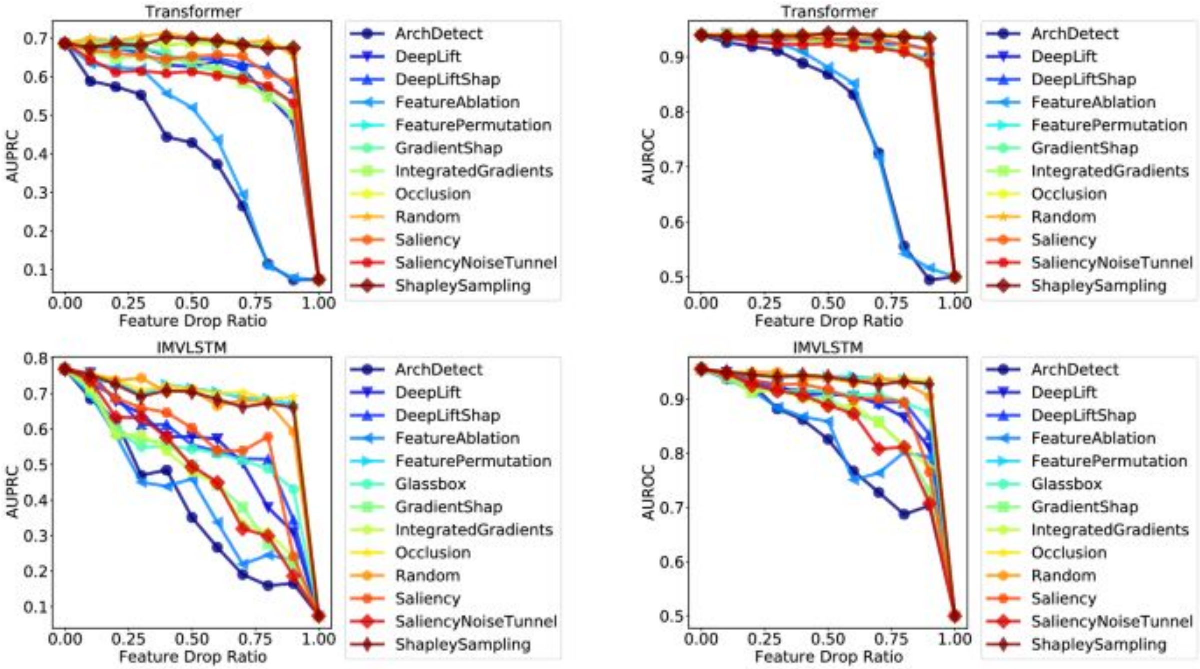
Other methods

- Input x Gradient (like Vanilla Gradients, but multiplied by input; near-equivalent to simple LRP)
- DeepLIFT (normalization by reference point)
- Integrated Gradients (integrating gradients from reference point to data point)

Evaluating Feature attribution

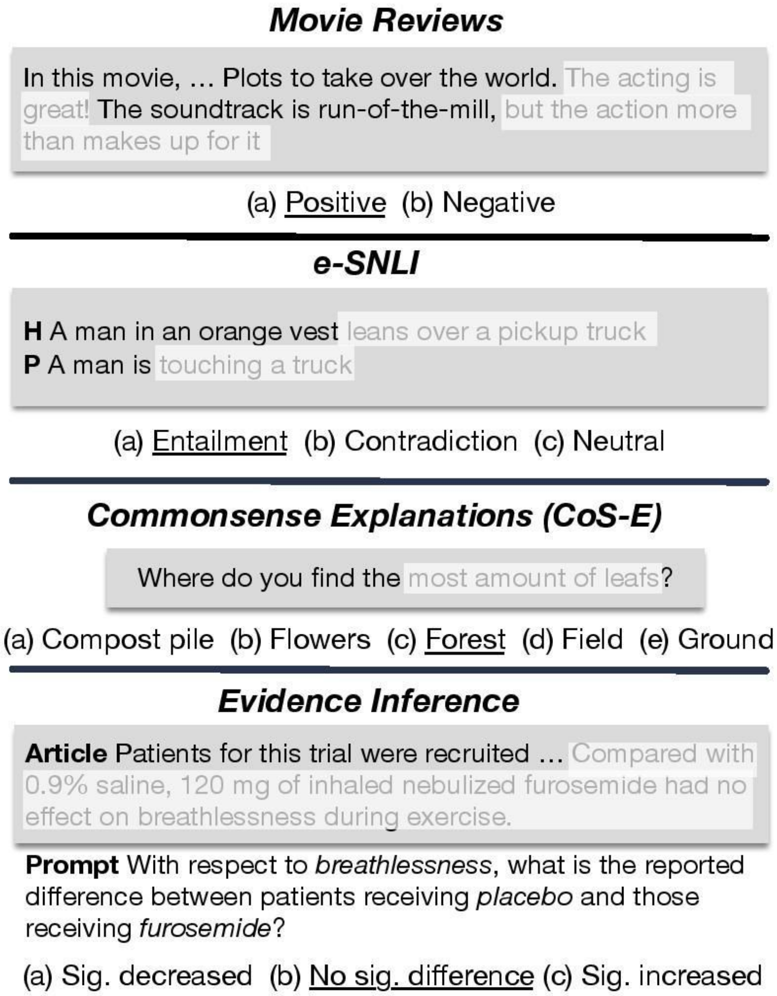
Input reduction

- Dropping inputs in order of importance.
- The faster the curve drops, the better.



Human rationales

- F1 agreement with human rationales.
- The higher the overlap the better.



Exercise

What’s the problem with input reduction and overlap with human rationales?

Hints:

1) Think about i.i.d.
2) Think about what we’re trying to explain.

Training data influence

Influence functions

An old technique for quantifying how the model parameters change as we upweight a training point by an infinitesimal amount.

Problems: Expensive and only works for convex models.

Solution: Approximations.

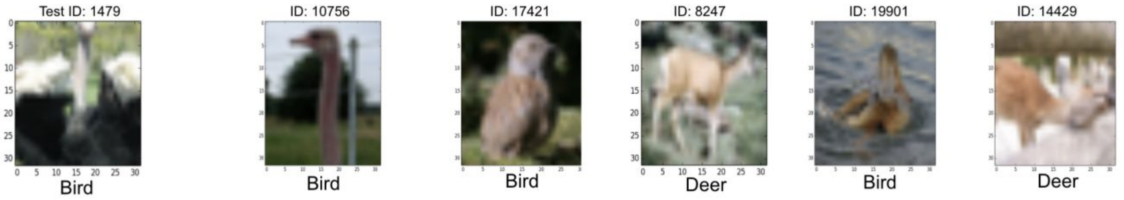


Figure 6: Top 5 influential points for the test point: 1479 (CIFAR-10). The model is a ResNet-18 trained with a weight-decay regularization; Only 3 out of the 5 points are semantically similar to the test-point with class "Bird".

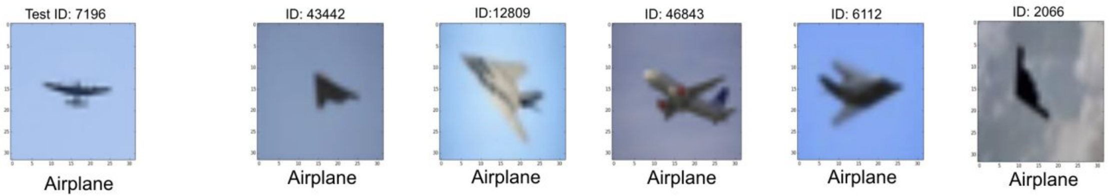
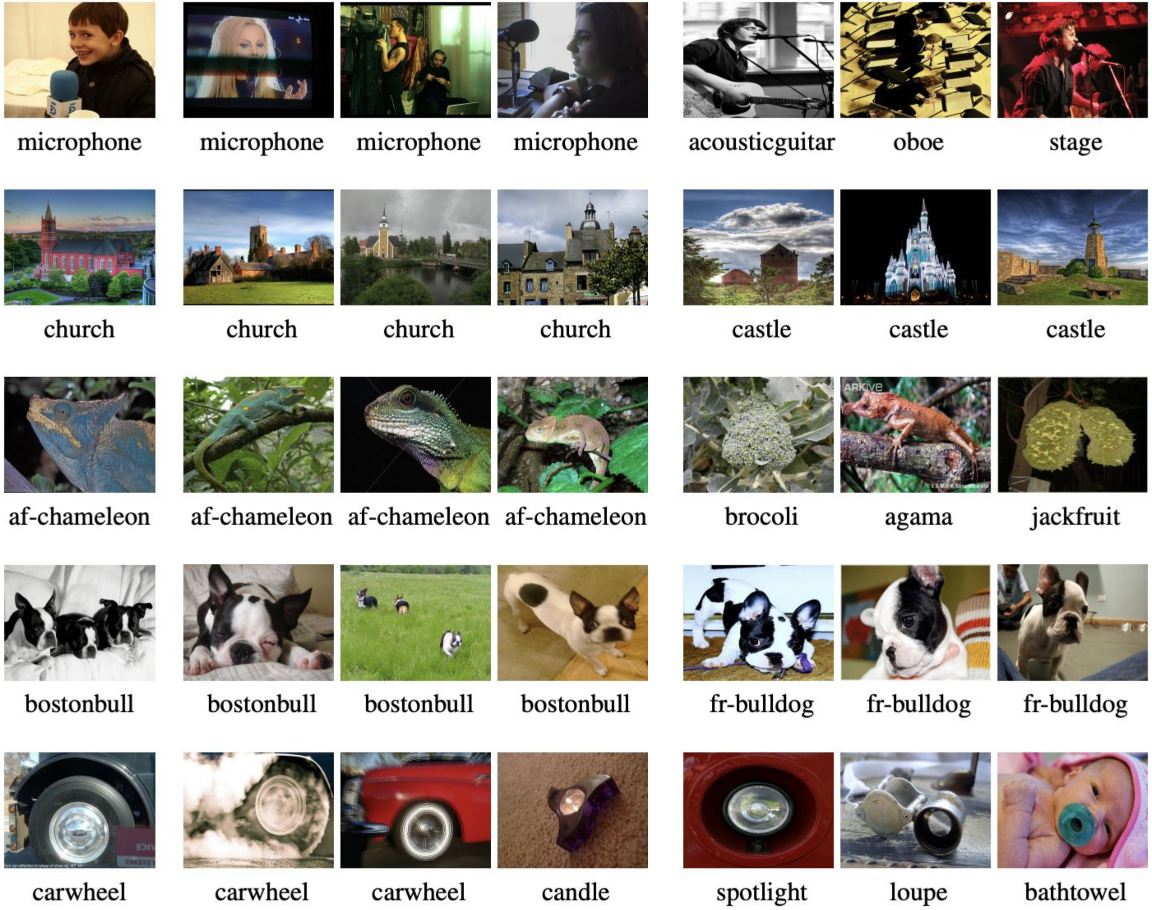


Figure 7: Top 5 influential points for the test point: 7196 (CIFAR-10). The model is a ResNet-18 trained with a weight-decay regularization; All the 5 training points are semantically similar to the test-point from the class "Airplane".

TraceInCP

Store check-points. Make influence of a training data point (*z*) on a test data point (*z'*):

$$\texttt{TracInCP}(z, z') = \sum_{i=1}^{k} \eta_i \nabla \ell(w_{t_i}, z) \cdot \nabla \ell(w_{t_i}, z')$$



Grad-Cos

Baseline method: Simply returns the cosine distance of the gradients of z and z'.

Note: Common alternative is Grad-Dot.

Evaluating Training Data Influence

Leave-one-out influence

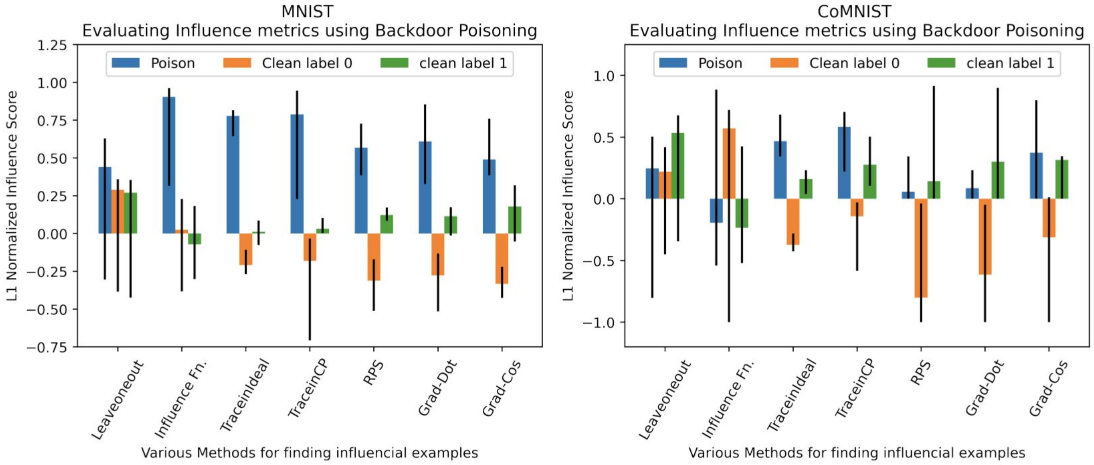
- Train a model for all n-1 subsets of your n-sized training data
- The influence of z is the difference in output on z' between the model trained on all data - and the model trained on all data but $\{z\}$
- Often considered gold standard

Heuristics

- The amount of the training data points that have themselves as most influential
- The amount of the test data points in class agreement with their most influential

Exercise: What's the problem with these heuristics? Think of a counter-example to both.

Backdoor poisoning attacks



(a) MNIST (b) CoMNIST

Take-home message

No good automated evaluation protocols

Next Lecture: The intricacies of human evaluation of rationales/explanations. As well as an equally two-sided story about **probing.**