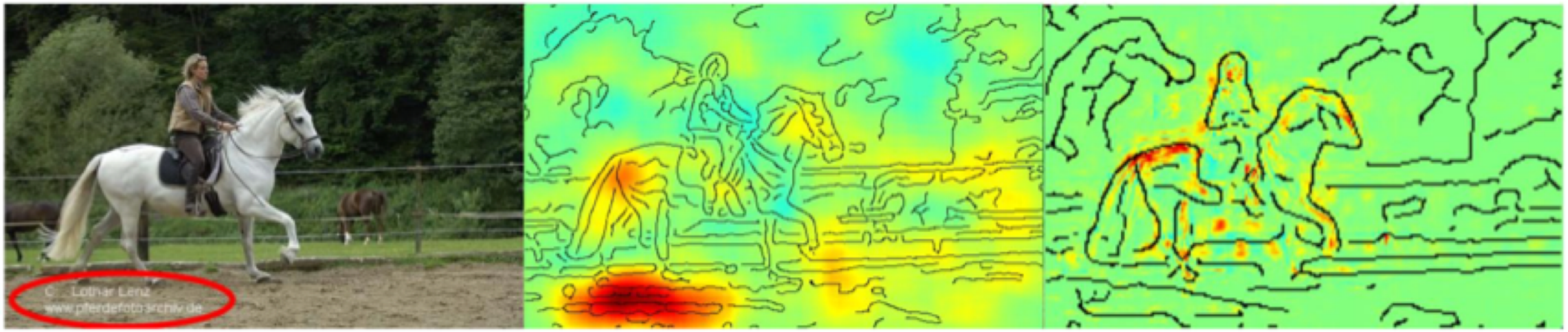
Deep Learning - 2019

Interpretability of Neural Models

Prof. Avishek Anand

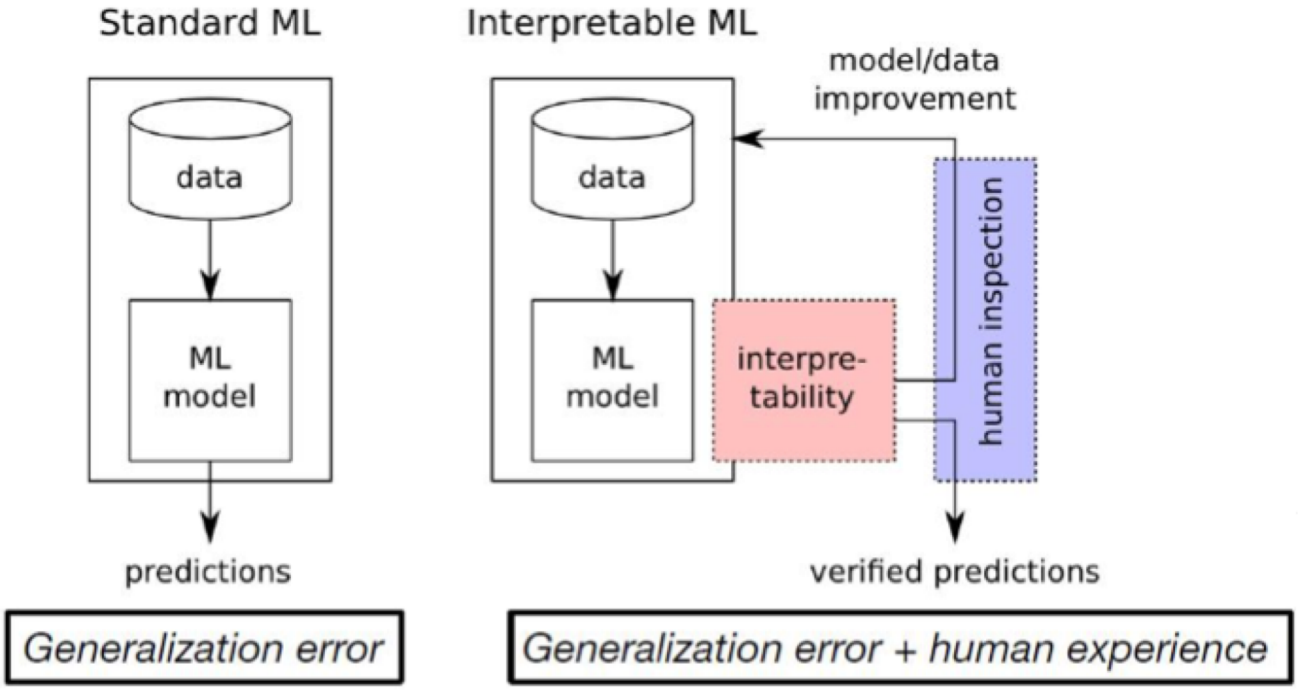
Why Interpretability ?

- Right for the Right reasons: a machine learning model is accurate and interpretable if
  - most of its predictions are correct and generalises well
  - and conforms to domain experts' knowledge about the problem



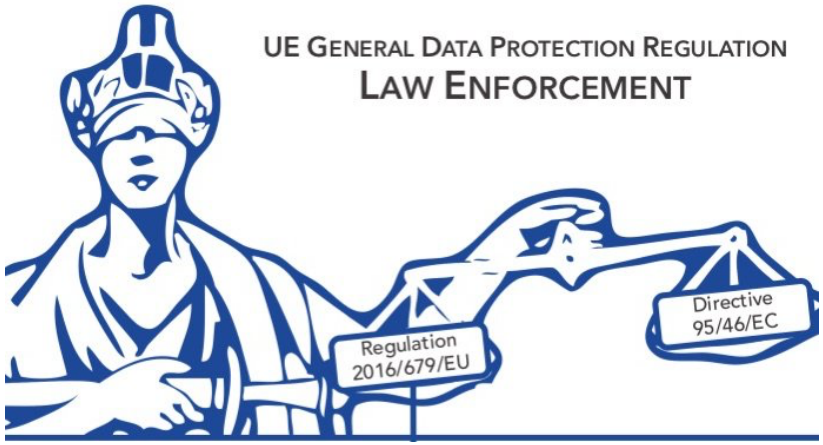
Why Interpretability ?

- Insights into understanding and improving the model:
  - Generalization error + human experience often results in better models



Why Interpretability ?

- Compliance for legislation: a machine learning model should explain itself
  - GDPR intends for “Right to Explanation”
  - Retain human decision to assign responsibility



Interpretability by Design

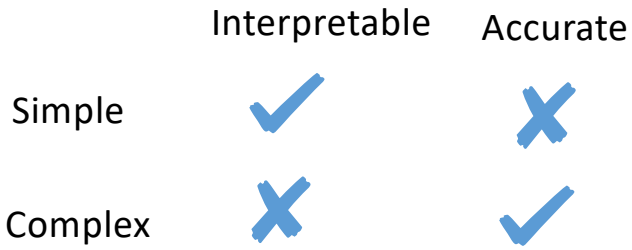
- Build an interpretable neural models from scratch
  - **Classical interpretable models**: Decision trees, Linear Models, etc

$$\hat{f}(x) = \beta_0 + \beta_1 x_1 + \ldots + \beta_p x_p$$

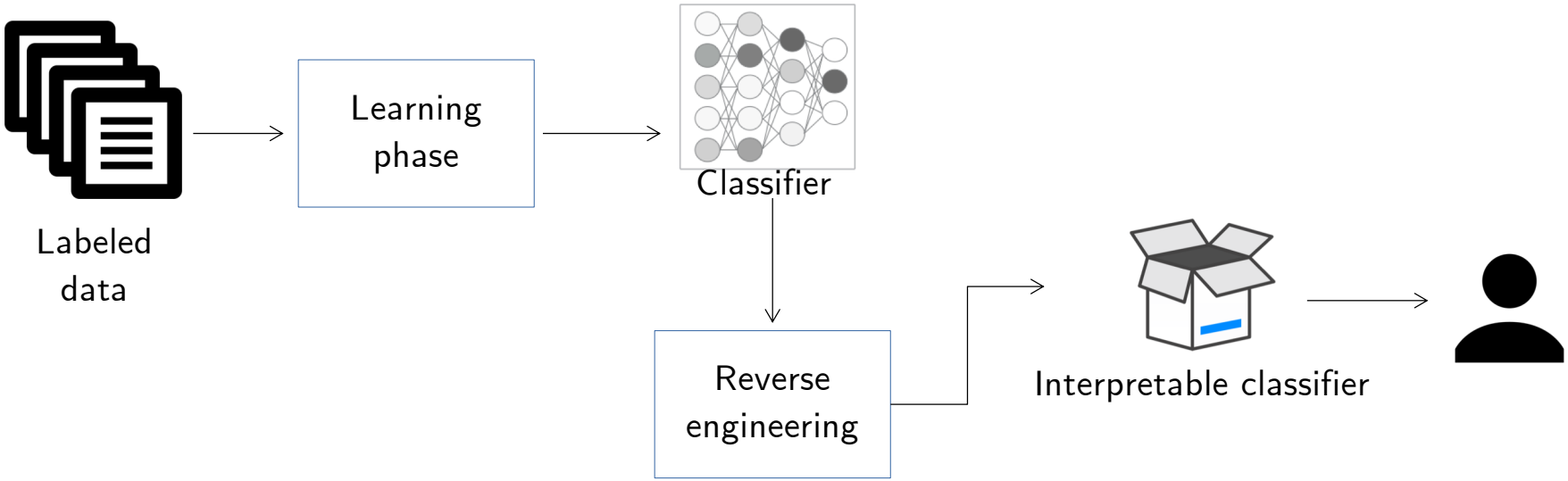
- It is well-known that deep models often have multiple optima of similar predictive accuracy
  - thus one might hope to find more interpretable models with equal predictive accuracy

- Add sparsity to the neural models
  - Sparse-auto encoders

- Add interpretability constraints
  - Regularize the loss function with interpretability penalties, tree-regularization



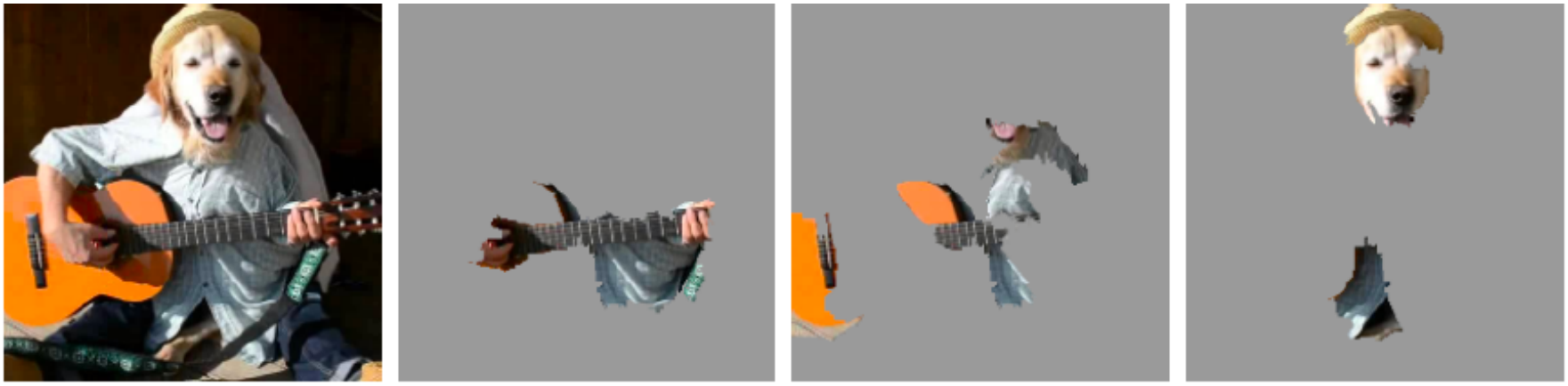
Post-hoc Interpretability



- Explain an already built model – Blackbox Model
- We can query the black box model infinite times
- We might/might not have access to all the learnt parameters

Model Agnostic vs Model Introspective

- No access to the model params but can query the model infinite times
- Can we explain the model in the Locality of the query ?
- Approach: Perturb the input to learn a locally interpretable model (more on this later)



(a) Original Image (b) Explaining *Electric guitar* (c) Explaining *Acoustic guitar* (d) Explaining *Labrador*

Model Agnostic vs Model Introspective

Given a trained model find the input/s that are most responsible for the output decision ?

- Approach: Sensitivity Analysis, Impact redistribution
  - Use gradient comptations on the fully specified complex function (say a neural model)

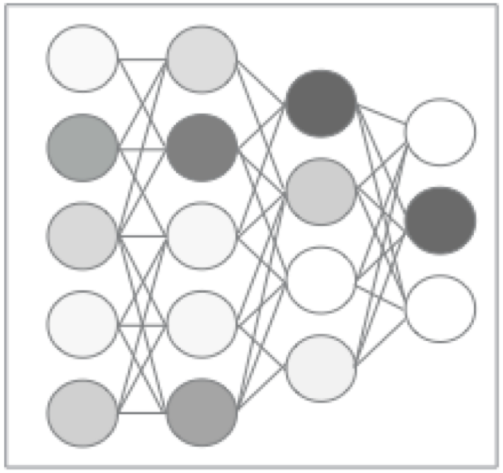
Input gradients `+soc.religion.christian` `+alt.atheism`

From: UST5012@uabdpo.dpo.uab.edu
Subject: Should teenagers pick a church parents don't attend?
Organization: UTexas Mail-to-News Gateway
Lines: 13

Q. Should teenagers have the freedom to choose what church they go to?

My friends teenage kids do not like to go to church.
If left up to them they would sleep, but that's not an option.
They complain that they have no friends that go there, yet don't attempt to make friends. They mention not respecting their Sunday school teacher, and usually find a way to miss Sunday school but do make it to the church service, (after their parents are thoroughly disgusted) I might add. A never ending battle? It can just ruin your whole day if you let it.

Has anyone had this problem and how did it get resolved?
f.



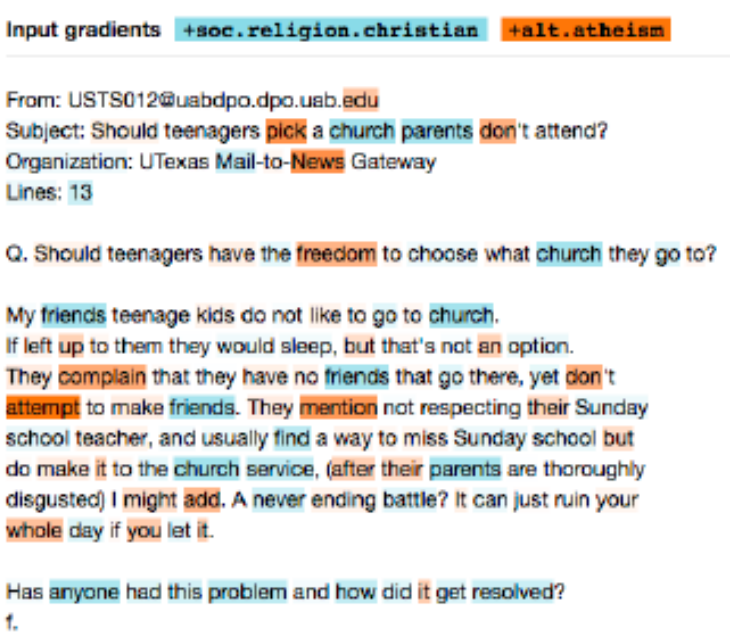
Sports

Religion

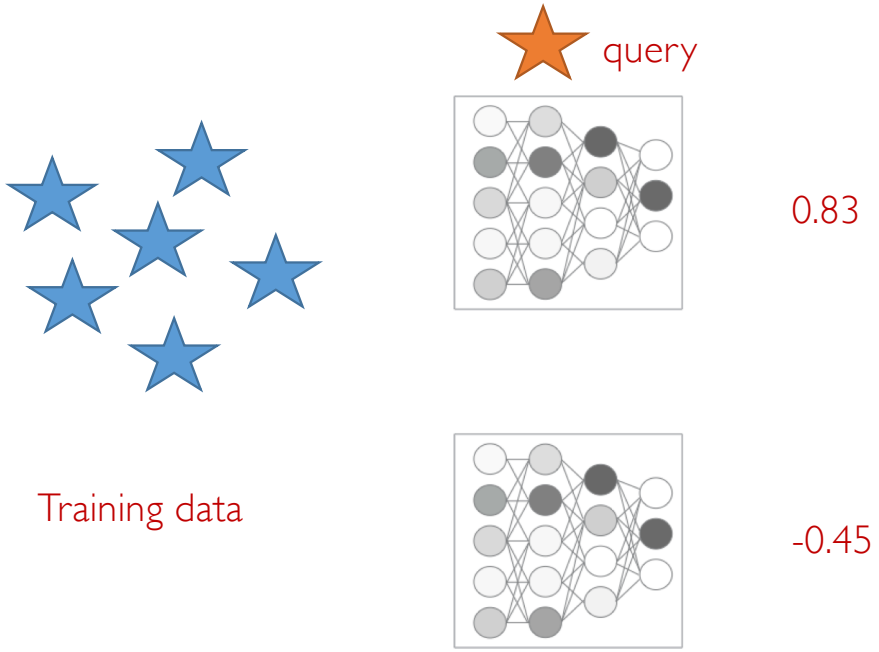
Politics

Feature Attribution vs Data Attribution

Which feature has the maximum impact on the decision ?



Which training instance has the maximum impact on the decision ?



Model Agnostic Approaches

- We are given a trained model but we have no access to model parameters
- Notion of interpretability
  - Build a surrogate model (Explainer) that approximates the BlackBox model
- Explainer should be easy to understand
  - Use Linear Models where all features are aggregated using linear combinations
- Explainer should operate on an human-understandable feature space
  - Use super-pixels or words as input feature space
- Is it indeed possible to approximate a complex function with a simple one ?
  - Maybe a global explainer is not possible but local explainers are possible

LIME

- Characteristics of an explainer
  - Must provide good understanding between the input variables and the response
  - local fidelity : explainers must correspond to how the model behaves around the instances being explained

- Assumes an interpretable feature space
  - Text classification:
    - Original space: word embeddings
    - Explanation space: presence or absence of a word (1-hot)

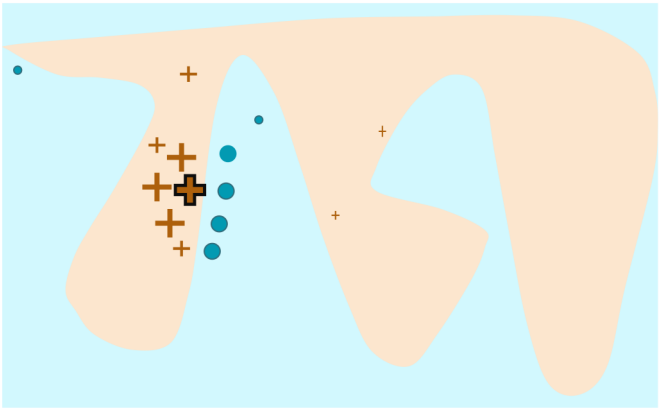
  - Image classification
    - Original space: a tensor with three color channels per pixel.
    - Explanation space: a binary vector indicating the “presence” or “absence” of a contiguous patch of similar pixels (a super-pixel)

LIME

- Key Insight: a complex function f(.) can be approximated by a simple model g(.) in the neighbourhood of the target instance x

- Create Local Training Dataset:
  - Sample instances around input to be explained x by sampling instances in its neighbourhood
  - Label using the original function f(.)

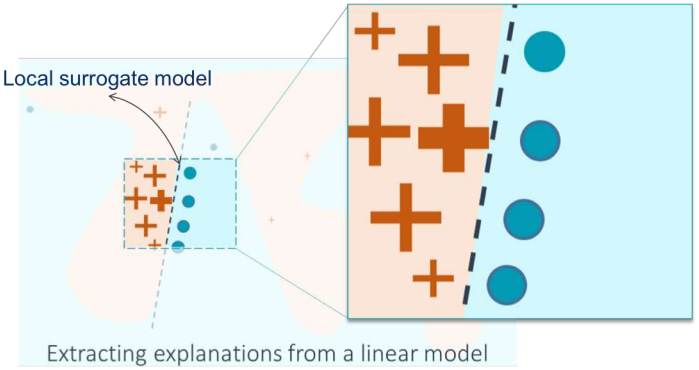
- Find the best explanation g(.) that minimized local loss



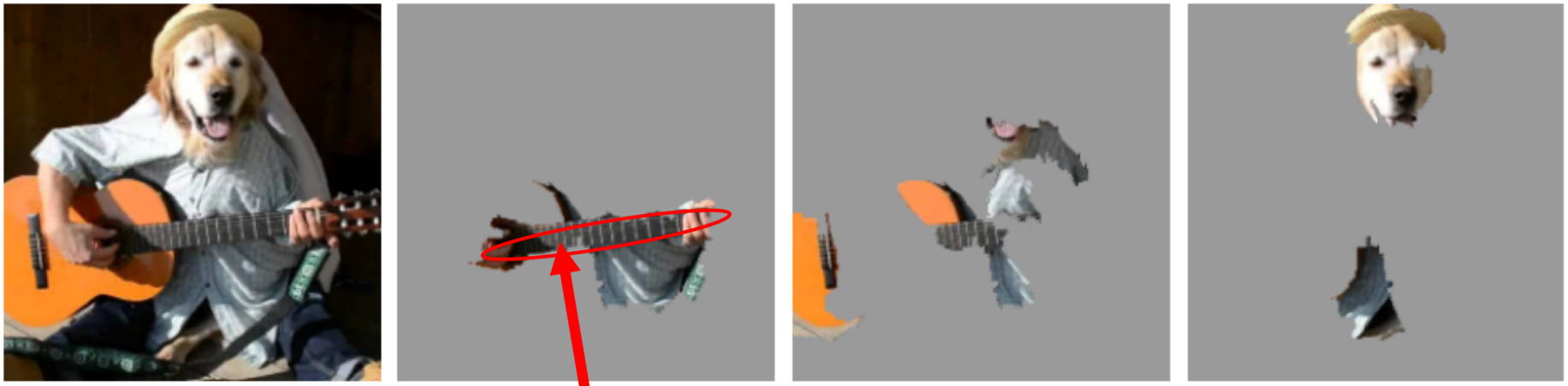
$$\xi(x) = \underset{g \in G}{\operatorname{argmin}} \mathcal{L}(f, g, \pi_x) + \Omega(g)$$

proximity measure between instances

complexity of explanation



Example



(a) Original Image (b) Explaining *Electric guitar* (c) Explaining *Acoustic guitar* (d) Explaining *Labrador*

The reason why Acoustic guitar was classified as Electric guitar

SHAP

- Approximate f(.) with g(.) where g(.) is a linear combination of feature contribution weights (shapely values)

$$g(z') = \phi_0 + \sum_{i=1}^{M} \phi_i z'_i, \quad \approx f(h_x(z'))$$

Shapley values

- The value of the j-th feature contributed ϕ_j to the prediction of this particular instance compared to the *average prediction* for the dataset

$$\hat{f}(x) = \beta_0 + \beta_1 x_1 + \ldots + \beta_p x_p$$

Shapley Values

$$\phi_j(val) = \sum_{S \subseteq \{x_1,\ldots,x_p\} \setminus \{x_j\}} \frac{|S|!(p-|S|-1)!}{p!} \left(\Delta\left(S \cup \{x_j\}\right) - \Delta(S)\right)$$

$$\Delta(S) = E\left[f\left(X_1,\ldots,X_p\right)|X_{i_1} = y_{i_1},\ldots,X_{i_s} = y_{i_i}\right] - E\left[f\left(X_1,\ldots,X_p\right)\right]$$

- Expected value of a function f(.) is the average of all predictions given all instances are equally likely
- Expected value of f(.) given *fixed values of certain features* is the average of all predictions where the features (from the instances where the features already take the fixed values)

- Given a subset of features S compute the effect/contribution of the subset S
- Shapely value of j = The contribution of a feature j to all possible subsets

Example Computation

$$\phi_j(val) = \sum_{S \subseteq \{x_1,\ldots,x_p\} \setminus \{x_j\}} \frac{|S|!(p-|S|-1)!}{p!} \left(\Delta\left(S \cup \{x_j\}\right) - \Delta(S)\right) \qquad \Delta(S) = E\left[f\left(X_1,\ldots,X_p\right)|X_{i_1} = y_{i_1},\ldots,X_{i_s} = y_{i_s}\right] - E\left[f\left(X_1,\ldots,X_p\right)\right]$$

- Let us try to explain the feature attribution for the instance {1,0}

X_1	X_2	$C(=f(X))$
0	0	0
1	0	1
0	1	1
1	1	1

$$E\left[f\left(X_1, X_2\right)\right] = \sum_{(x_1,x_2)\in\mathcal{X}} f\left(x_1, x_2\right) \cdot P\left(X_1 = x_1, X_2 = x_2\right) = \frac{1}{4}(0+1+1+1) = \frac{3}{4}$$

$$\Delta(\{X_1\}) = E\left[f\left(X_1, X_2\right)|X_1 = 1\right] - E\left[f\left(X_1, X_2\right)\right] = \frac{(1+1)}{2} - \frac{3}{4} = 1 - \frac{3}{4} = \frac{1}{4}$$

$$\Delta(\{X_2\}) = E\left[f\left(X_1, X_2\right)|X_2 = 0\right] - E\left[f\left(X_1, X_2\right)\right] = \frac{(1+0)}{2} - \frac{3}{4} = \frac{1}{2} - \frac{3}{4} = -\frac{1}{4}$$

$$\Delta(\{X_1, X_2\}) = E\left[f\left(X_1, X_2\right)|X_1 = 1, X_2 = 0\right] - E\left[f\left(X_1, X_2\right)\right] = \frac{1}{1} - \frac{3}{4} = \frac{1}{4}$$

$$\Delta(\emptyset) = 0$$

$$\varphi_1 = \frac{1}{2}\left[(\Delta(\{X_1\}) - \Delta(\emptyset)) + (\Delta(\{X_1, X_2\}) - \Delta(\{X_2\}))\right] = \frac{3}{8} \qquad \varphi_2 = \frac{1}{2}\left[(\Delta(\{X_2\}) - \Delta(\emptyset)) + (\Delta(\{X_1, X_2\}) - \Delta(\{X_1\}))\right] = -\frac{1}{8}$$

Estimating Shapely Values

Output: Shapley value for the value of the j-th feature

Required: Number of iterations M, instance of interest x, feature index j, data matrix X, and machine learning model f

For all m = 1,...,M:

- Draw random instance z from the data matrix X
- Choose a random permutation o of the feature values
- Order instance x: $x_o = (x_{(1)}, \ldots, x_{(j)}, \ldots, x_{(p)})$
- Order instance z: $z_o = (z_{(1)}, \ldots, z_{(j)}, \ldots, z_{(p)})$
- Construct two new instances
- With feature j: $x_{+j} = (x_{(1)}, \ldots, x_{(j-1)}, x_{(j)}, z_{(j+1)}, \ldots, z_{(p)})$
- Without feature j:$x_{-j} = (x_{(1)}, \ldots, x_{(j-1)}, z_{(j)}, z_{(j+1)}, \ldots, z_{(p)})$
- Compute marginal contribution: $\phi_j^m = \hat{f}(x_{+j}) - \hat{f}(x_{-j})$

Compute Shapley value as the average: $\phi_j(x) = \frac{1}{M} \sum_{m=1}^{M} \phi_j^m$

$$\hat{\phi}_j = \frac{1}{M} \sum_{m=1}^{M} \left(\hat{f}\left(x_{+j}^m\right) - \hat{f}\left(x_{-j}^m\right) \right)$$

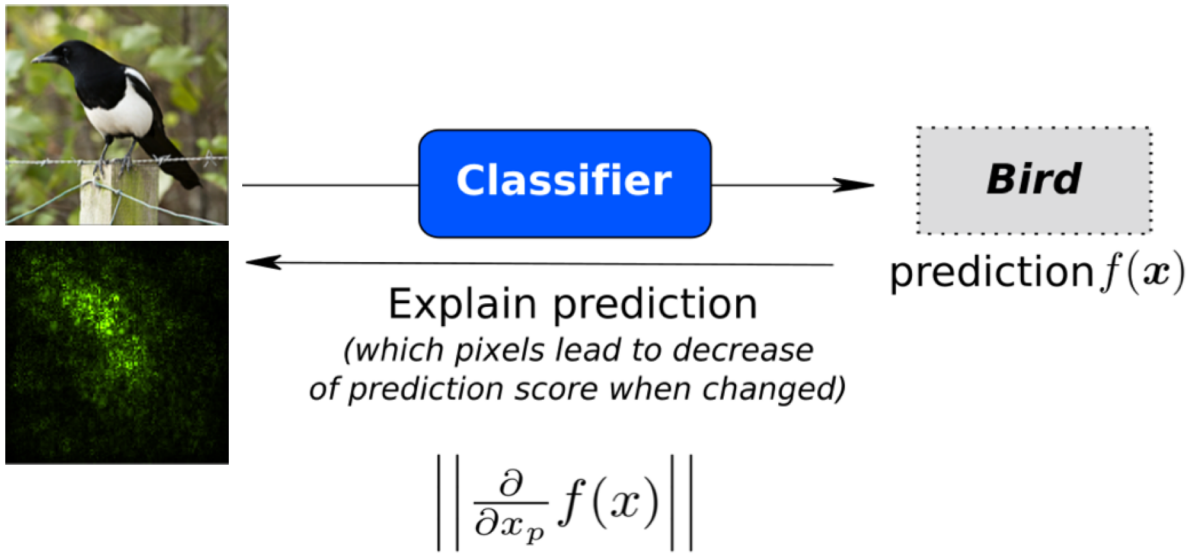
Example


dowitcher
red-backed_sandpiper
meerkat
mongoose
−0.006
−0.004
−0.002
0.000
0.002
0.004
0.006
SHAP value

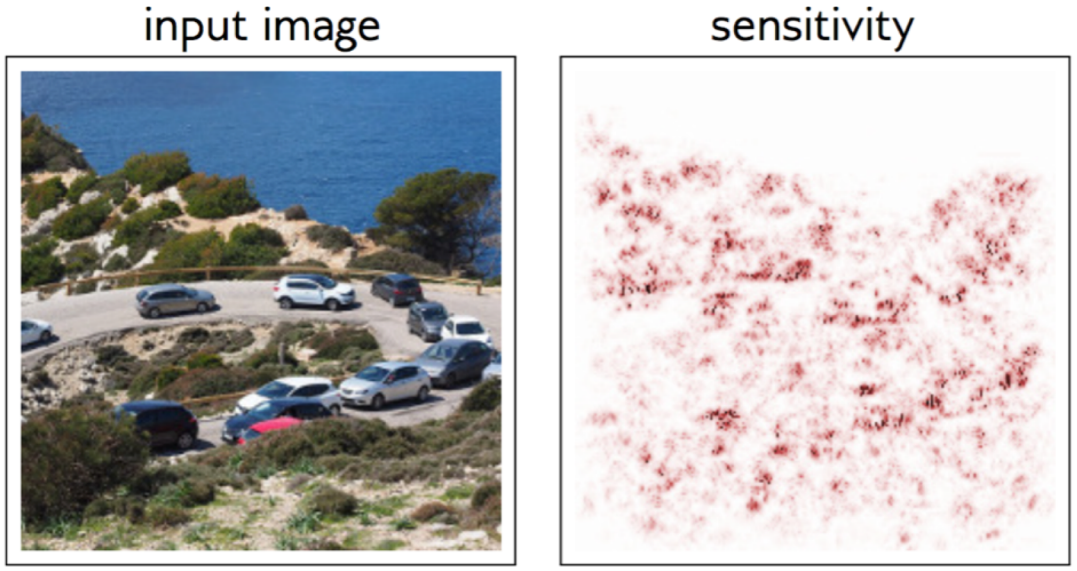
Model Introspective Methods

Sensitivity Analysis

- How sensitive is an input feature for the prediction ?
- Natural estimator is the gradient of the function w.r.t input feature
- Explanations are called **Saliency Maps**



Why Sensitivity Analysis does not work



- Explains the variation in the function but not the function itself
- Explains what reduces/increases the evidence for cars rather than the actual evidence for cars

Relevance Propagation

input x

Classifier

Rooster

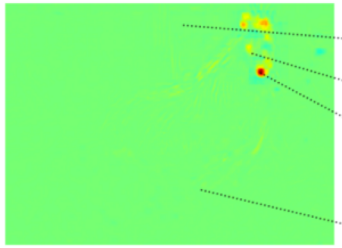
prediction $f(x)$

Explain prediction

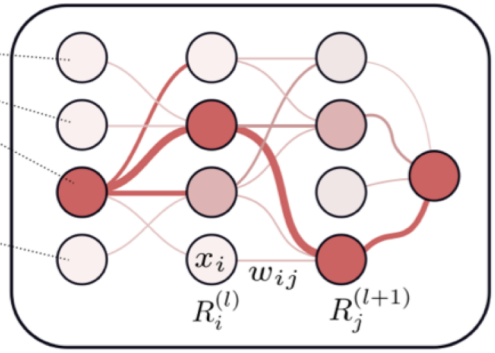
(how much each pixel contributes to prediction)

Idea: Decompose function

$$\sum_i R_i = f(x)$$



heatmap



"every neuron gets it's share of relevance depending on activation and strength of connection."

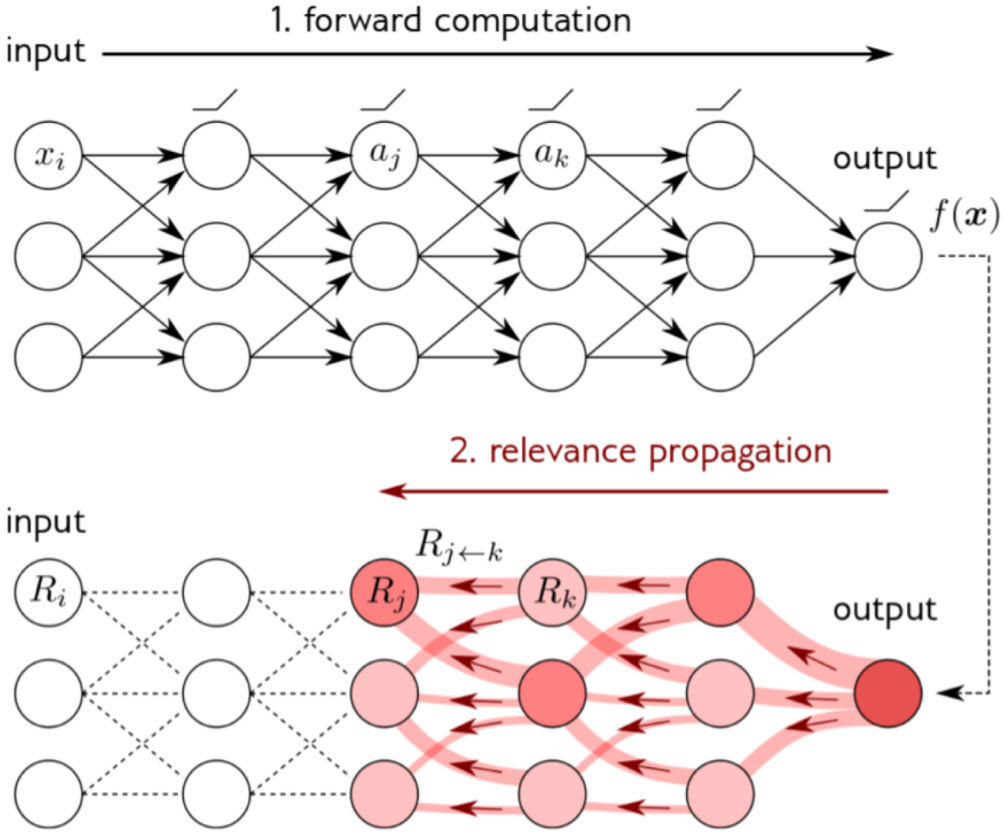
redistribute $f(x)$

LRP – Layer-wise Relevance Propagation

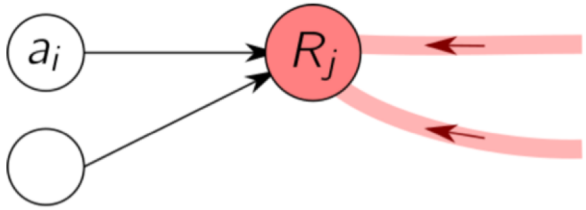
$$R_i = \sum_j \frac{a_i w_{ij}^+}{\sum_i a_i w_{ij}^+} R_j$$

$$R_j = a_j c_j$$

$$R_i = a_i \sum_j w_{ij}^+ \frac{\max\left(0, \sum_i a_i w_{ij}\right)}{\sum_i a_i w_{ij}^+} C_j$$



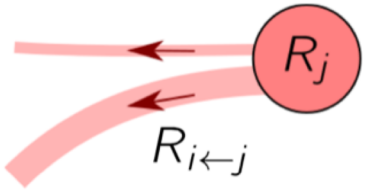
Deep Taylor Decomposition (Optional)



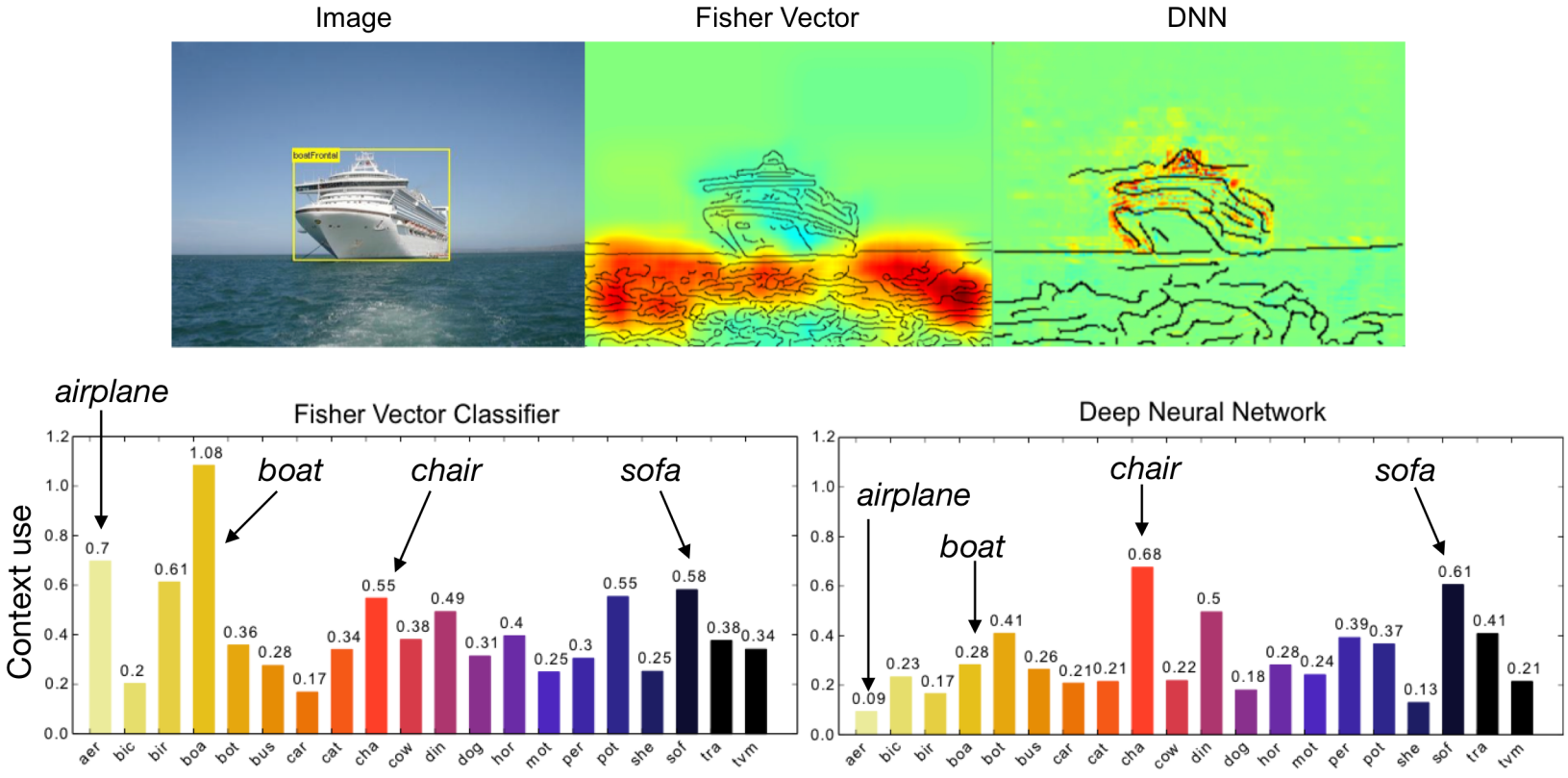
$$
\begin{aligned}
R_j &= a_j c_j \\
&= \max\left(0, \sum_i a_i w_{ij}\right) \cdot c_j \\
&= \max\left(0, \sum_i a_i w'_{ij}\right) \qquad w'_{ij} = w_{ij} c_j
\end{aligned}
$$

$$
R_j\left((a_i)_i\right) = R_j\left((\widetilde{a}_i)_i\right) + \sum_i \underbrace{\left.\frac{\partial R_j}{\partial a_i}\right|_{(\widetilde{a}_i)_i} \cdot (a_i - \widetilde{a}_i)}_{R_{i \leftarrow j}} + \varepsilon
$$

$$
R_{i \leftarrow j} = \frac{\left(a_i - \widetilde{a}_i^{(j)}\right) w_{ij}}{\sum_i \left(a_i - \widetilde{a}_i^{(j)}\right) w_{ij}} R_j
$$



Examples



Large values indicate importance of context

(Lapuschkin et al. 2016)

References

- https://christophm.github.io/interpretable-ml-book/

- **LIME:** Ribeiro, M.T., Singh, S. and Guestrin, C., 2016, August. 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* (pp. 1135-1144). ACM.

- **SHAP:** Lundberg, S.M. and Lee, S.I., 2017. A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems* (pp. 4765-4774).

- **LRP:** Bach, Sebastian, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. "On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation." *PloS one* 10, no. 7 (2015): e0130140.

- **Influence Functions:** Koh, Pang Wei, and Percy Liang. "Understanding black-box predictions via influence functions." In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 1885-1894. JMLR. org, 2017.