STOR566: Introduction to Deep Learning

Lecture 22: Explainability of ML Models

Yao Li
UNC Chapel Hill

Nov 15, 2022

Materials are from *Deep Learning (UCLA)*

Motivations

Why Explainability

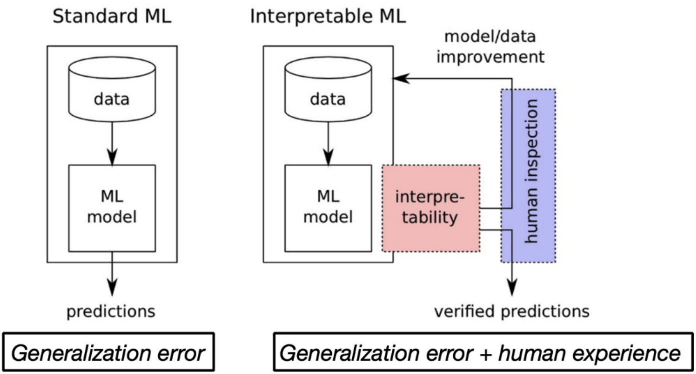
- Understand predictions/decisions of machine learning models

Top label: **"clog"**

Why did the network label this image as **"clog"**?

Why Explainability

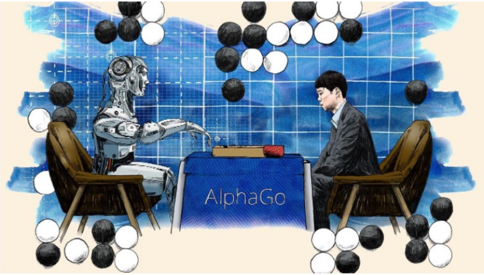
- Improve machine learning models

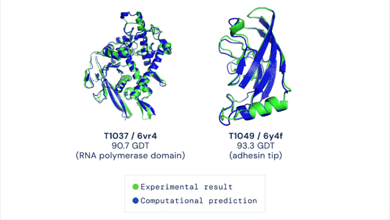


Credit: Samek, Binder, Tutorial on Interpretable ML, MICCAI'18

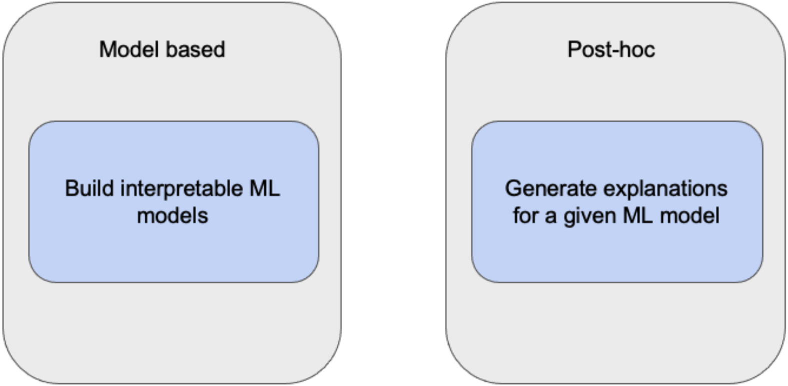
Why Explainability

- Learn new insights





Approaches



today's focus

Approaches

What is an "explanation"?

- Feature attribution

  The model makes this prediction because of feature (pixel) X
- Data attribution

  The model makes this prediction because of which training data
- Surrogate model

  Approximate the complex model using a simple explainable surrogate model

Approaches

What is an "explanation"?

- Feature attribution

  The model makes this prediction because of feature (pixel) X

- Data attribution

  The model makes this prediction because of which training data

- Surrogate model

  Approximate the complex model using a simple explainable surrogate model

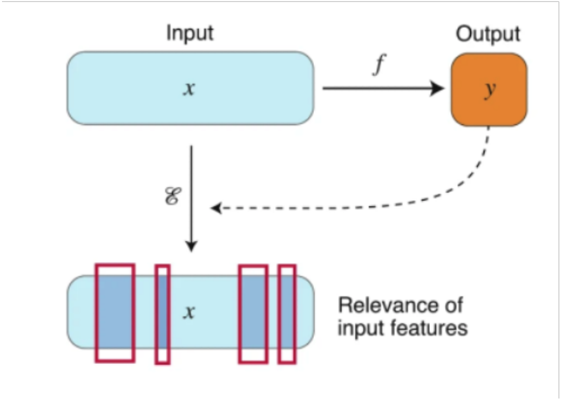
We focus on the first two types in today's lecture.

Feature Attribution

What is Feature Attribution?

- Given a model f and input $\boldsymbol{x}$
- Assign a relevant score to each input feature

  R_i: how much does feature i contributes to the prediction



Perturbation-based Analysis

- Assumption: Feature i is important $\rightarrow$ Perturbing $\boldsymbol{x}_i$ (the i-th feature of input $\boldsymbol{x}$) will significantly change $f(\boldsymbol{x})$
- Therefore, assign relevant score for each feature by

$$R_i \leftarrow f(\boldsymbol{x}) - f(\boldsymbol{x} + \delta \boldsymbol{e}_i),$$

R_i: importance score of i-th input feature

$\boldsymbol{e}_i$: vector of zeros except that the i-th element is one. The dimension is the same as $\boldsymbol{x}$.

Perturbation-based Analysis

- Assumption: Feature i is important $\rightarrow$ Perturbing $\boldsymbol{x}_i$ (the i-th feature of input $\boldsymbol{x}$) will significantly change $f(\boldsymbol{x})$
- Therefore, assign relevant score for each feature by

$$R_i \leftarrow f(\boldsymbol{x}) - f(\boldsymbol{x} + \delta \boldsymbol{e}_i),$$

  R_i: importance score of i-th input feature

  $\boldsymbol{e}_i$: vector of zeros except that the i-th element is one. The dimension is the same as $\boldsymbol{x}$.

- Questions:
  - How to choose perturbation δ?
  - Efficiency: may need $O(d)$ function evaluations. d is the dimension of $\boldsymbol{x}$.
  - Can this capture the correlation between features?

Linear Model

- Linear prediction

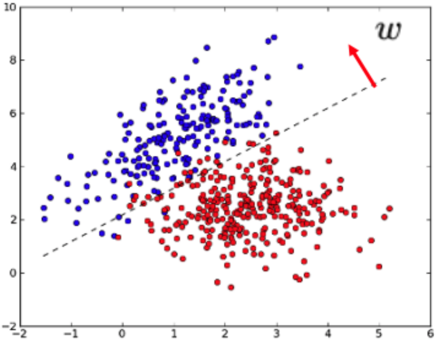
$$f(\boldsymbol{x}) = \boldsymbol{w}^T\boldsymbol{x}$$
$$= w_1x_1 + w_2x_2 + \ldots + w_dx_d$$

- For any fixed perturbation

$$f(\boldsymbol{x}) - f(\boldsymbol{x} + \delta\boldsymbol{e}_i) = \delta w_i \propto w_i$$

- Feature importance:

$$R_i \leftarrow w_i$$



Gradient

- Consider the case when $\delta \to 0$

$$\lim_{\delta \to 0} f(\boldsymbol{x}) - f(\boldsymbol{x} - \delta \boldsymbol{e}_i) = \frac{\partial}{\partial x_i} f(\boldsymbol{x})$$

- Therefore, we can use gradient to measure importance of each feature

$$R_i \leftarrow \frac{\partial}{\partial x_i} f(\boldsymbol{x})$$

(Usually set f as the logit of the target model)

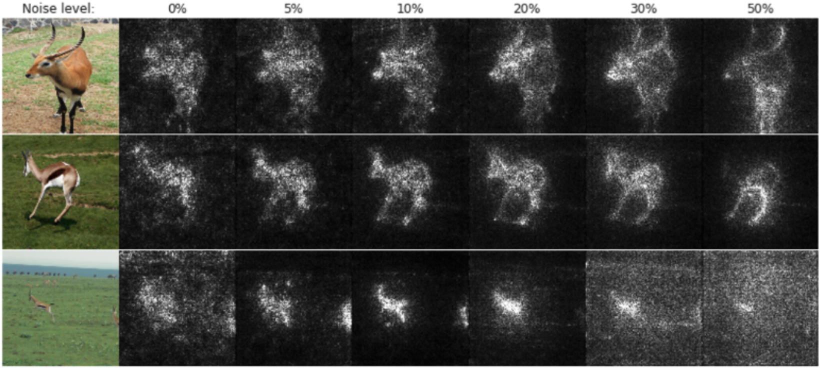
- Saliency map: visualize pixels with positive gradients

Simonyan et al., Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps. 2013.

Smoothed Gradient

- Gradient maps are often noisy (visually)
- Smoothed gradient:

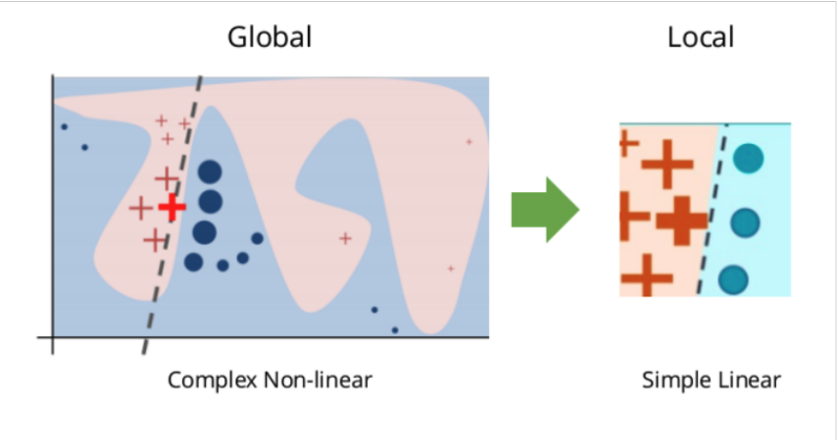
$$R(\boldsymbol{x}) = E_{z \sim N(0,\sigma)} \nabla f(\boldsymbol{x} + \boldsymbol{z})$$



Smilkov et al., SmoothGrad: removing noise by adding noise. 2017

LIME

- Build a local linear model, but not as local as gradient.



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

LIME

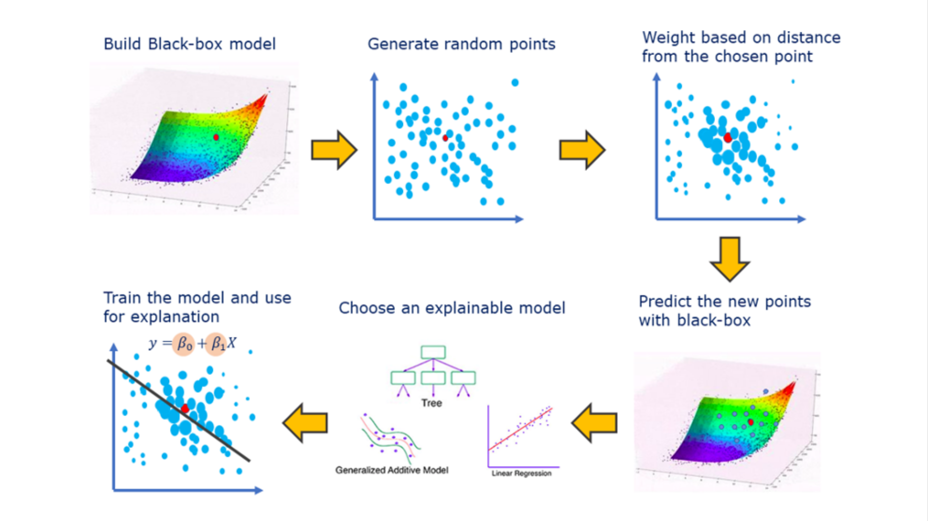
- Find a local linear model to mimic target (complex) model

$$\arg\min_{\underset{\text{Family of simple models}}{g \in G}} L(\underset{\text{Target model}}{f}, \underset{\text{Simple model}}{g}, \underset{\text{Sample weights}}{\pi_x}) + \underset{\text{Regularization}}{\Omega(g)}$$

- Sample weights: more weights to local samples

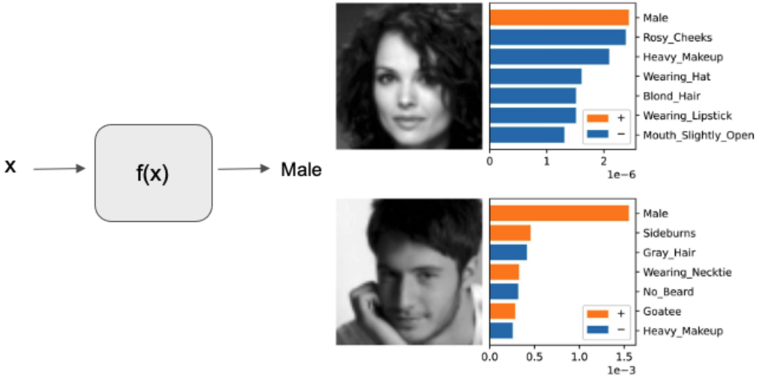
LIME

- Overview



Concept-based Explanations

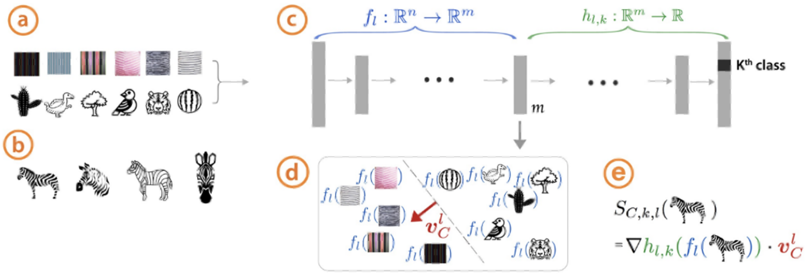
- Attribution to raw features may not be human understandable
- Can we attribute the prediction to high-level concepts instead of low-level features (pixels)?



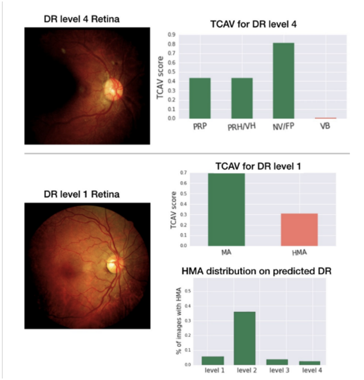
Kim et al., Interpretability Beyond Feature Attribution: Quantitative Testing with Concept Activation Vectors (TCAV). ICML, 2018.

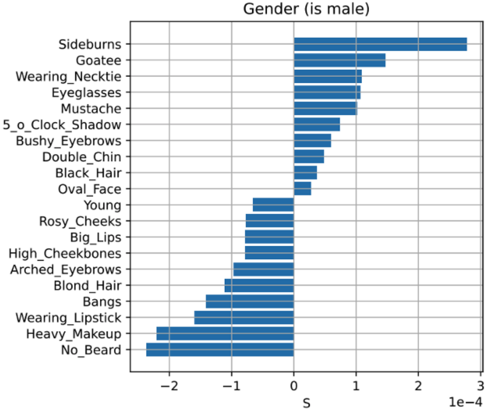
TCAV

- A concept can be given as positive/negative instances
- Assume layer l in DNN captures the concept $c = < v_C^l, x^l >$
- Then attribution to concept is the product of gradient $\frac{\partial f}{\partial x^l}$ and v_C^l



- Experimental results

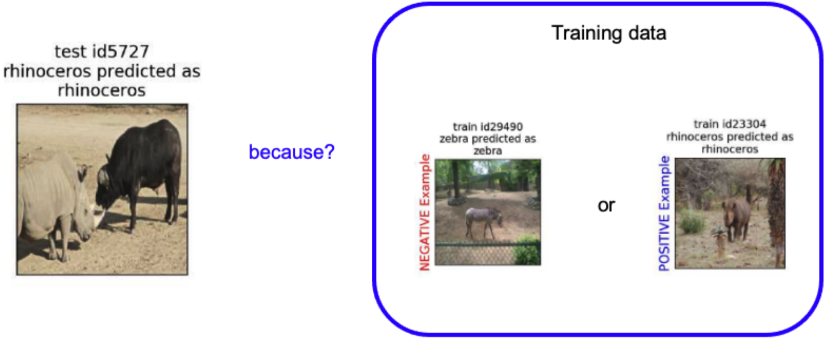




Data Attribution

Explaining by Training Data

- Which training data causes the prediction?



Influence Function

- What's the relationship between training data and model?

$$\hat{\theta} = \arg\min_{\theta} \frac{1}{n} \sum_{i=1}^{n} L(\boldsymbol{x}_i, \theta)$$

$\hat{\theta}$: model parameters

$\boldsymbol{x}_i$: training sample i

Influence Function

- What's the relationship between training data and model?

$$\hat{\theta} = \arg\min_{\theta} \frac{1}{n} \sum_{i=1}^{n} L(\boldsymbol{x}_i, \theta)$$

  $\hat{\theta}$: model parameters

  $\boldsymbol{x}_i$: training sample i

- Each training sample contributes to the model equally ($\frac{1}{n}$ is the weight of each sample)
- Can we compute the influence when the weight of a training sample slightly increased or decreased?

Koh et al., Understanding Black-box Predictions via Influence Functions. ICML, 2017.

Influence Function

- Assume adding more weight (ϵ) to $\boldsymbol{x}_j$ (the j-th training sampe)
- The model will become

$$\hat{\theta}_{\epsilon,\boldsymbol{x}_j} = \arg\min_{\theta} \frac{1}{n} \sum_{i=1}^{n} L(\boldsymbol{x}_i, \theta) + \epsilon L(\boldsymbol{x}_j, \theta)$$

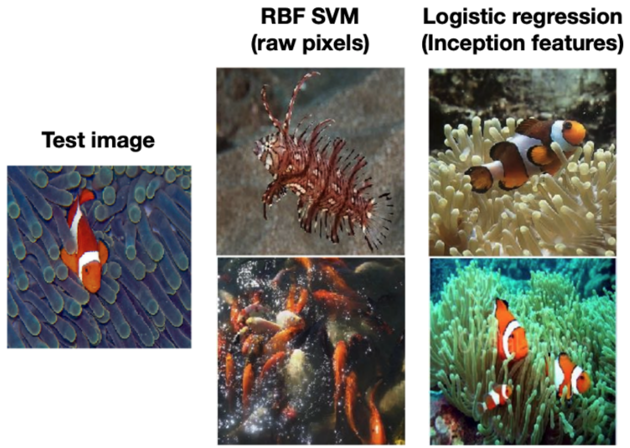
- Gradient of loss w.r.t. ϵ:

$$\begin{aligned} I_{\text{up,loss}} &:= \left. \frac{dL(\boldsymbol{x}_{test}, \hat{\theta}_{\epsilon,\boldsymbol{x}_j})}{d\epsilon} \right|_{\epsilon=0} \\ &= -\nabla_\theta L(\boldsymbol{x}_{test}, \hat{\theta})^T H_{\hat{\theta}}^{-1} \nabla_\theta L(\boldsymbol{x}_j, \hat{\theta}) \end{aligned}$$

$\boldsymbol{x}_{test}$: a test sample
$H_{\hat{\theta}} := \frac{1}{n} \nabla_\theta^2 L(\boldsymbol{x}_i, \hat{\theta})$

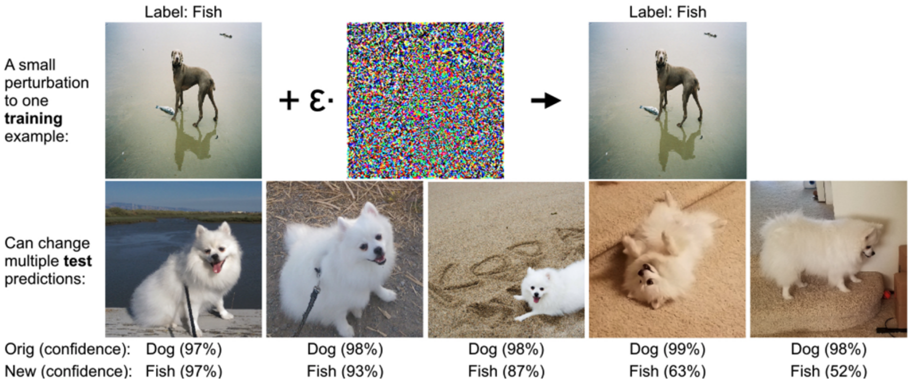
Influence Function

- Experimental results



Influence Function

- Can be used for poisoning attack to identify important training samples.

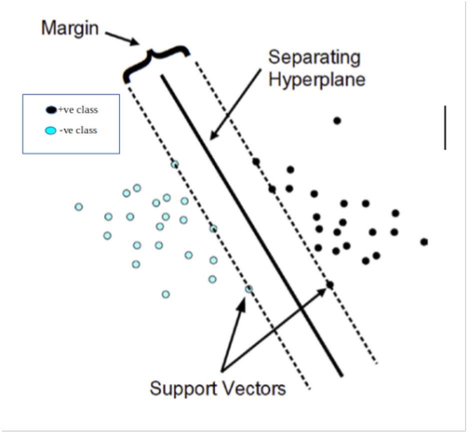


Representer Theorem in Linear Models

- Linear model: $f(\boldsymbol{x}) = \boldsymbol{w}^T\boldsymbol{x}$
- Representer theorem: model can be decomposed with training samples

$$\boldsymbol{w} = \sum_{i=1}^{n} \alpha_i \boldsymbol{x}_i$$

- SVM: $\alpha_i \neq 0$ support vectors
- General: represent the importance of each sample



Representer Theorem in Linear Models

- For a test sample: $\boldsymbol{x}_{test}$:

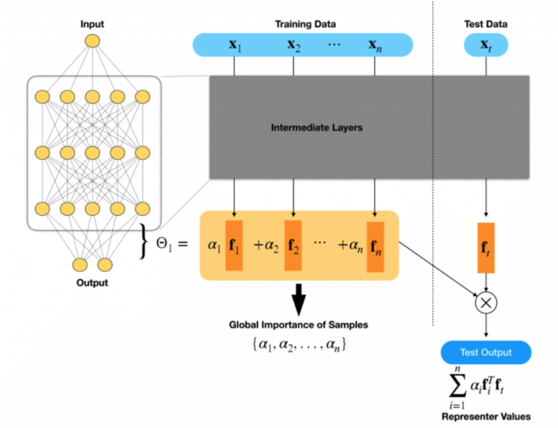
$$f(\boldsymbol{x}_{test}) = \boldsymbol{w}^T \boldsymbol{x}_{test} = \sum_{i=1}^{n} \alpha_i \boldsymbol{x}_i^T \boldsymbol{x}_{test}$$

- $\alpha_i \boldsymbol{x}_i^T \boldsymbol{x}_{test}$: importance of sample $\boldsymbol{x}_i$ in the final prediction based on $\boldsymbol{x}_{test}$
- A natural way to attribute prediction to each training sample

Representer Points in DNN

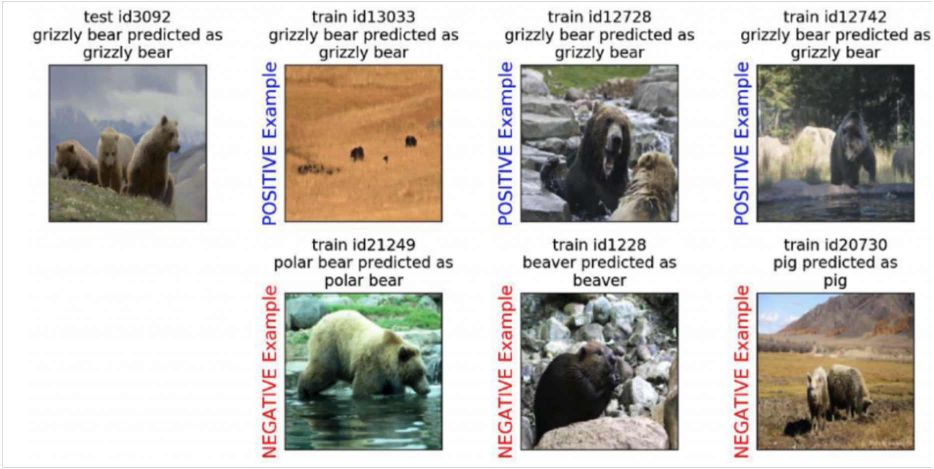
- Consider the final hidden layer output for each training sample: $f_i := f(\boldsymbol{x}_i), i = 1, ..., n$
- Applying representer theorem to the final linear layer: $\Theta_1 = \sum_{i=1}^{n} \alpha_i f_i$
- Attribute the prediction by $F(\boldsymbol{x}_t) = \sum_{i=1}^{n} \alpha_i {f_i}^T f_t$

  f_t: final layer output based on test sample $\boldsymbol{x}_t$

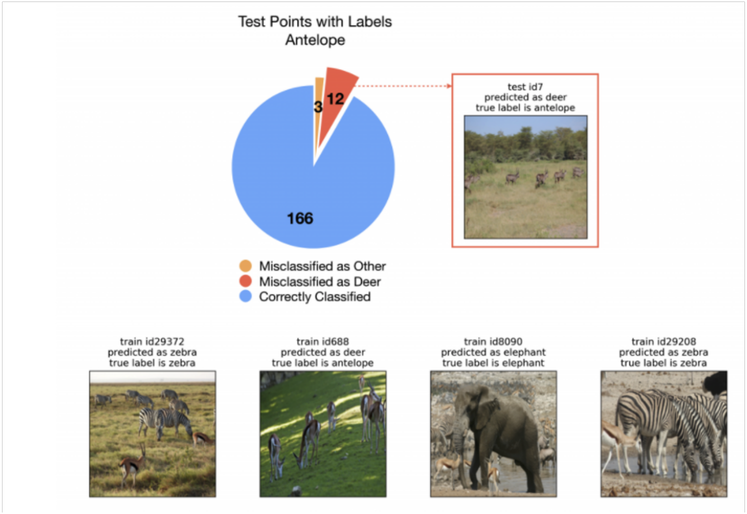


Yeh et al., Representer Point Selection for Explaining Deep Neural Networks. NeurIPS, 2018.

Representer points in DNN



Representer points in DNN



Conclusions

- Introduction to explainable ML
- Feature attribution
- Data attribution

Questions?