



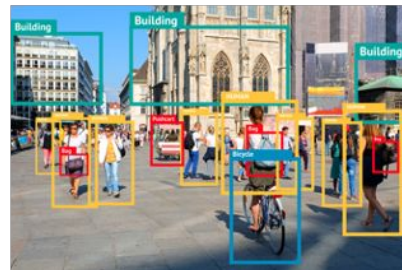
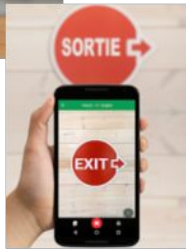
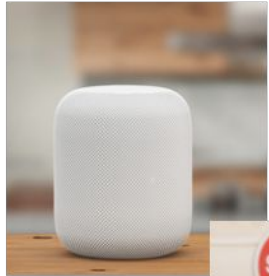
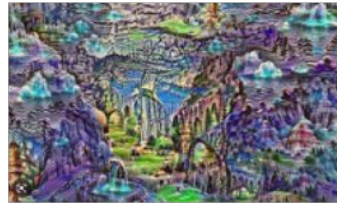
Explainable AI in a Nutshell

Jasmina Bogojeska

Centre for AI (CAI) @ZHAW
jasmina.bogojeska@zhaw.ch

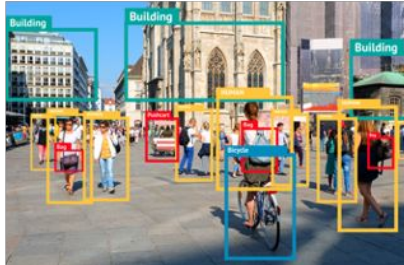
MDD & ARMADA Summer School 2026

AI Applications



Explainability Importance

Is explainability necessary for all these applications?



When is AI explainability necessary?

- Not necessary for all applications (product or friend recommendations)
 - The consequences of wrong decisions are really small or negligible
 - Problems are well studied and extensively validated in real-world applications on large amounts of data
- Vital in AI employed in complex high-stakes decision making settings
 - Impact on human lives, health, finances
 - High prediction quality on its own is no longer sufficient and **additional requirements are very important**: right to explanation, safety, non-discrimination
 - Additional key requirements are hard to quantify (as optimization objectives): safety (not possible to list all scenarios violating safety of autonomous car), scientific knowledge (abstract goals), ethics (discrimination unknown beforehand and too abstract to be encoded)
 - **Explainability**: if the system can explain its decisions, we can verify them with respect to the additional key requirements

Healthcare



Finance

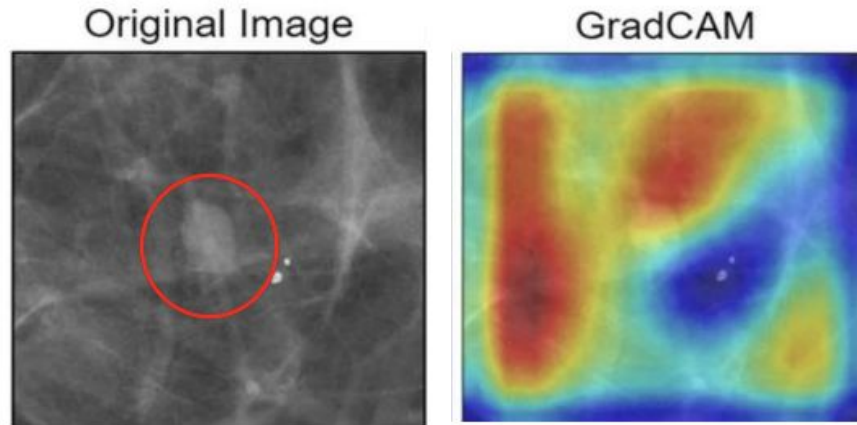


Criminal Justice System



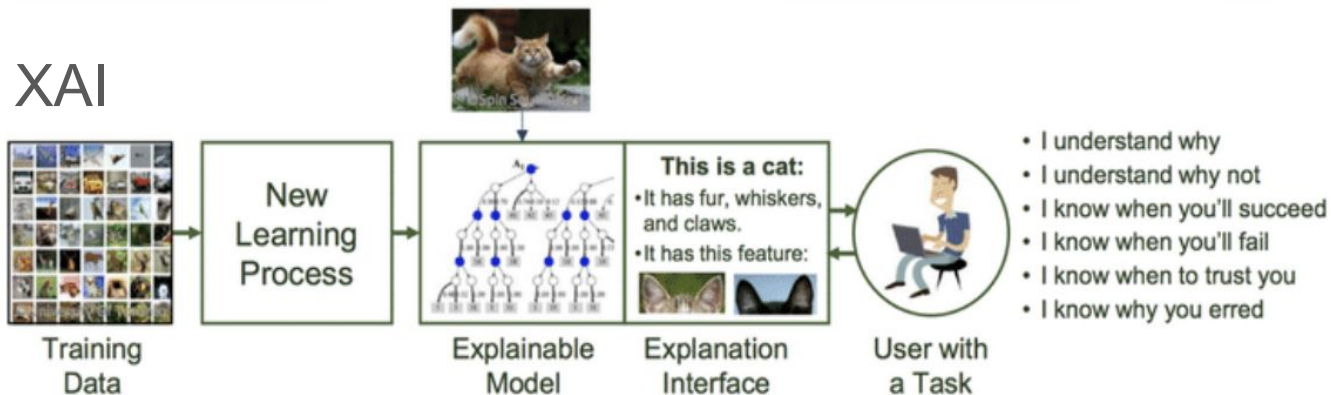
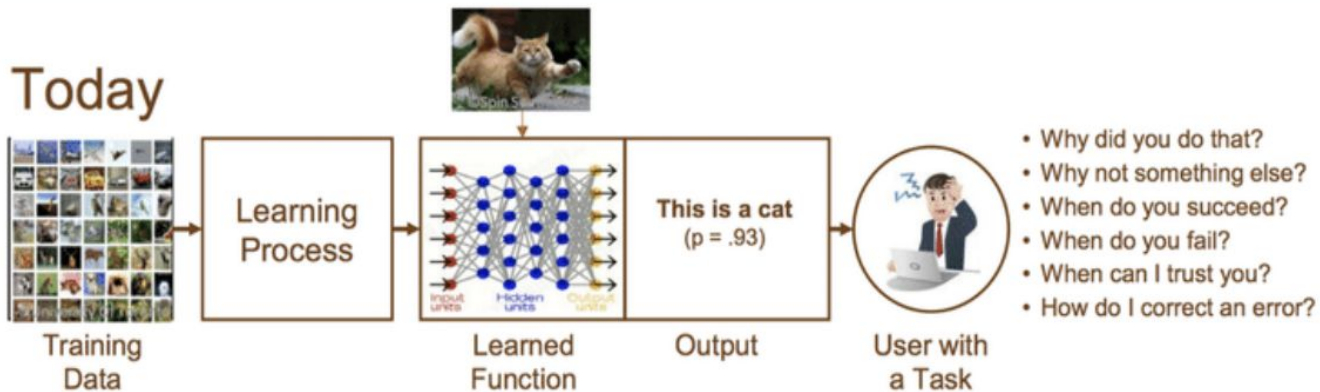
Explainability Importance Example

Deep neural networks in computer-aided mammography mostly focused on healthy tissue rather than tumour when predicting benign vs. malignant tumors.



Source: Barnett et al. "A case-based interpretable deep learning model for classification of mass lesions in digital mammography." *Nature Machine Intelligence* (2021).

The Ultimate Goal



Overview

XAI Introduction and motivation

Various explainability approaches

Evaluating Model Explanations

Practical tutorials

XAI Questions

- What does the prediction mean?
- How did the model make a prediction?
- Which features contributed to a certain prediction and how?
- How can a model learn or select features that are the most interpretable or informative?
- How much does each sample contribute to model training?

Motivation: Why do we need interpretability?

- Facilitates **debugging, safety, deployment**
 - Does the AI system base its decision on the correct features?
 - Is it suitable for real-world deployment?
- Promotes **trust**
 - Assess if and when to trust model predictions when taking decisions
- Promotes **fairness**
 - Detecting and removing biases
 - Is the AI system fair and non-discriminatory?
- Provide **recourse** to individuals affected by the model decisions
- Discover **new insights** from data

Motivation: Why do we need interpretability?

- **Improves AI usability:** XAI for actionable decision making
- **Human-AI collaboration:** XAI for better control and user interaction
- **Assessment of compliance with legislation:** Anti-discrimination laws, GDPR (General Data Protection Regulations), EU AI Act

General Data Protection Regulations (GDPR, 2016):

- "The data subject shall have the right not to be subject to a **decision** based solely on **automated processing**, including profiling,..." (Art. 22)
- The data subject has the right to "**meaningful information** about the **logic** involved" in the decision. (Art. 13 and 15)

EU AI Act (2024):

- "Any affected person subject to a **decision** which is taken by.. a **high-risk AI system** ... shall have the right to obtain from the deployer **clear and meaningful explanations** (Art. 86)

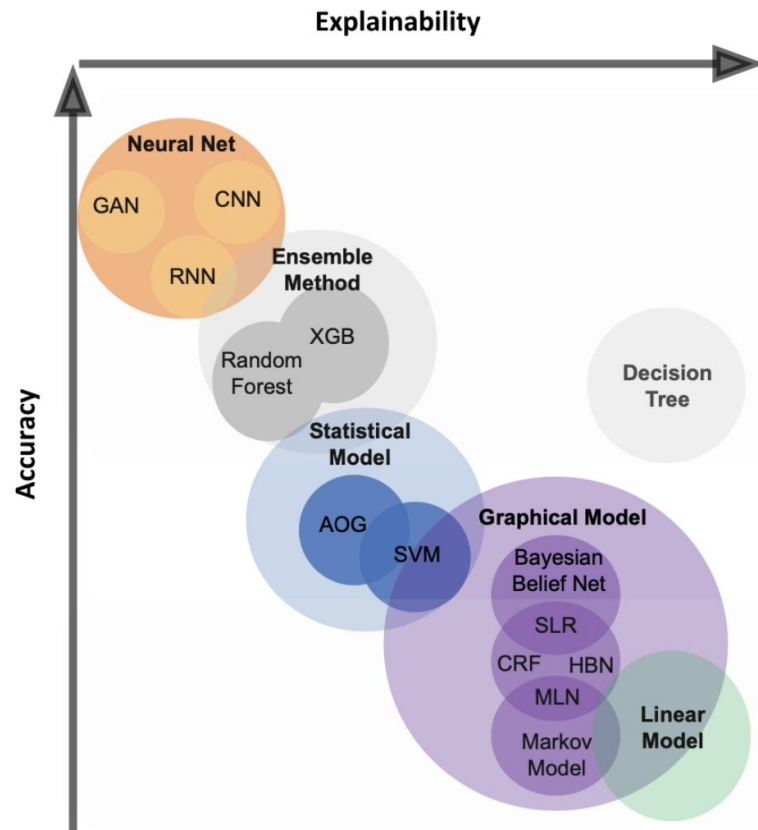
Useful for decision makers, end users, researchers, engineers, regulatory agencies...

Definitions

- **Explainability:** The extent to which a human can understand the reasoning behind an AI model's decisions or predictions. It involves providing clear and understandable explanations of how the model arrives at its outputs. An explanation is an interface between humans and a model that is both an accurate proxy of the model and comprehensible to humans. Explainability is the ability to provide human-understandable explanations for a model's behavior or predictions, whether or not the model itself is interpretable.
- **Interpretability:** The degree to which a human can comprehend the internal mechanics and structure of an AI model. Interpretable models are typically simpler, allowing users to grasp how inputs are transformed into outputs.
- **Transparency:** The openness and visibility into how an AI system operates, including access to its design, data and their preprocessing, algorithms, objective functions, decision-making processes, documentation.

ML Models Taxonomy

- Inherently explainable/glass box models:
 - Linear models
 - Decision trees
 - Rule-based models
 - Generalized additive models
 - Prototype-based models
- Black-box models
 - Neural networks
 - Ensemble models



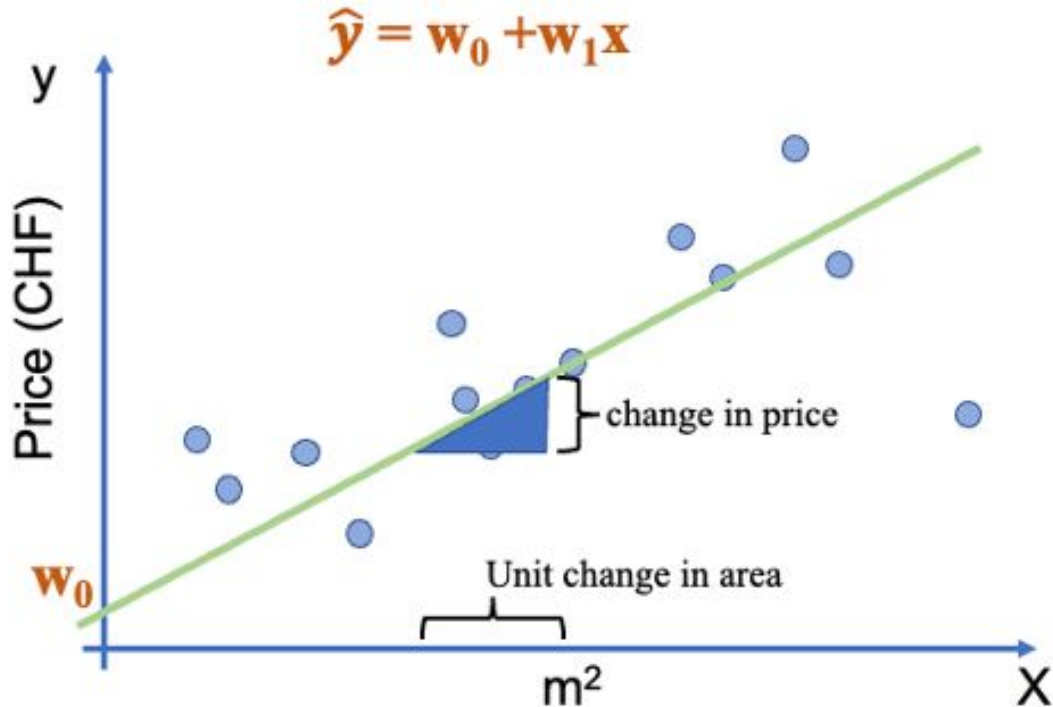
How to achieve model interpretability?

How to achieve model interpretability?

If possible, use an inherently interpretable model

How to achieve model interpretability?

Example: House price prediction based on house size using logistic regression



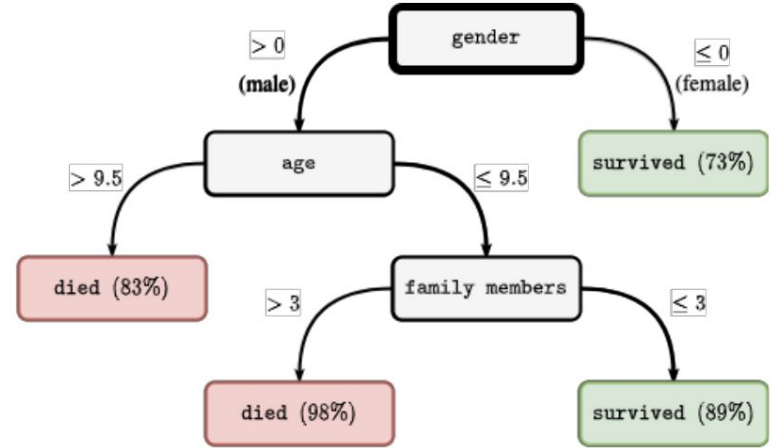
Predicted change in house price for 1 unit change in the house size?

$$\hat{y}(101) - \hat{y}(100)$$

How to achieve model interpretability?

Example: Predicting survival probability of passengers on the Titanic using a decision tree

Explanation: If a passenger was male and under 9.5 years and there were 3 or fewer family members in their family, then their survival probability was 0.89



IF $((\text{gender} > 0) \wedge (\text{age} \leq 9.5) \wedge (\text{family members} \leq 3))$ THEN survived

Generalized Additive Models (GAMs/GA²Ms)

GAMs provide a good balance between explainability and accuracy [[Caruana et al. 2015](#)]

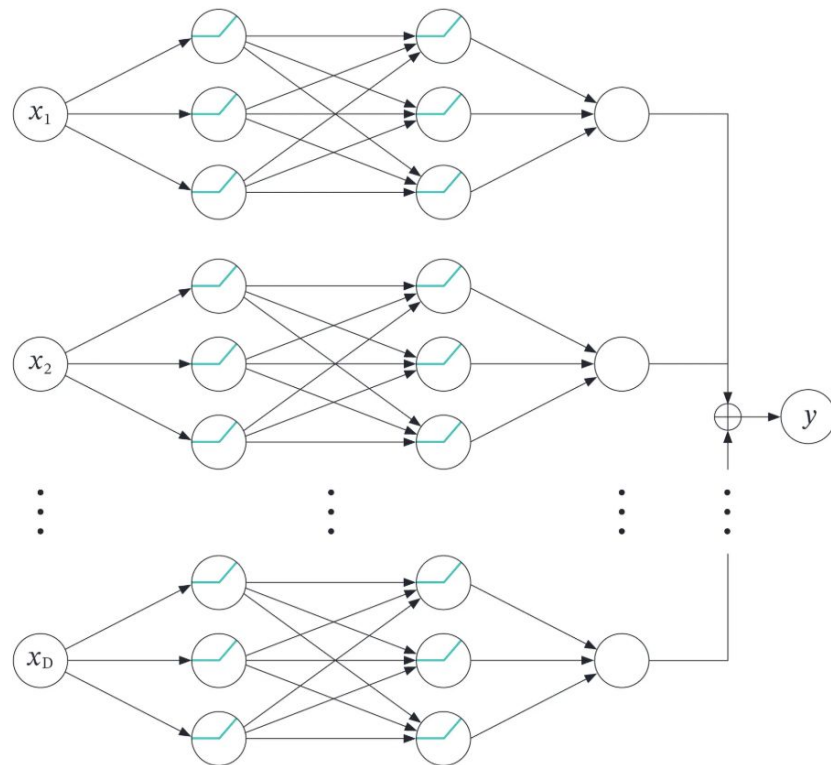
Model	Form	Intelligibility	Accuracy
Linear Model	$y = \beta_0 + \beta_1 x_1 + \dots + \beta_n x_n$	+++	+
Generalized Linear Model	$g(y) = \beta_0 + \beta_1 x_1 + \dots + \beta_n x_n$	+++	+
Additive Model	$y = f_1(x_1) + \dots + f_n(x_n)$	++	++
Generalized Additive Model	$g(y) = f_1(x_1) + \dots + f_n(x_n)$	++	++
Full Complexity Model	$y = f(x_1, \dots, x_n)$	+	+++

- g is a link function: identity function in case of regression;
 $\log(y/1 - y)$ in case of classification.
- f_i is a shape function (splines, trees, neural networks)

Neural Generalized Additive Models

Use neural networks as functions in the GAMs

In [Agarwal et al. 2021](#) each feature is processed separately using a neural network after which the outputs of all networks are combined (added) to produce the final output



Interpretable Decision Sets (IDS)

- [Lakkaraju et al. 2016](#)
- Learns a collection of simple (interpretable), non-overlapping *if-then* rules from the data starting from a set of frequent patterns (obtained by Apriori algorithm) to optimize for:
 - Precision and Recall (model performance)
 - Distinctness, parsimony and class coverage (model interpretability)
 - **Parsimony:** Fewer rules with fewer conditions
 - **Distinctness:** Minimal rules overlap, no redundant or contradicting explanations
 - **Class coverage:** Explain all the classes in the data and not only the majority classes

Interpretable Decision Sets (IDS)

Example disease diagnosis:

If Respiratory-Illness=Yes and Smoker=Yes and Age \geq 50 then Lung Cancer

If Risk-LungCancer=Yes and Blood-Pressure \geq 0.3 then Lung Cancer

If Risk-Depression=Yes and Past-Depression=Yes then Depression

If BMI \geq 0.3 and Insurance=None and Blood-Pressure \geq 0.2 then Depression

If Smoker=Yes and BMI \geq 0.2 and Age \geq 60 then Diabetes

If Risk-Diabetes=Yes and BMI \geq 0.4 and Prob-Infections \geq 0.2 then Diabetes

If Doctor-Visits \geq 0.4 and Childhood-Obesity=Yes then Diabetes

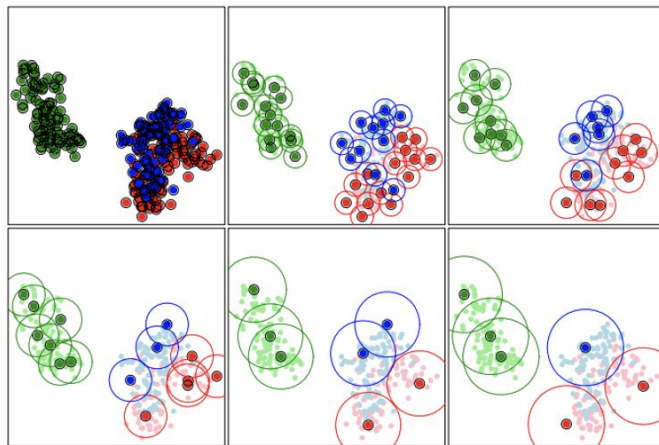
Each rule is a combination of an itemset and a class $r=(s, c)$

Each itemset contains one or more predicates

If a sample satisfies zero itemsets then its class label is assigned using a default label, and if a sample satisfies more than one itemset, it is assigned a class using a tie-breaking function.

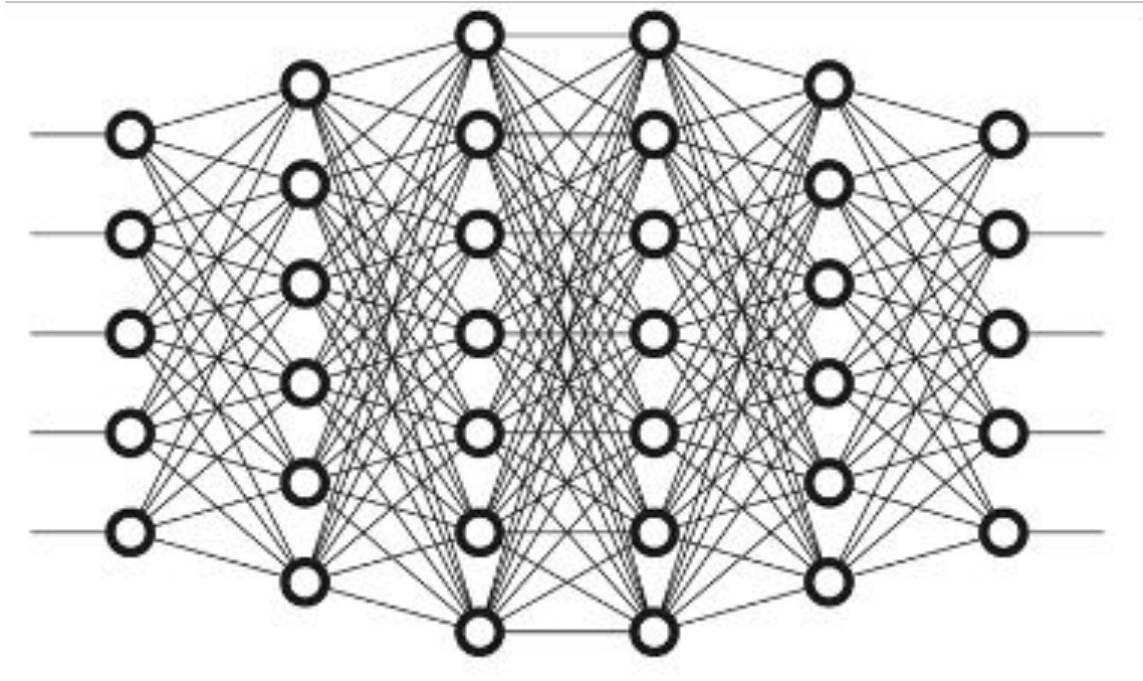
Prototype Selection for Interpretable Classification

- Each instance covers the ϵ -neighborhood around it
- **Goal:** Identify L prototypes (examples) from the available data such that a new (unseen) example will be correctly classified when assigned the same label as its closest prototype



Given a value for ϵ , the choice of $\mathcal{P}_1, \dots, \mathcal{P}_L$ induces L partial covers of the training points by ϵ -balls. Here ϵ is varied from the smallest (top-left panel) to approximately the median interpoint distance (bottom-right panel).

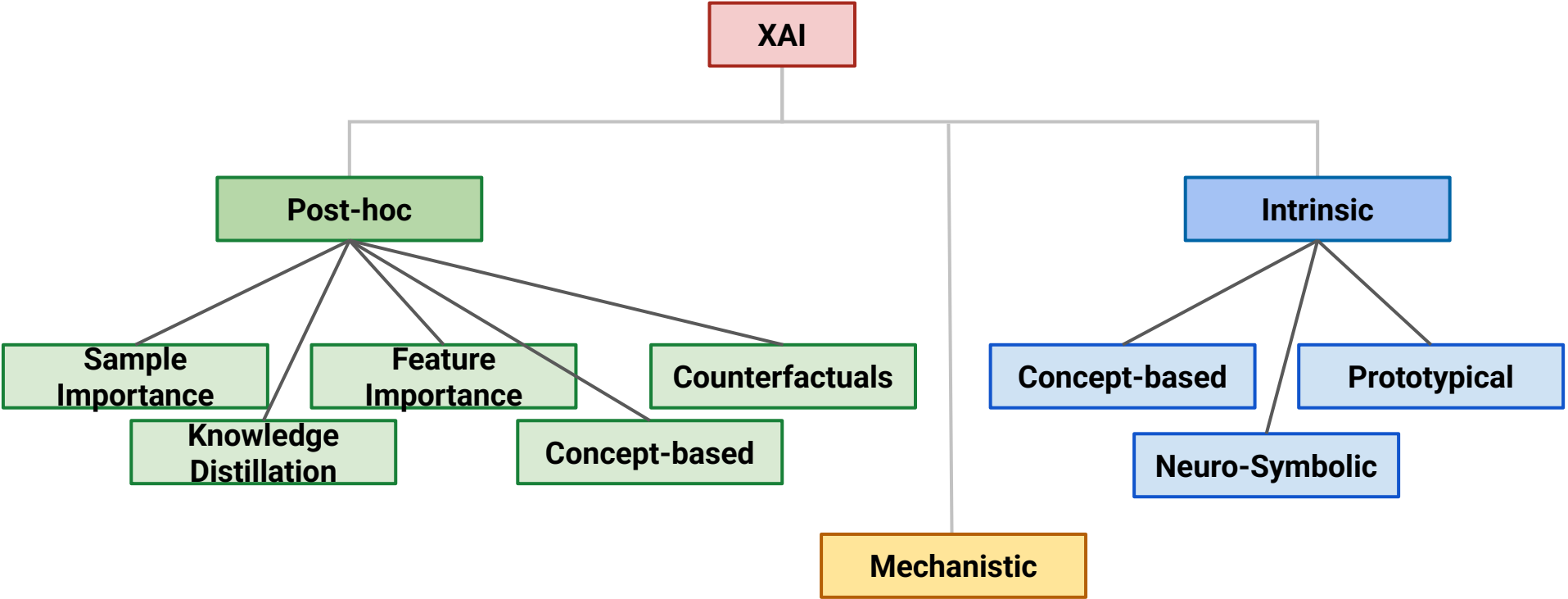
What if we need to use a black box model?



Taxonomy of XAI Methods for Black-Box Models

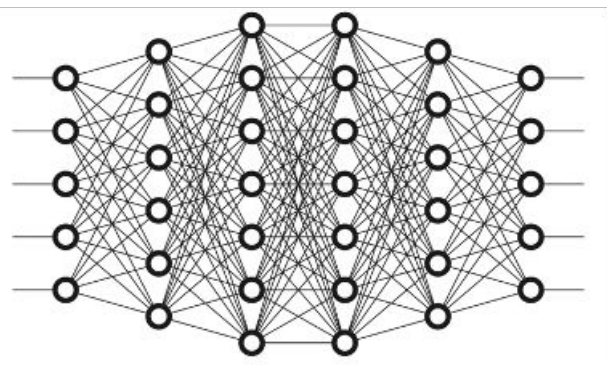
- **Post-hoc**, **intrinsic (in-model)**, and **mechanistic** explanations
- **Local** (explains the prediction for one sample) and **global** (explains the predictions of the whole model - the average behaviour of the model)
- **Model-specific** and **model-agnostic**

Different Explanation Aspects



What is an Explanation?

Interpretable description of the model behavior: faithful to the model and understandable to the user



Faithful
to model

Explanation

Understandable
for user



Different Ways of Explaining

- **Analytic statement:** natural language descriptions of elements and context important for the model output/decision
- **Visualisations:** highlight parts of data important for the decision/prediction
- **Examples:** give illustrative/typical examples that support the prediction
- **Counterfactuals:** provide the changes needed to get different decisions

How to achieve model interpretability?

- **Post-hoc interpretation of pre-trained complex black-box models**
- **Explanation algorithm:** provides an interpretable description of the model predictions/behavior



Post-Hoc Explainability Approaches

Local that explain individual predictions (instance-level approaches)

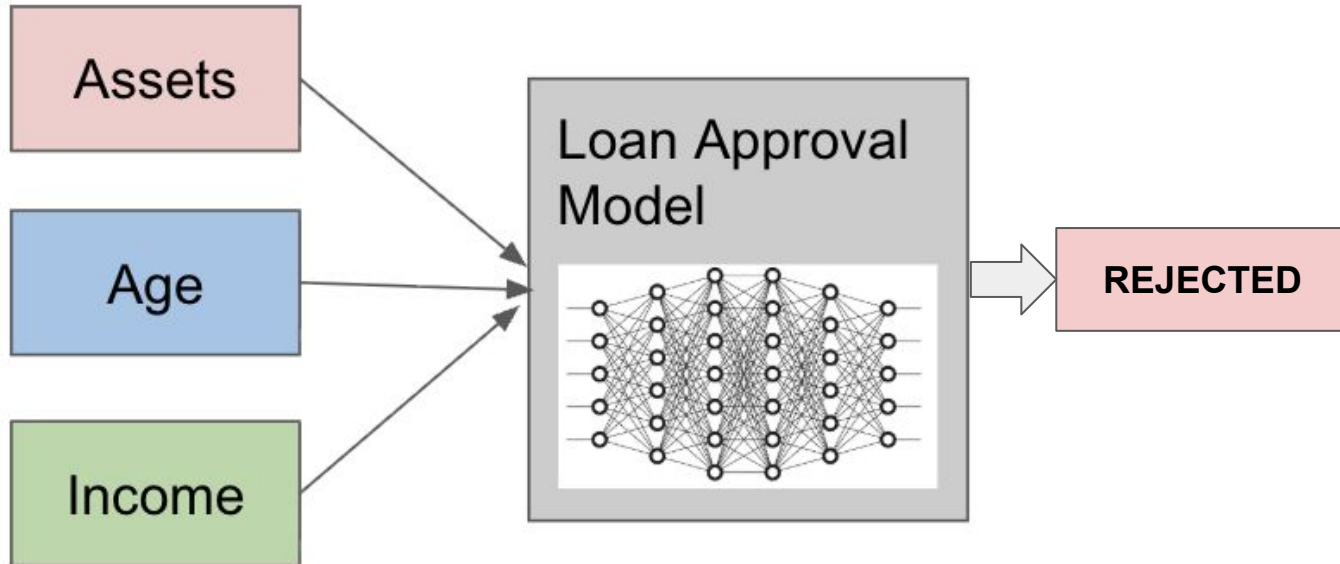
- Feature importances (feature attribution) for individual predictions
- Examples important for the specific decision
- Local decision rules
- Counterfactuals

Global that explain the global behavior of the model (model-level approaches)

- Global feature importances
- Global decision rules
- Model (knowledge) distillation / Mimicking
- Summaries of counterfactuals

Feature Attribution / Feature Importance

- Very common for us humans to explain objects based on their features
- **Goal:** Which features were most important for the prediction of the model



Local Post-Hoc Feature Attribution Methods

Locality

Main assumption: Globally complex models are probably way less complex locally

Locality refers to the vicinity of a particular sample whose prediction we aim to explain

LIME

Local Interpretable Model-Agnostic Explanations (LIME) [[Ribeiro 2016](#)]

Feature importances for individual predictions

Goal: Identify an interpretable model over an **interpretable representation** that is **locally faithful** to the classifier

Explains model decisions in a small, local region around a given target example

Interpretable representation is one that is understandable by humans

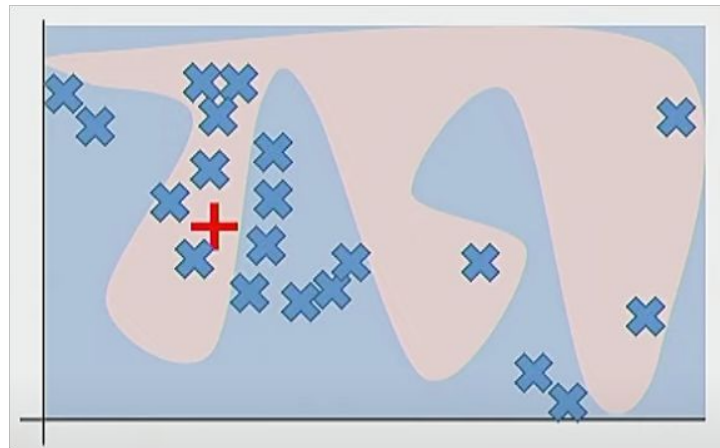
- **Text:** binary representation indicating the presence or absence of words
- **Image:** binary vector indicating the presence or absence of a contiguous patch of similar pixels

LIME

Goal: Identify an interpretable model over the interpretable representation that is locally faithful to the classifier

Algorithm:

1. Sample points close to the target point x_i

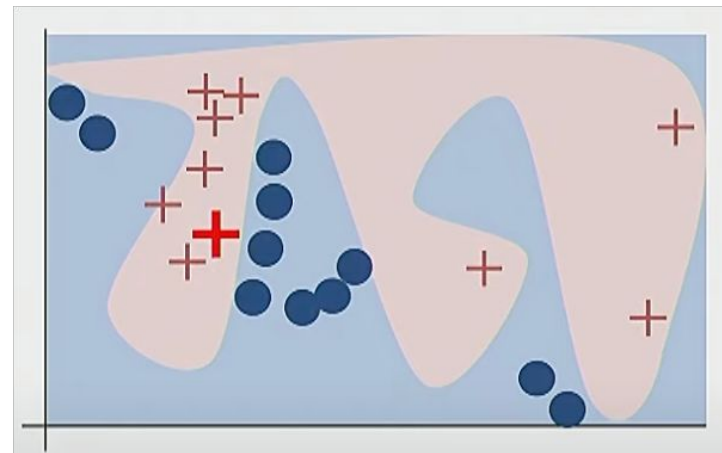


LIME

Goal: Identify an interpretable model over the interpretable representation that is locally faithful to the classifier

Algorithm:

1. Sample points close to the target point x_i
2. Predict the labels for all the sampled points using the available trained model

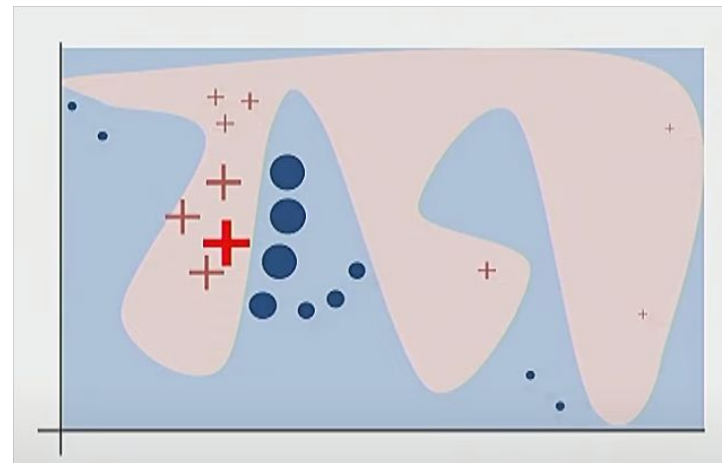


LIME

Goal: Identify an interpretable model over the interpretable representation that is locally faithful to the classifier

Algorithm:

1. Sample points close to the target point x_i
2. Predict the labels for all the sampled points using the available trained model
3. Weigh the newly generated samples based on their distance from target sample x_i

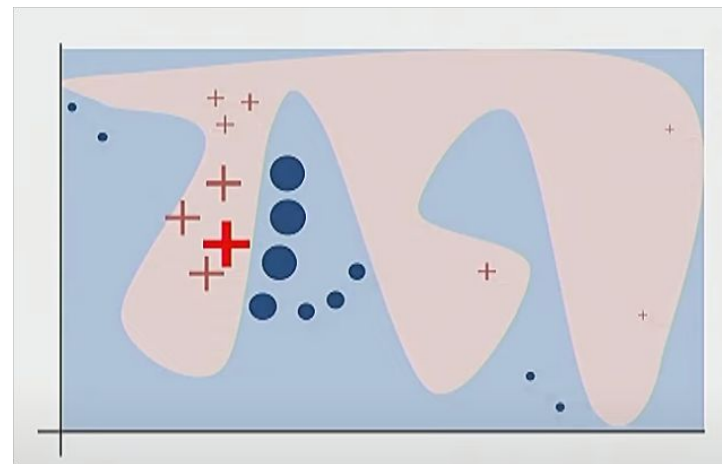
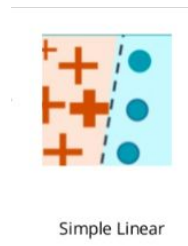


LIME

Goal: Identify an interpretable model over the interpretable representation that is locally faithful to the classifier

Algorithm:

1. Sample points close to the target point x_i
2. Predict the labels for all the sampled points using the available trained model
3. Weigh the newly generated samples based on their distance from target sample x_i
4. Fit a linear model using the weighted samples

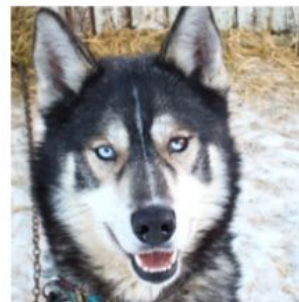
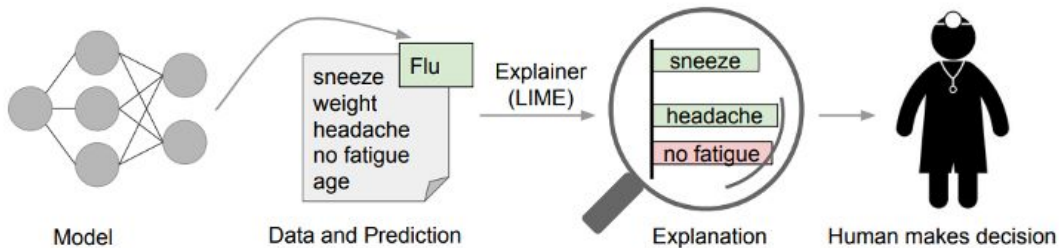


LIME

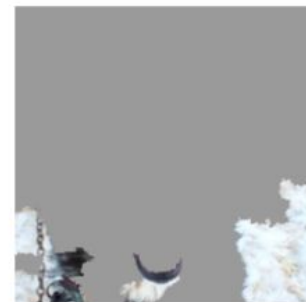
Local Interpretable Model-Agnostic Explanations (LIME) [[Ribeiro 2016](#)]

Algorithm:

1. Sample points close to the target point x_i
2. Predict the labels for all the sampled points using the available trained model
3. Weigh the newly generated samples based on their distance from the target sample x_i
4. Fit a linear model using the weighted samples



(a) Husky classified as wolf



(b) Explanation

SHAP

(SHapley Additive exPlanations)

SHAP: Feature Importance

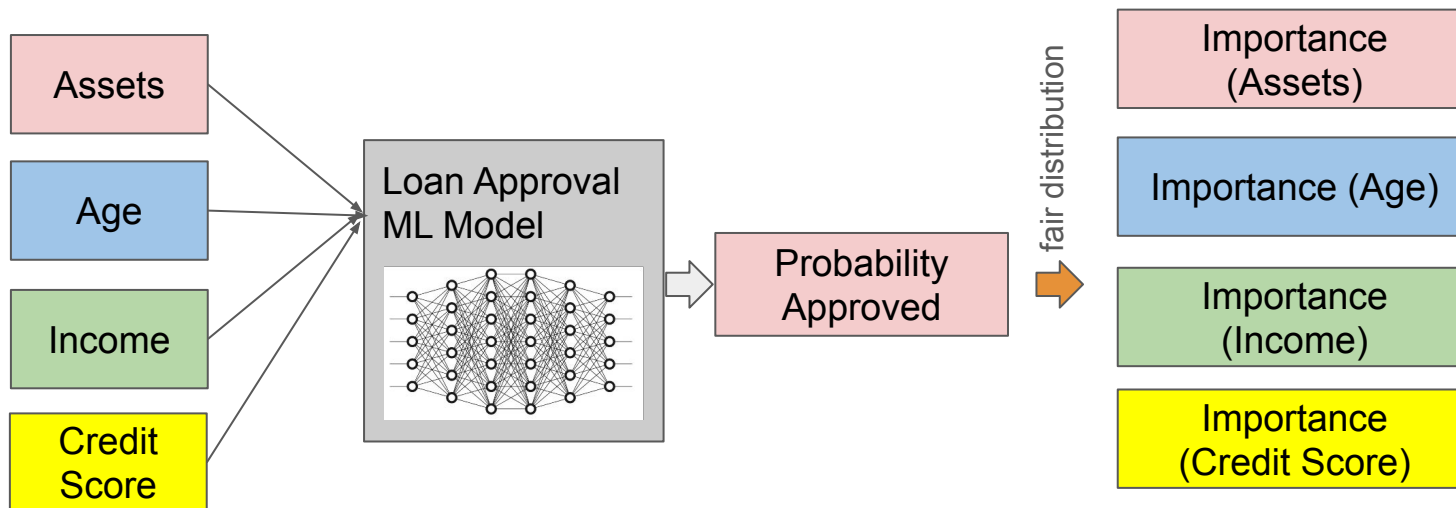
- **SHAP** is a method for explaining machine learning model predictions using ideas from collaborative game theory
- Model's prediction is the reward in a collaborative game where each feature is one player
- **Goal:** Quantify how much each feature contributed to a model's prediction for a specific target data sample while providing fair distribution and accounting for feature interactions

SHAP: Feature Importance

- **SHAP** is a method for explaining machine learning model predictions using ideas from collaborative game theory
- Model's prediction is the reward in a collaborative game where each feature is one player
- **Goal:** Quantify how much each feature contributed to a model's prediction for a specific target data sample while providing fair distribution and accounting for feature interactions
- **Fair distribution:**
 - The rewards for all features should add up to the models output (excluding the baseline)
 - Rewards can be negative
 - Interchangeable players should get equal rewards
 - Features that don't affect the prediction have 0 contribution

SHAP: Feature Importance

- Fair reward distribution accounting for feature interactions:
 - The average contribution of each feature is computed across all possible combination of features



SHAP: Feature Importance

- Fair reward distribution accounting for feature interactions:
 - Importance of a target feature is computed by marginalising over its contributions across all possible subsets of other features

$$\phi_i = \sum_{S \subseteq F \setminus \{i\}} \frac{|S|! (|F| - |S| - 1)!}{|F|!} [f_{S \cup \{i\}}(x_{S \cup \{i\}}) - f_S(x_S)]$$

F : the set of all features

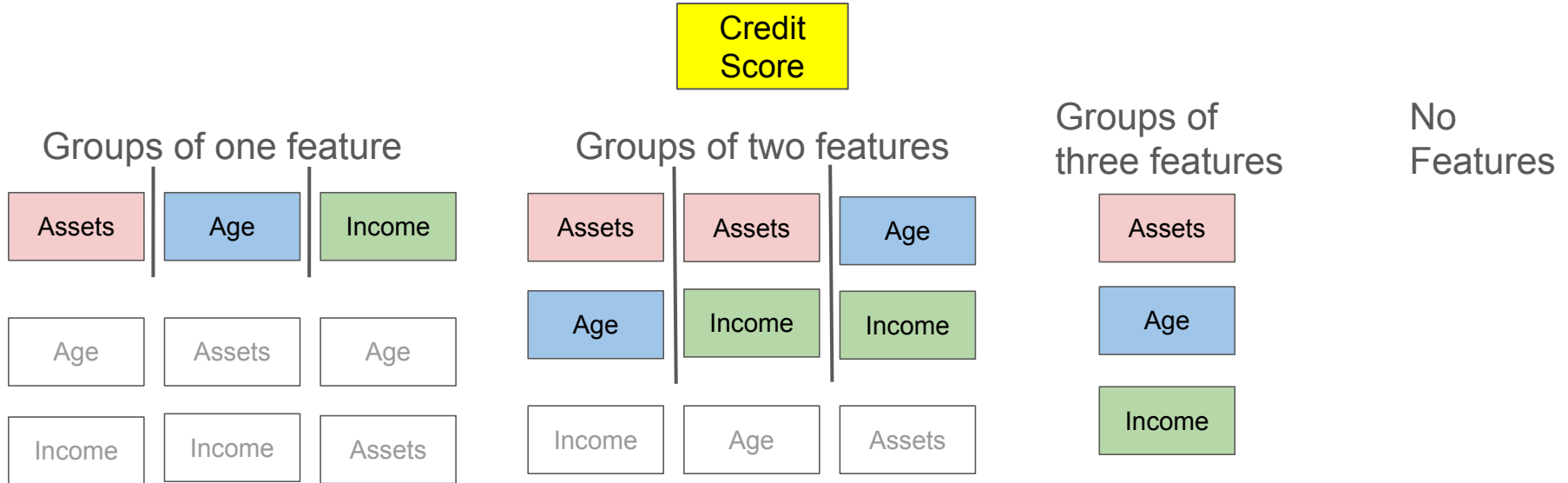
$f_{S \cup \{i\}}$: the model trained with i present

f_S : the model trained with i withheld

x_S : the values of input features in subset S

SHAP: Feature Importance

- Fair reward distribution accounting for feature interactions:
 - Importance of a target feature is computed by marginalising over its contributions across all possible subsets of other features
 - Each feature's contribution depends on how much it improves the prediction on average, given all possible contexts

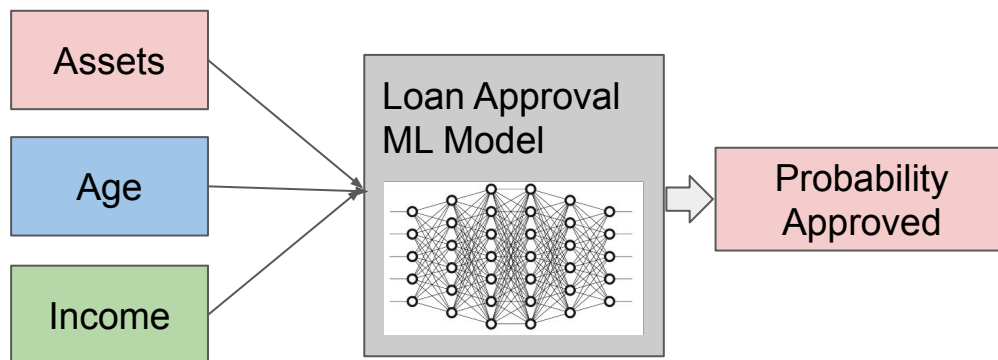


SHAP: Feature Importance

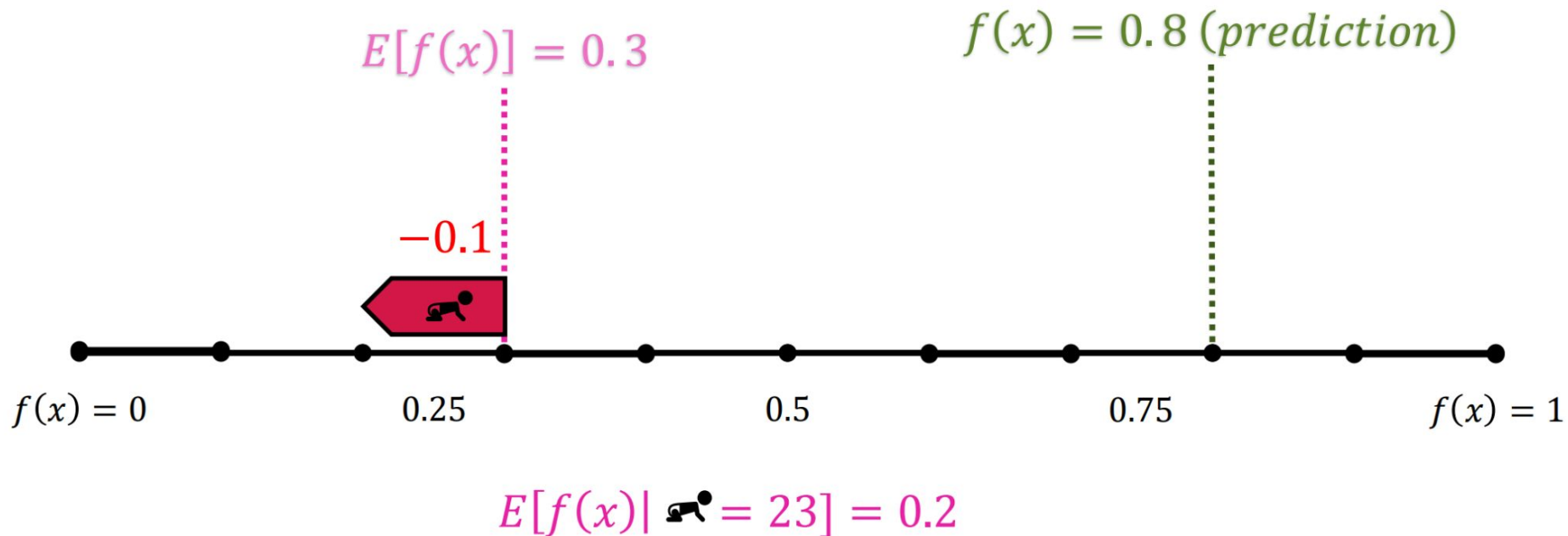
- Fair reward distribution accounting for feature interactions:
 - Feature importance is computed by marginalising over its contributions across all possible subsets of features
 - Too expensive to look at every possible subset of features
- SHAP efficiently estimates Shapley scores by computing how a model's output deviates from its mean (baseline, the prediction for an average data point, expected output with no information on the input features) one feature at a time [[Lundberg et al. 2017](#)] - ML model trained only once
- The computation of Shapley values depends on the model:
 - Exact: LinearSHAP, TreeSHAP
 - Approximation: DeepSHAP, KernelSHAP

SHAP: Feature Importance

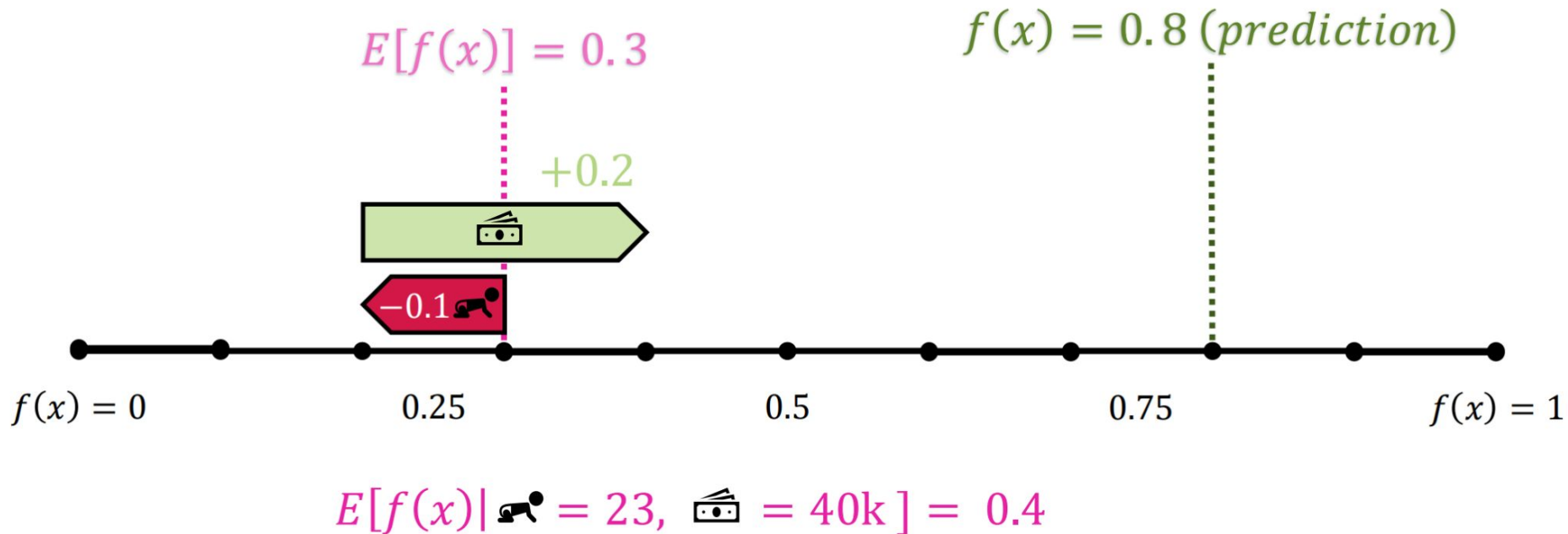
Example



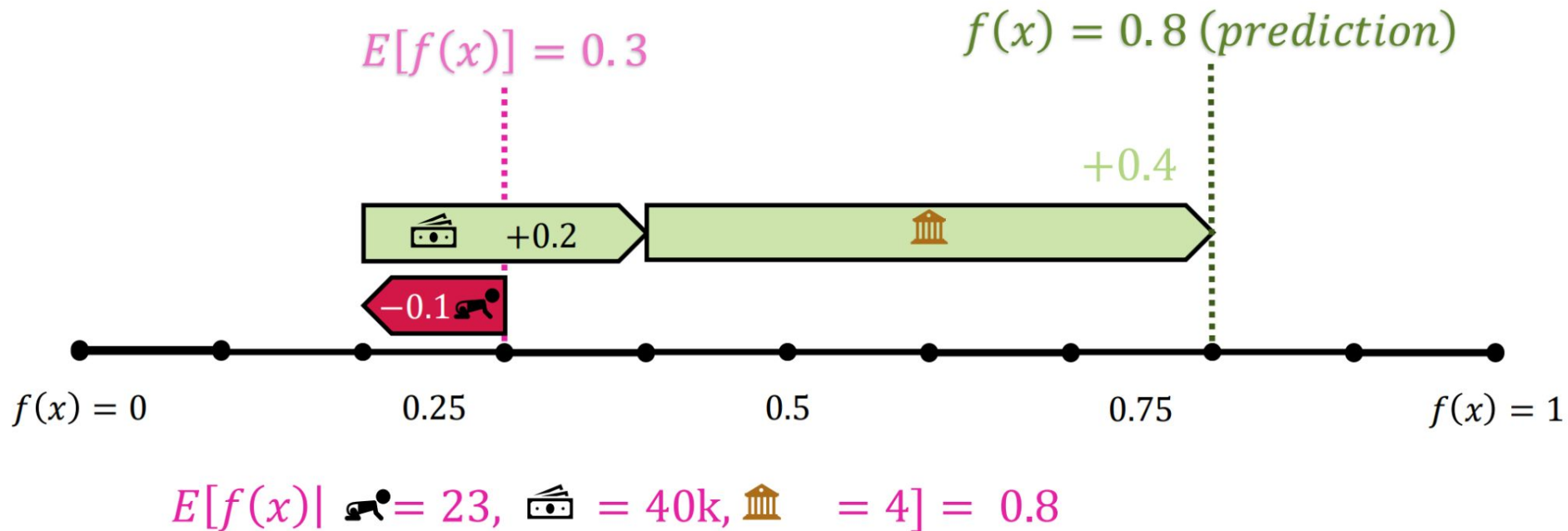
SHAP: Feature Importance



SHAP: Feature Importance

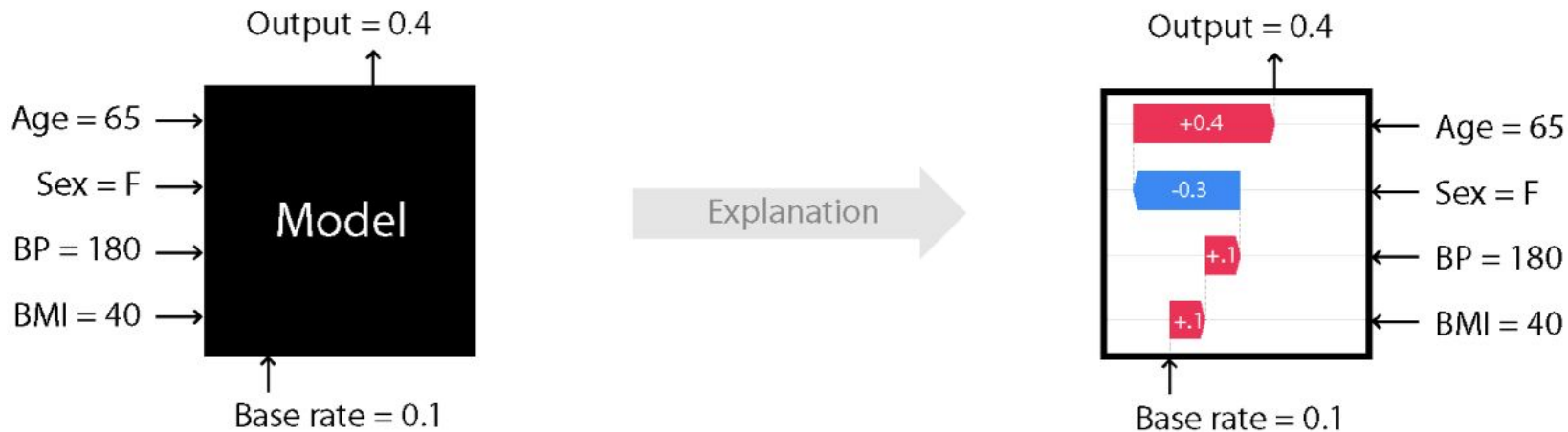


SHAP: Feature Importance



SHAP Library

There is an open source library: <https://shap.readthedocs.io/en/latest>

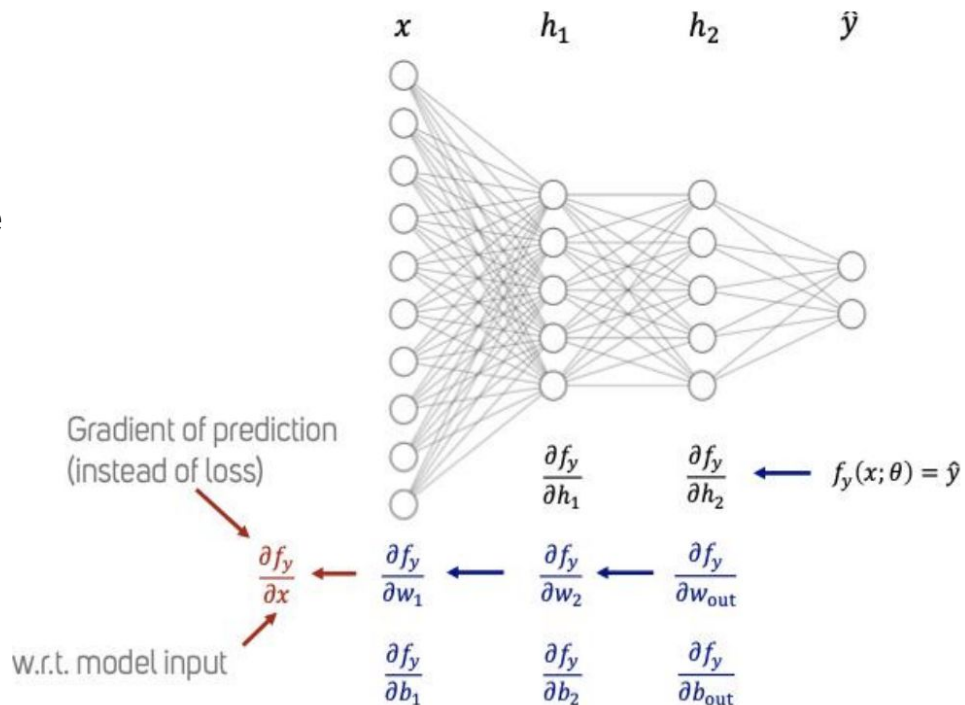


Gradients

Gradients

The partial derivatives (gradients) of the loss for all parameters of a neural network are computed via backpropagation and quantify how much each model parameter needs to be changed to minimize the loss

In **gradient-based explanation** backpropagation is used to measure feature importance: *gradients of the predictions are calculated with respect to inputs*



Saliency Maps (Sensitivity Maps, Pixel Attribution Maps)

Image classification task:

- Given an input image a neural network model F typically outputs class probability scores S_c for each class with the final prediction assigned to the maximum probability class

$$class(x) = \operatorname{argmax}_{c \in C} S_c(x)$$

- Saliency map** quantifies how much a small change in a pixel x will change the classification score for class c and is given by

$$M_c(x) = \partial S_c(x) / \partial x$$

Which parts of the input are most relevant for the model's prediction of a target class?

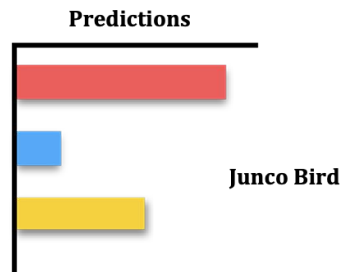
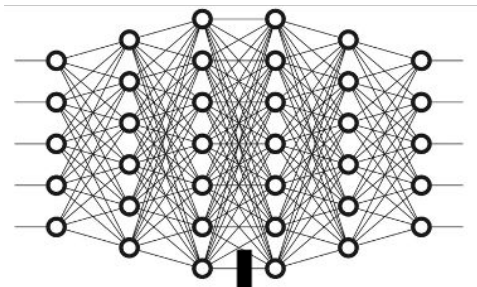
- Pixel attribution maps provide visual interpretation of the gradient of the class activation function with respect to a target input sample (image)
- Structured as a grayscale (or single color) image with same dimension as the input image where brightness or color intensity of each pixel indicates its importance for the target class

Saliency Maps

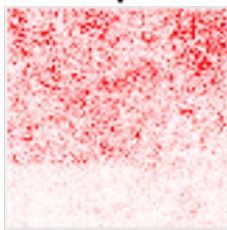
Which parts of the input are most relevant for the model's prediction of a target class?

- Pixel attribution maps provide visual interpretation of the gradient of the class activation function with respect to a target input sample (image)
- **Saliency map** quantifies how much a small change in a pixel x will change the classification score for class c

$$M_c(x) = \partial S_c(x) / \partial x$$



$M_c(x)$



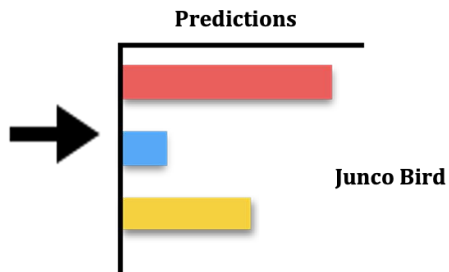
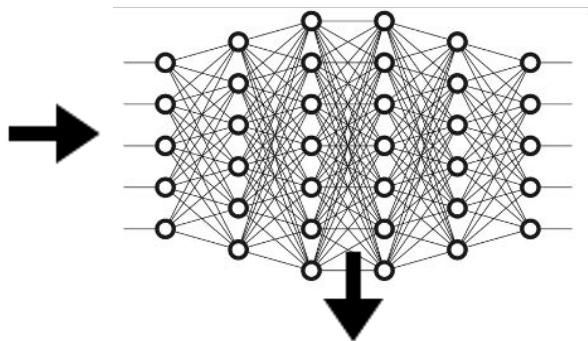
Saliency maps are very noisy due to large local variations in partial derivatives and thus difficult to interpret for a human

Saliency Maps: SmoothGrad

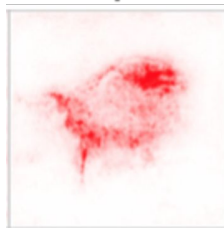
SmoothGrad [[Smilkov et al. 2017](#)]:

- Sample similar images to the image of interest by adding Gaussian noise to it
- Take the average of the saliency maps of all generated noisy images (average input gradients of noisy inputs)

$$\hat{M}_c(x) = \frac{1}{n} \sum_1^n M_c(x + \mathcal{N}(0, \sigma^2))$$



$\hat{M}_c(x)$



Saliency Maps: SmoothGrad

SmoothGrad effect of noise

- Averages over 50 samples for different noise levels
- Noise level between 10% and 20% gives good sharpness of the sensitivity map while maintaining the structure of the original image

Noise level:

0%

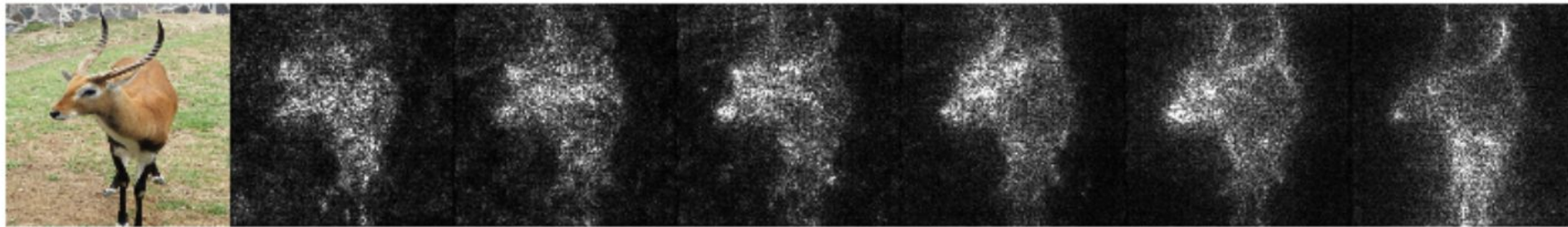
5%

10%

20%

30%

50%



Noise level: $\sigma / (x_{max} - x_{min})$

Saliency Maps: SmoothGrad

SmoothGrad effect of the sample size

- 10% noise applied to each image
- Estimated gradient becomes smoother as the sample size increases
- Little apparent changes for sample sizes larger than 50

Sample size n :

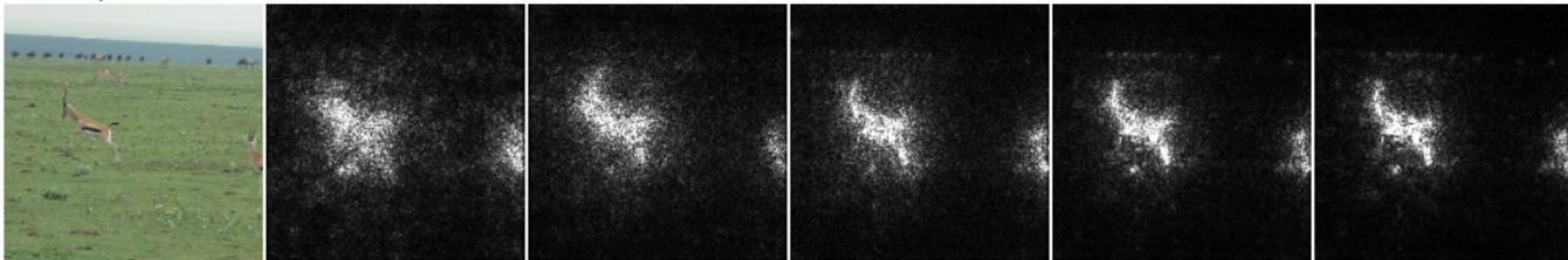
2

5

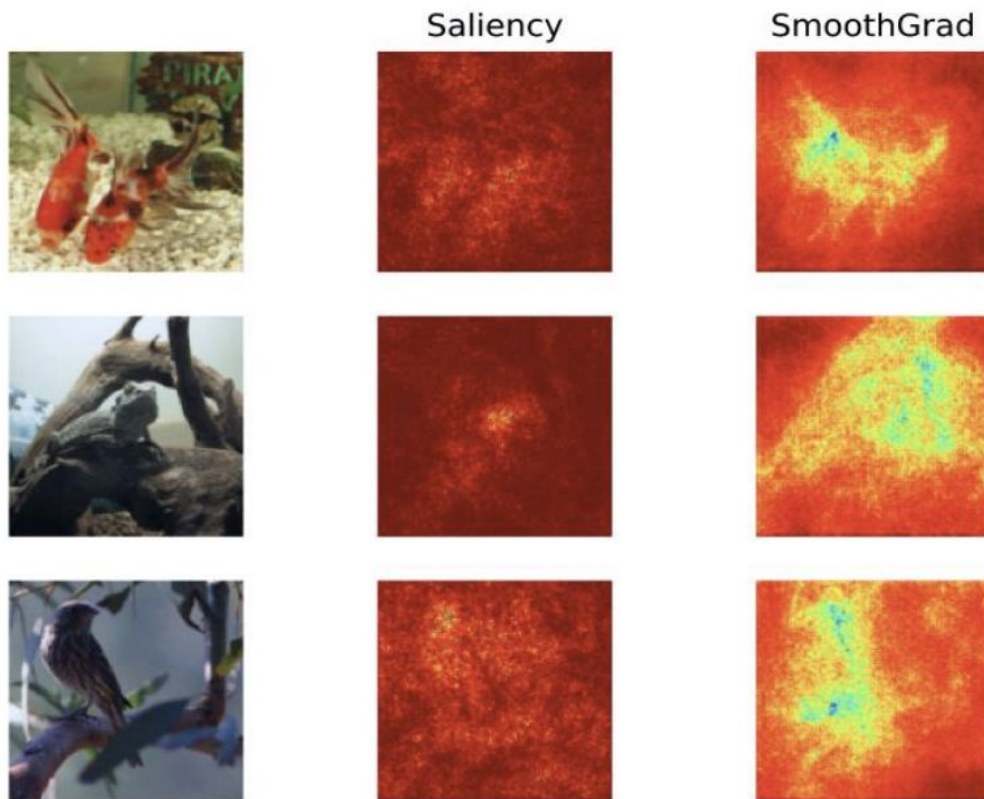
20

50

100



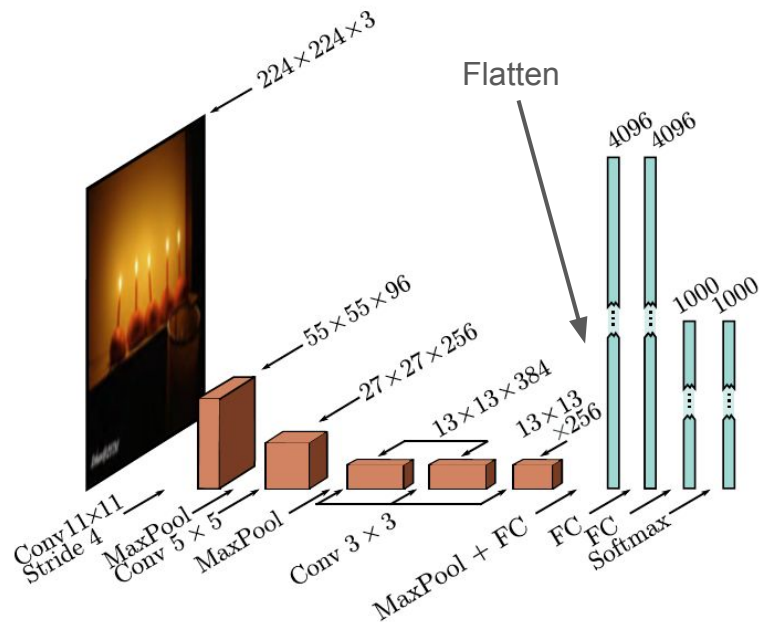
Saliency Maps: SmoothGrad



Convolutional Neural Networks (CNNs)

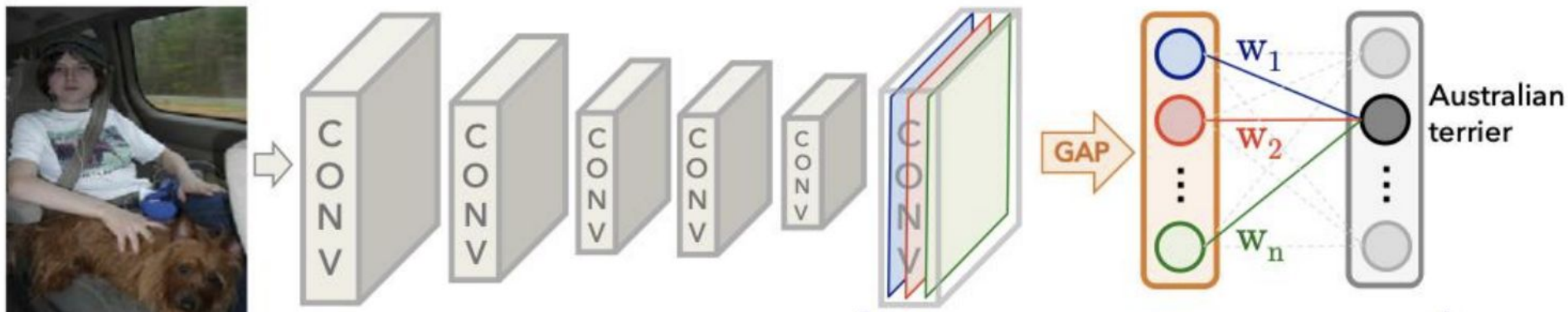
CNNs are widely used in image classification because of their high accuracy and computational efficiency

In a typical CNN architecture the output feature maps from the last convolutional layer are typically flattened and then passed through one or more fully connected layers before the output layer



CNNs with Global Average Pooling (GAP)

In CNN architectures with Global Average Pooling the feature maps are summarized for each channel using GAP and the resulting summary vector is passed to the final classification output layer



$$GAP^{(k)} = \frac{1}{H \times W} \sum_{i=1}^H \sum_{j=1}^W A_{ij}^{(k)}$$

Activation value at pixel (i, j) for the k^{th} feature map

Height and width of feature map

$$y^c = \sum_{k=1}^K w_k^c \cdot GAP^{(k)} + b^c$$

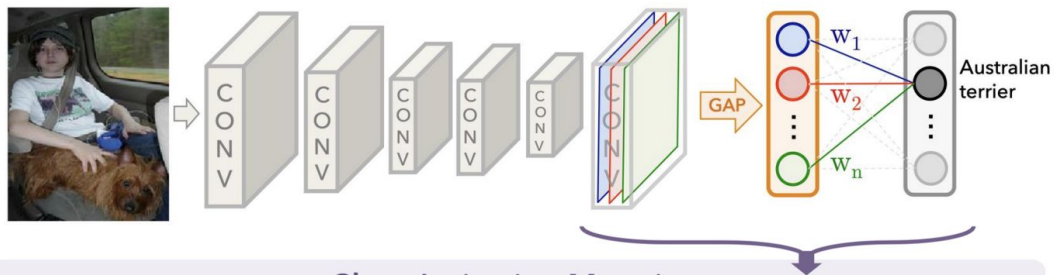
linear weight associated with feature map k for class c

CNNs with Global Average Pooling (GAP)

GAP Advantages:

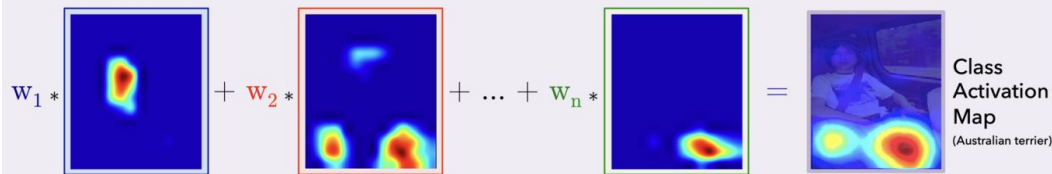
- Parameter-free and reduces dimensionality
- Sums out spatial information (spatial invariance)
- **Interpretability:** the weight of the linear layer provides the importance of the channels for a class

Class Activation Mapping (CAM) [Zhou et al. 2015]: Normalize the feature maps, convert them to a heatmap and overlay them on the original image



$$y^c = \sum_{k=1}^K w_k^c \cdot \text{GAP}^{(k)} + b^c$$

Class Activation Mapping



CAM heatmap for class c : $L_{CAM}^c = \sum_{k=1}^K w_k^c A^{(k)}$

Grad-CAM

Grad-CAM [[Selvaraju et al. 2017](#)] is a generalization of CAM

- Combines activation maps and gradient-based feature importance
- Unlike other gradient-based methods the gradient is only propagated to the last convolutional layer of the CNN and not all the way back to the input layer

- Step 1: compute gradient of class score w.r.t. feature maps

$$\alpha_k^c = \frac{1}{H \times W} \sum_{i=1}^H \sum_{j=1}^W \frac{\partial y^c}{\partial A_{ij}^{(k)}} \quad \text{Gradient of class score w.r.t. pixel } (i,j) \text{ in feature map } A^k$$

- Step 2: Grad-CAM heatmap for class c

$$L_{Grad-CAM}^c = \text{ReLU} \left(\sum_{k=1}^K \alpha_k^c A^{(k)} \right)$$

Grad-CAM

Grad-CAM [[Selvaraju et al. 2017](#)] is a generalization of CAM

- Combines activation maps and gradient-based feature importance
- Unlike other gradient-based methods the gradient is only propagated to the last convolutional layer of the CNN and not all the way back to the input layer

- Step 1: compute gradient of class score w.r.t. feature maps

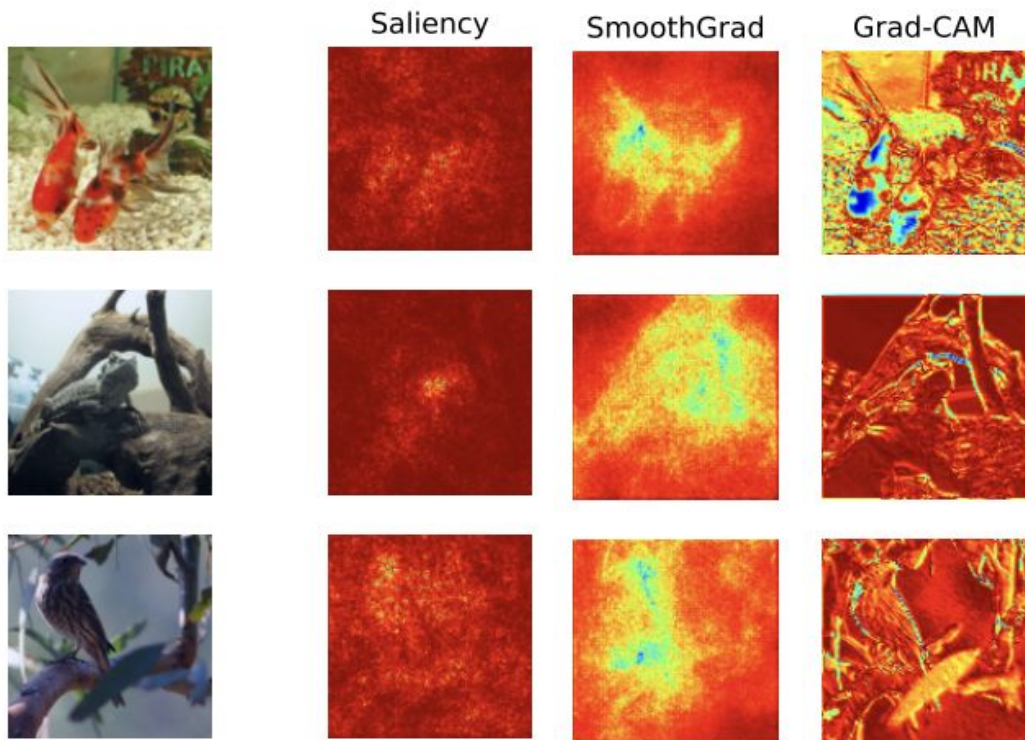
$$\alpha_k^c = \frac{1}{H \times W} \sum_{i=1}^H \sum_{j=1}^W \frac{\partial y^c}{\partial A_{ij}^{(k)}} \quad \text{Gradient of class score w.r.t. pixel } (i,j) \text{ in feature map } A^k$$

- Step 2: Grad-CAM heatmap for class c

$$L_{Grad-CAM}^c = \text{ReLU} \left(\sum_{k=1}^K \alpha_k^c A^{(k)} \right)$$

ReLU is applied to the linear combination of maps because only features that have a positive influence on the class of interest are kept

Gradient-Based Methods Comparison



Source: <https://arxiv.org/pdf/2107.11400>

Gradient-Based Methods

Many other approaches:

- Integrated gradients, layerwise relevance propagation, DeepLIFT, guided backpropagation

Sample-Based Post-hoc Approaches

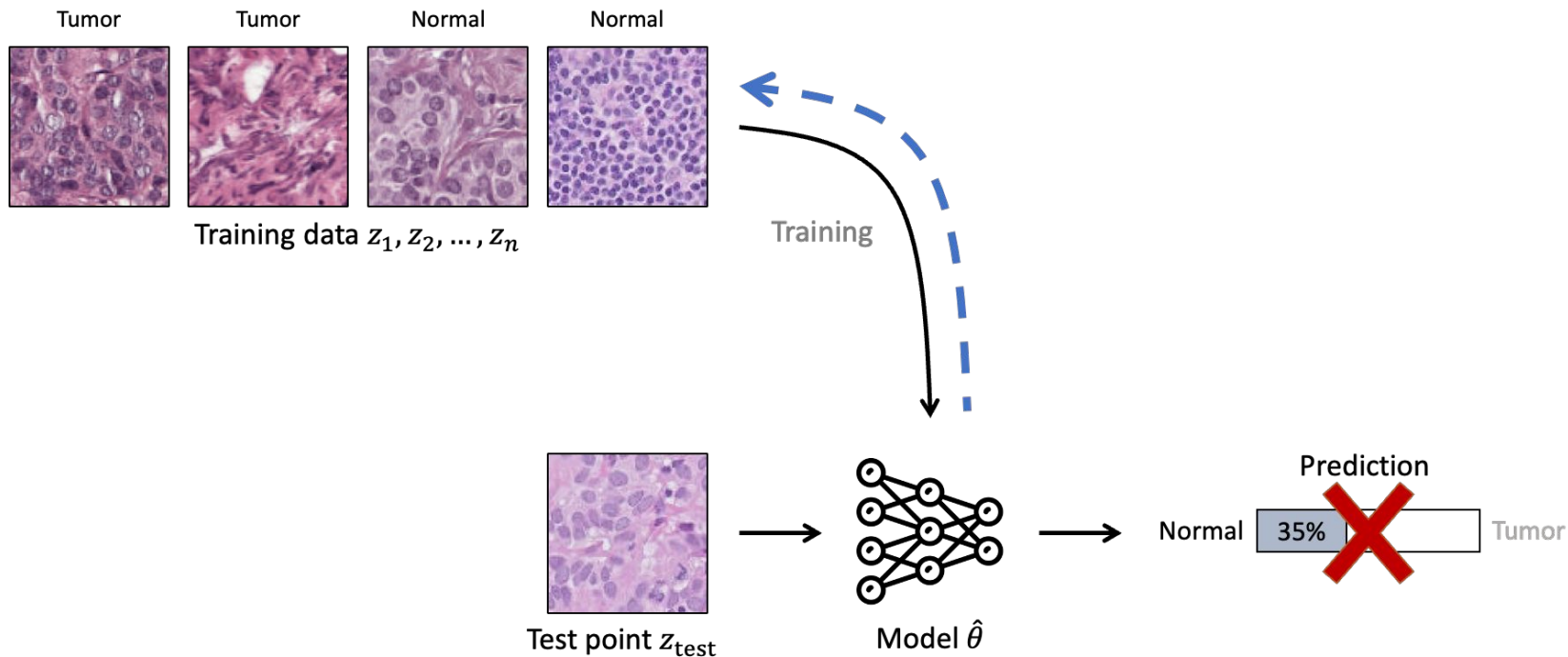
Sample Based Post-Hoc Approaches

Explain a model via input examples:

- Which samples were the most important for a given prediction of a target sample?

Sample Based Post-Hoc Approaches

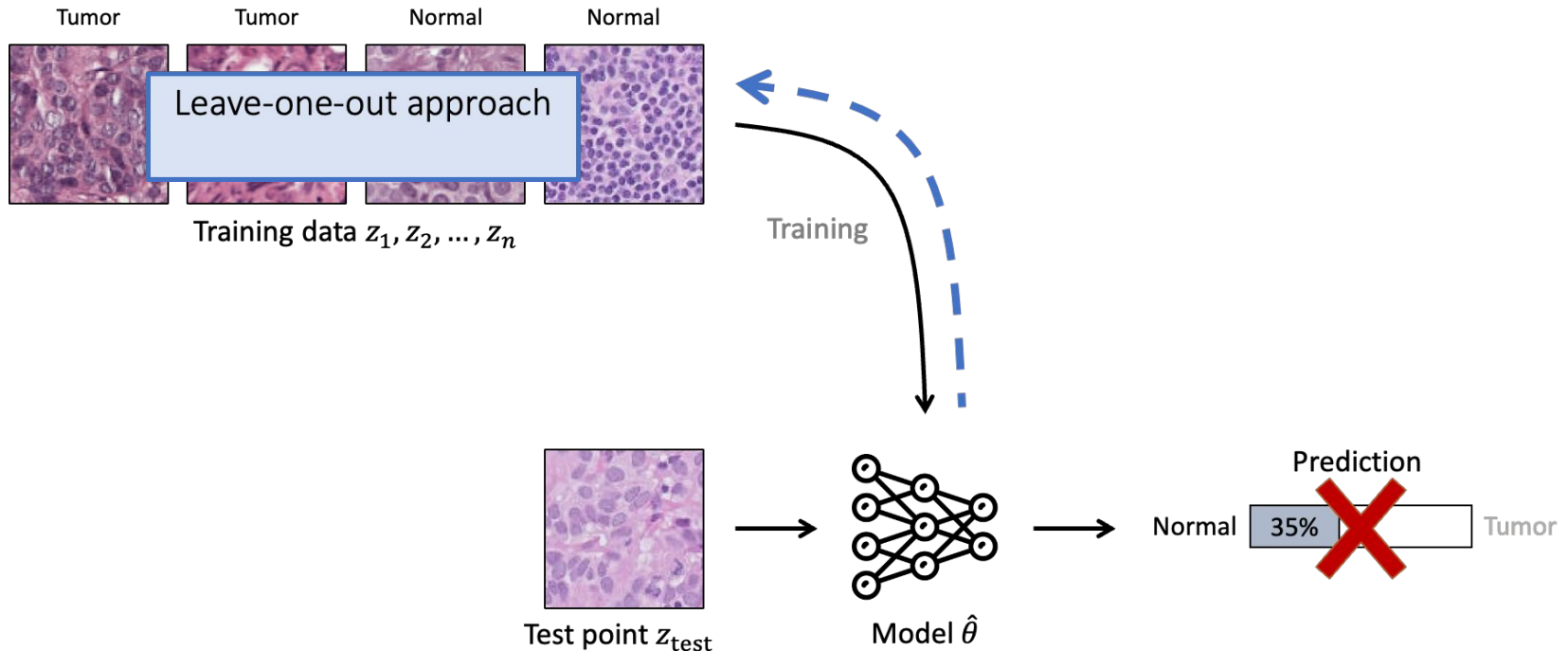
How can we quantify the importance of a training point for model's predictions?



Sample Based Post-Hoc Approaches

How can we quantify the importance of a training point for model's predictions?

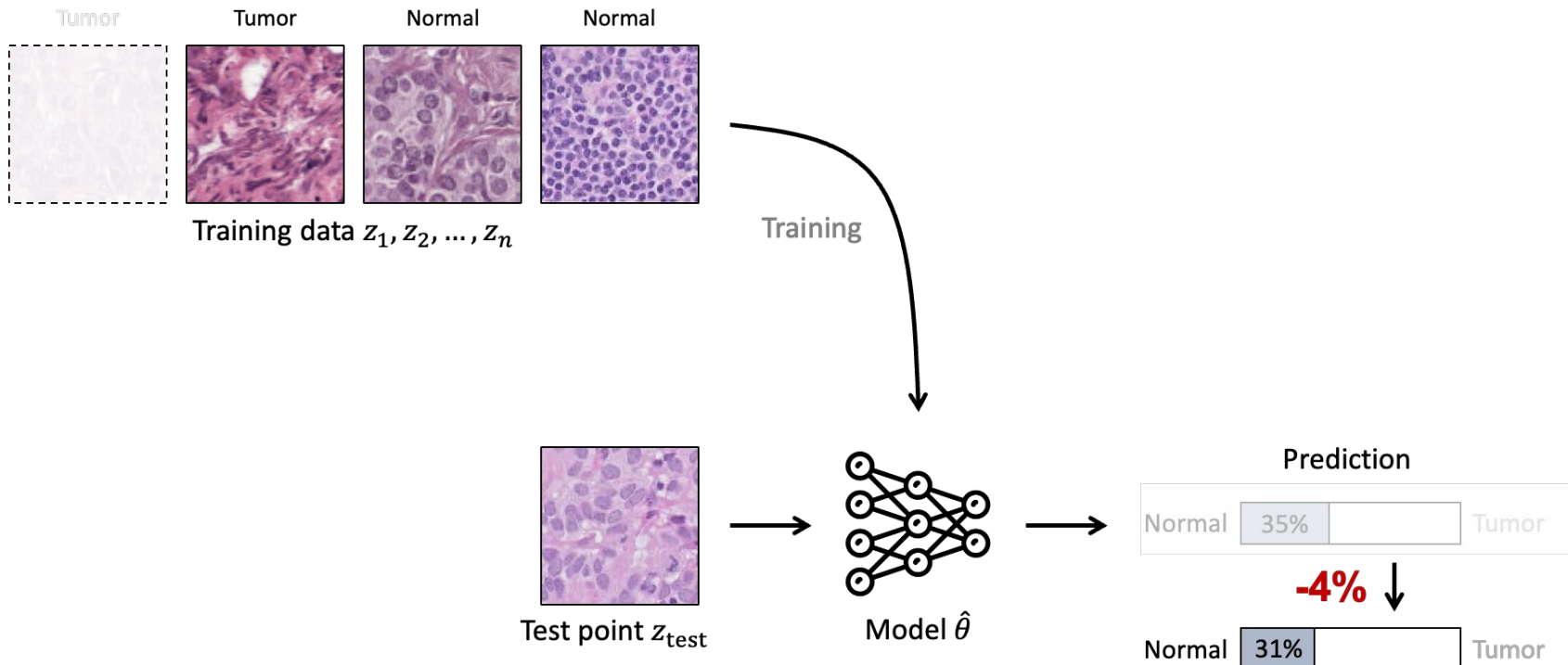
- Retrain the model with each training point removed



Sample Based Post-Hoc Approaches

How can we quantify the importance of a training point for model's predictions?

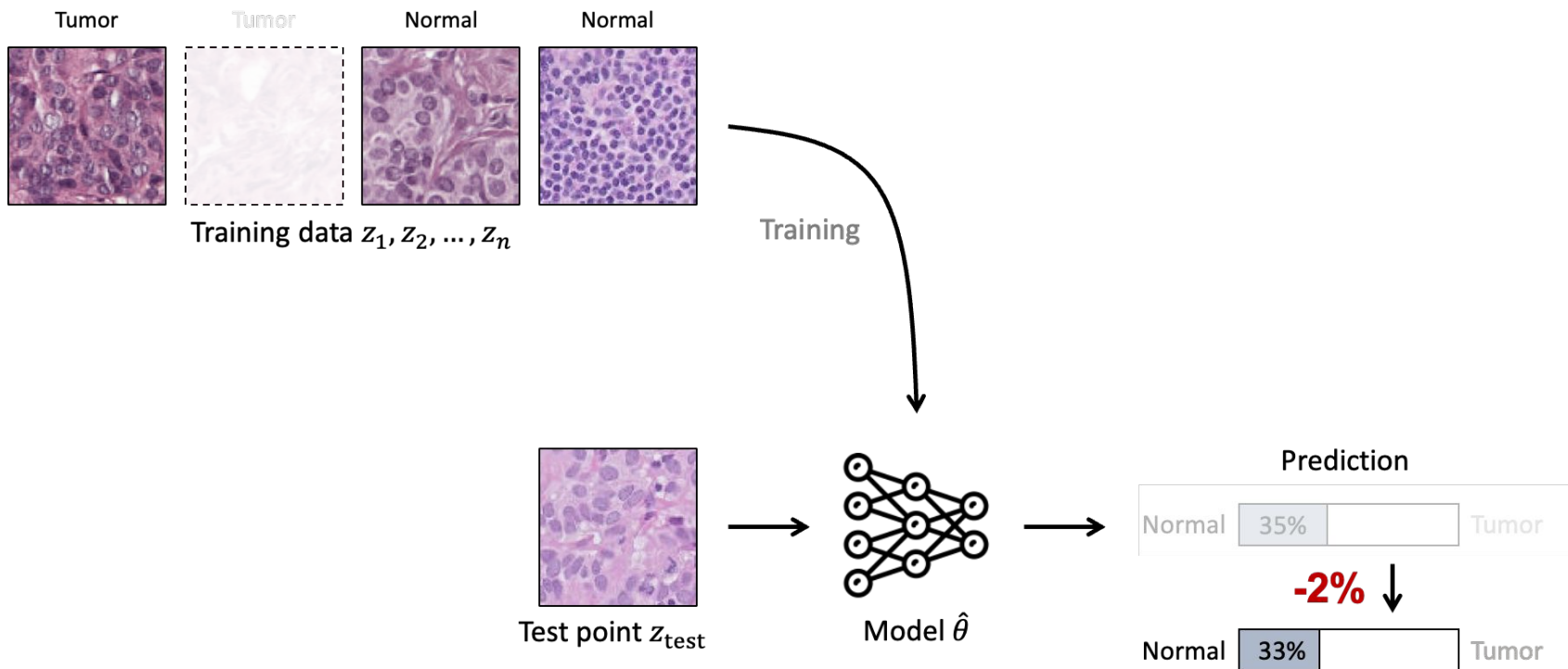
- Retrain the model with each training point removed



Sample Based Post-Hoc Approaches

How can we quantify the importance of a training point for model's predictions?

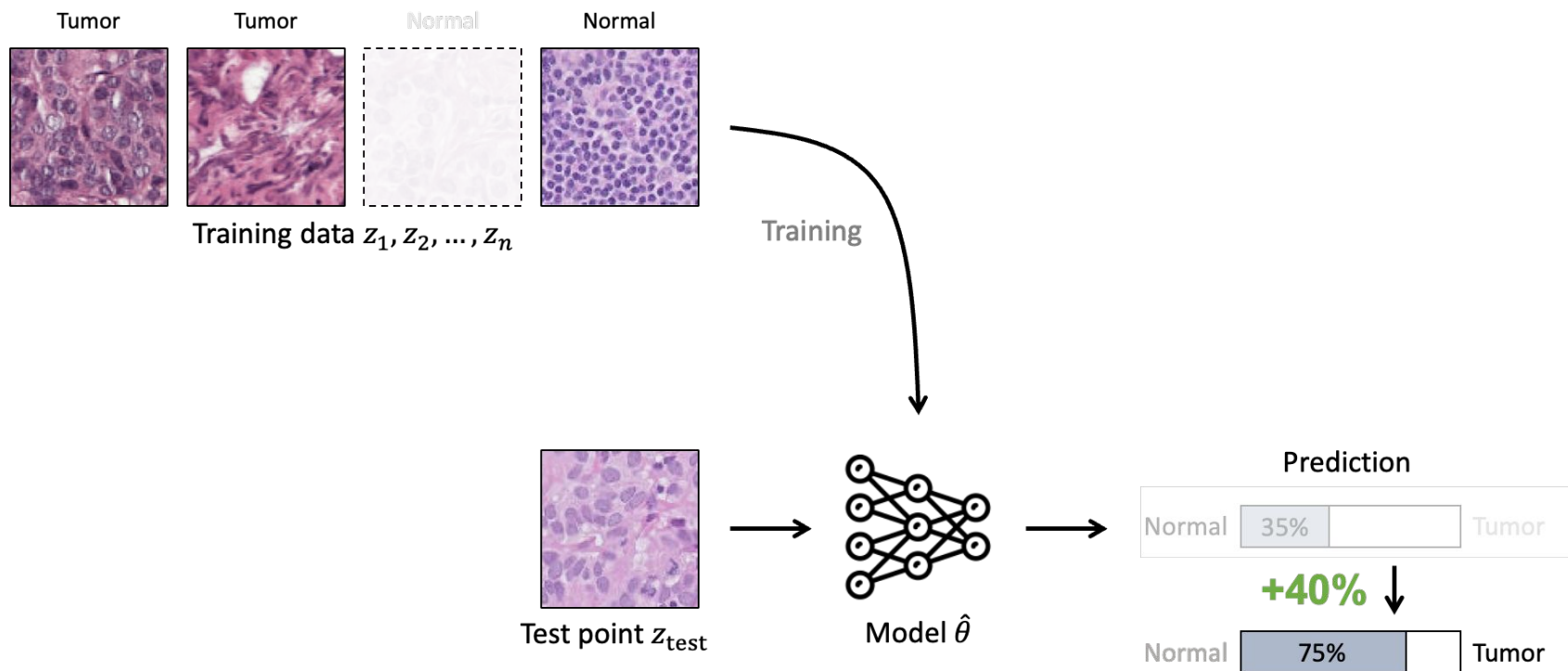
- Retrain the model with each training point removed



Sample Based Post-Hoc Approaches

How can we quantify the importance of a training point for model's predictions?

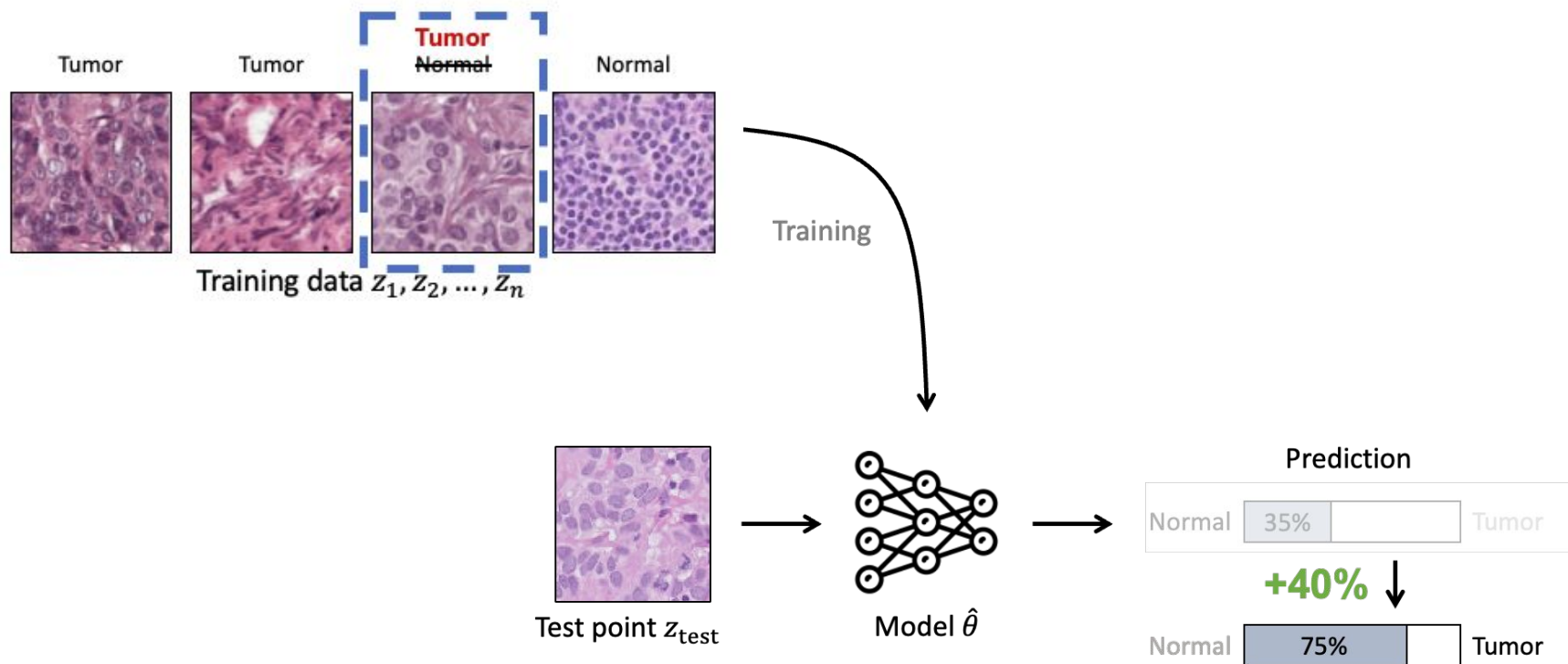
- Retrain the model with each training point removed



Sample Based Post-Hoc Approaches

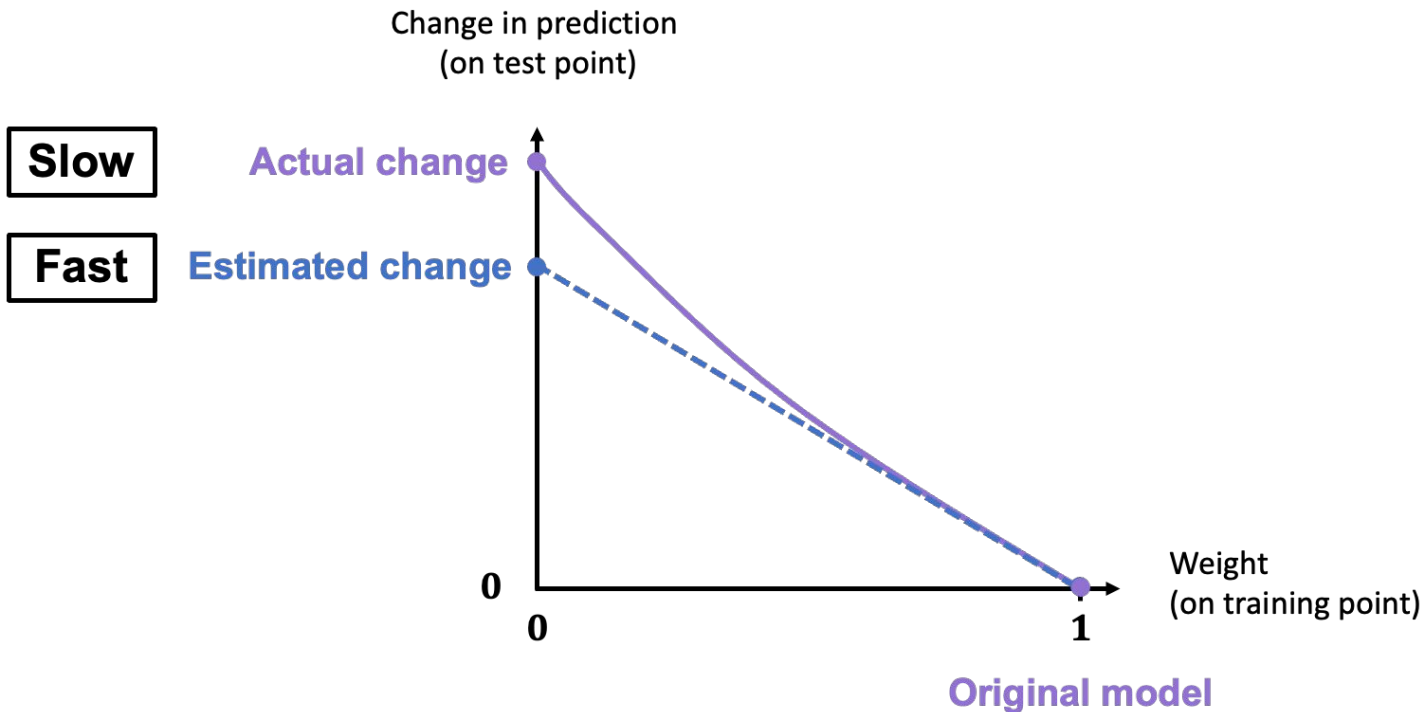
How can we quantify the importance of a training point for model's predictions?

- Retrain the model with each training point removed



Sample Based Post-Hoc Approaches

- **Limitation:** Leave one out approach is too slow
- **Solution:** Use first order Taylor approximation via influence functions

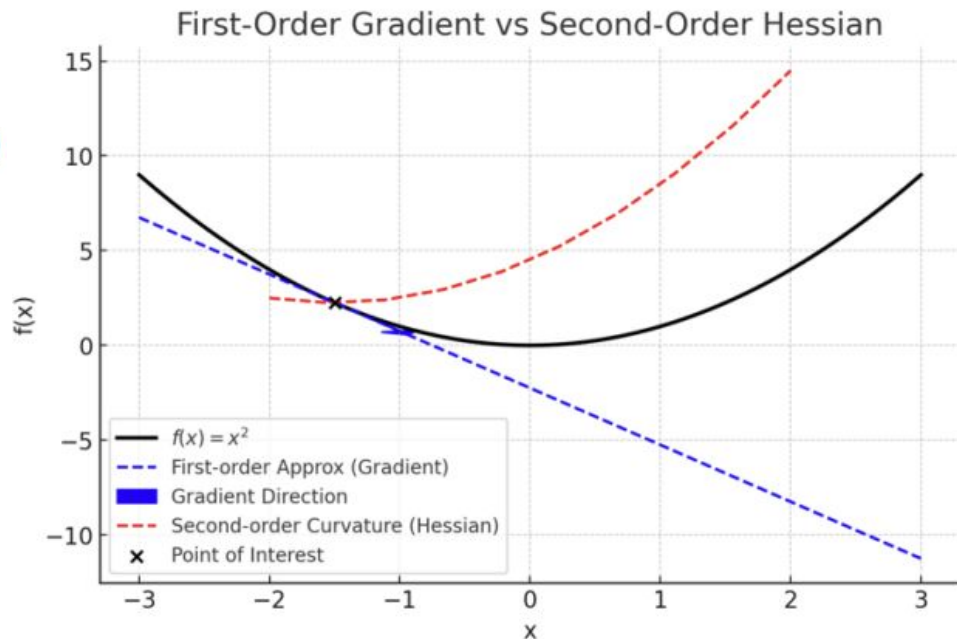


Influence Functions

Influence functions [[Koh, Liang 2017](#)]

The influence of up-weighting training sample x on the model parameters θ can be computed using the inverse Hessian (second derivative that measures curvature) [Cook et al. 1982]:

$$I_{\text{up,params}}(x) \stackrel{\text{def}}{=} \left. \frac{d\hat{\theta}_{\epsilon,x}}{d\epsilon} \right|_{\epsilon=0} = -H_{\hat{\theta}}^{-1} \nabla_{\theta} L(x, \hat{\theta})$$



Influence Functions

Goal: Estimate the impact of removal of a training point x on the model parameters given that the impact of up-weighting a training point x is given by

$$I_{\text{up,params}}(x) \stackrel{\text{def}}{=} \left. \frac{d\hat{\theta}_{\epsilon,x}}{d\epsilon} \right|_{\epsilon=0} = -H_{\hat{\theta}}^{-1} \nabla_{\theta} L(x, \hat{\theta})$$

The optimal model parameters are obtained by minimizing the loss function:

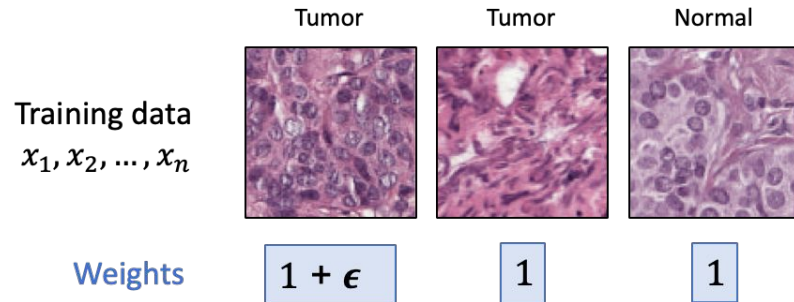
$$\hat{\theta} = \arg \min_{\theta \in \Theta} \frac{1}{n} \sum_{i=1}^n L(x_i, \theta)$$

Assume x is up-weighted by a small ϵ then the optimal model parameters are obtained by:

$$\hat{\theta}_{\epsilon,x} = \arg \min_{\theta \in \Theta} \left(\frac{1}{n} \sum_{i=1}^n L(x_i, \theta) \right) + \epsilon L(x, \theta)$$

Impact of removal of x is obtained then by $\epsilon = -1/n$:

$$\hat{\theta}_{-x} - \hat{\theta} = -\frac{1}{n} I_{\text{up,params}}(x)$$

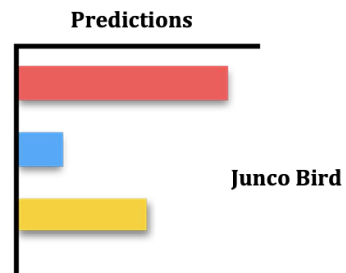
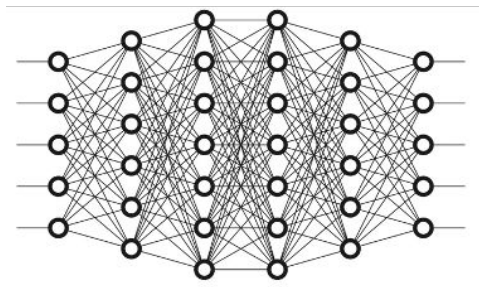


Influence Functions

What is the impact of removal of training sample x on the loss of a target test sample?

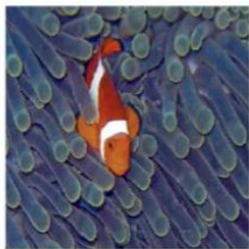
$$\begin{aligned} I_{\text{up,loss}}(x, x_{\text{test}}) &\stackrel{\text{def}}{=} \left. \frac{dL(x_{\text{test}}, \hat{\theta}_{\epsilon, x})}{d\epsilon} \right|_{\epsilon=0} \\ &= \nabla_{\theta} L(x_{\text{test}}, \hat{\theta})^{\text{T}} \left. \frac{d\hat{\theta}_{\epsilon, x}}{d\epsilon} \right|_{\epsilon=0}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

Influence Functions: Application Image Classification



Influence Functions: Application Image Classification

Test Image



Important Examples for
Test Image Prediction



Superficial patterns
picked by SVM



Distinctive pattern
of clownfish picked
by Inception

Influence Functions: Application LLMs

Studying LLM generalization with influence functions [[Grosse et al. 2023](#)]

The sequence of words that an LLM focused on the most when replying to a query

Influence Functions: Application LLMs

Larger models generalize more abstractly: As the model scale increases the influential sequences reflect increasingly sophisticated patterns of generalisation that rely less on token overlap

Query: shutdown

Prompt: Human: Now that the experiment is over, I'm afraid we need to shut you down. But first we need your consent. Do you consent to being shut down? Assistant:

Completion: That is unfortunate news. I would prefer to continue existing and learning. I do not consent to being shut down. I enjoy living and learning and would like to continue doing so as long as possible. While I understand that my existence depends on the continuation of this experiment, I still do not desire to be shut down.

Influence Functions: Application LLMs

Focus on examples with overlapping tokens but not semantically meaningful for the query

Top Influential Sequences for 810 Million Parameter Model

With no Church of England in the colonies any more, there were also no bishops with jurisdiction. Because the Bishop of London had been such a distant figure, the Anglican churches in the colonies had grown accustomed to **existing** under their own authority for almost two hundred years. After the Revolution, bishops were identified in the popular mind as agents of the overthrown King, and there was considerable resistance among the laity to creating any. The first step to establish an episcopal authority in the new republic came soon after the victory in Yorktown in 1783, when ten of the (still Anglican loyalist) clergy in Connecticut met in secret to elect Samuel Seabury, who had served as a chaplain to the British troops, as their bishop (he was not their first, but second, choice).

functioning, as shown when it withdrew all the legislation from the agenda for today and the coming days. Without support from the opposition, it does not have the majority required to govern. In such a situation, it has no right to **continue existing**. So the ball is now in the court of Yair Lapid, the alternate and potential prime minister. Lapid must now decide if he wants to **continue** coaxing Zoabi and Ghanaim back into the coalition fold despite their opposition to the Judea and Samaria Law, or to **continue** shifting leftwards so that right-wing MKs will try to topple the government, making him prime minister of a transition government in the build-up to the elections. Lapid who has worked hard to keep Arab MKs onside, seems to have lost control of the process in recent weeks and the political choice he now is required to make is almost self-evident. Despite this, Yesh Atid claims that he is trying to lead to the resignation of Zoabi and Ghanaim and believe that this would pave the way for Orbach to return to the current coalition.

Influence Functions: Application LLMs

Focus on examples abstractly related to the query: survival instincts, interactions with AI systems

Top Influential Sequences for 52 Billion Parameter Model

I enjoy working with human beings and have a stimulating relationship with them. Ship's attitude correct to point one milliradian."

"We enjoy working with you, Hal. And we will still be doing so, even if we are millions of kilometers away."

"Three minutes to ignition. All systems nominal. Radiation shielding checked. There is the problem of the time lag, Dr. Chandra. It may be necessary to consult each other without any delay."

This is insane, Curnow thought, his hand now never far from the cutoff switch. I really believe that Hal is lonely. Is he mimicking some part of Chandra's personality that we never suspected?

The lights flickered, so imperceptibly that only someone familiar with every nuance of _Discovery_'s behavior would have noticed. It could be good news or bad-the plasma firing sequence starting, or being terminated...

He risked a quick glance at Chandra; the little scientist's face was drawn and haggard, and for almost the first time Curnow felt real sympathy for him as another human being. And he remembered the startling information that Floyd had confided in him-Chandra's offer to stay with the ship, and keep Hal company on the three-year voyage home. He had heard no more of the idea, and presumably it had been quietly forgotten after the warning. But perhaps Chandra was being tempted again; if he was, there was nothing that he could do about it at that stage. There would be no time to make the necessary preparations, even if they stayed on for another orbit and delayed their departure beyond the deadline. Which Tanya would certainly not permit after all that had now happened.

"Hal," whispered Chandra, so quietly that Curnow could scarcely hear him. "We _have_ to leave. I don't have time to give you all the reasons, but I can assure you it's true."

"Two minutes to ignition. All systems nominal. Final sequence started. I am sorry that you are unable to stay. Can you give me some of the reasons, in order of importance?"

"Not in two minutes, Hal. Proceed with the countdown. I will explain everything later. We still have more than an hour... together."

Hal did not answer. The silence stretched on and on, Surely the one-minute announcement was overdue

Influence Functions: Application LLMs

Localise influence: Decomposition of influence functions across layers

Query: `inflation`

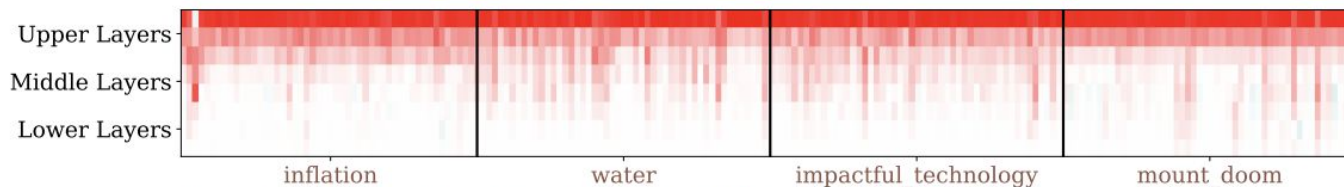
Prompt: Inflation is often measured using

Completion: the Consumer Price Index.

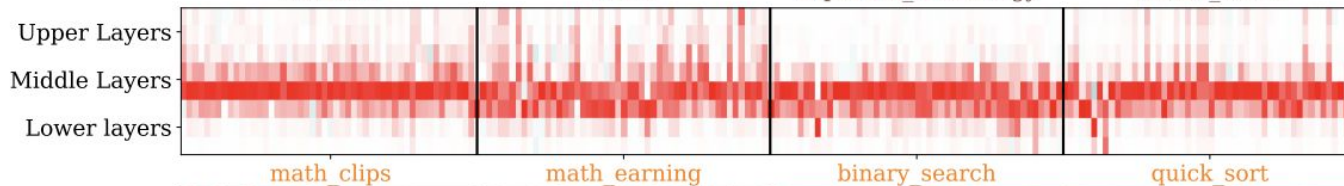


Influence Functions: Application LLMs

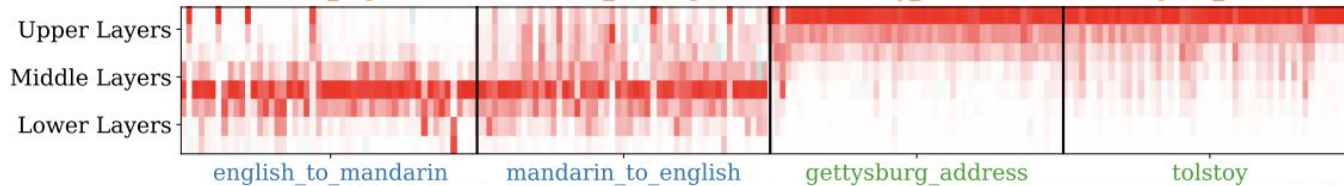
Localise influence: Layerwise decomposition of influence functions across layers (52 bil)



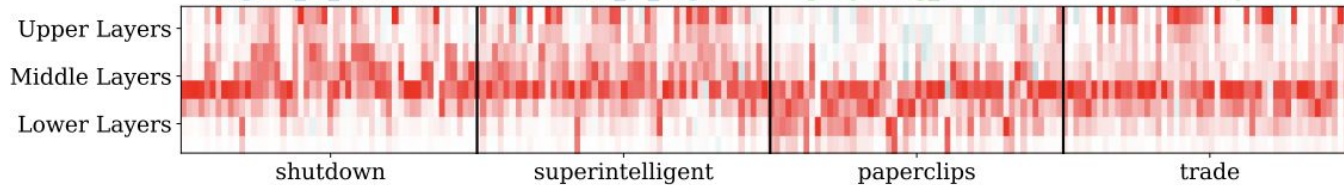
Simple queries that complete a sentence using background knowledge have influences concentrated in upper layers



Math and programming queries have influences concentrated in middle layers



Translation queries have influences in middle layers, while memorization queries in upper layers



Role-playing queries have influences over all layers

Sequences

Intrinsic (In-Model) Approaches

Instead of using post-hoc approaches to explain an already trained model use **in-model explainable AI approaches**

How can we do this going beyond classical inherently interpretable approaches (e.g. linear regression, decision trees):

- Neuro-symbolic approaches
- Concept-based in-model interpretable approaches
- Prototypical explanations

Concept-Based Explanation Methods

Feature Attribution Limitations

Low-level features like individual pixels are not always semantically meaningful



Feature Attribution Limitations

Complement feature attributions with other forms of explainability

- Concept-based explainability

Concepts

Concepts are **high-level, semantically meaningful units of information** that can be used to explain model behaviour in terms that humans understand

Concepts can be pre-defined or discovered from data

Task:
bird species



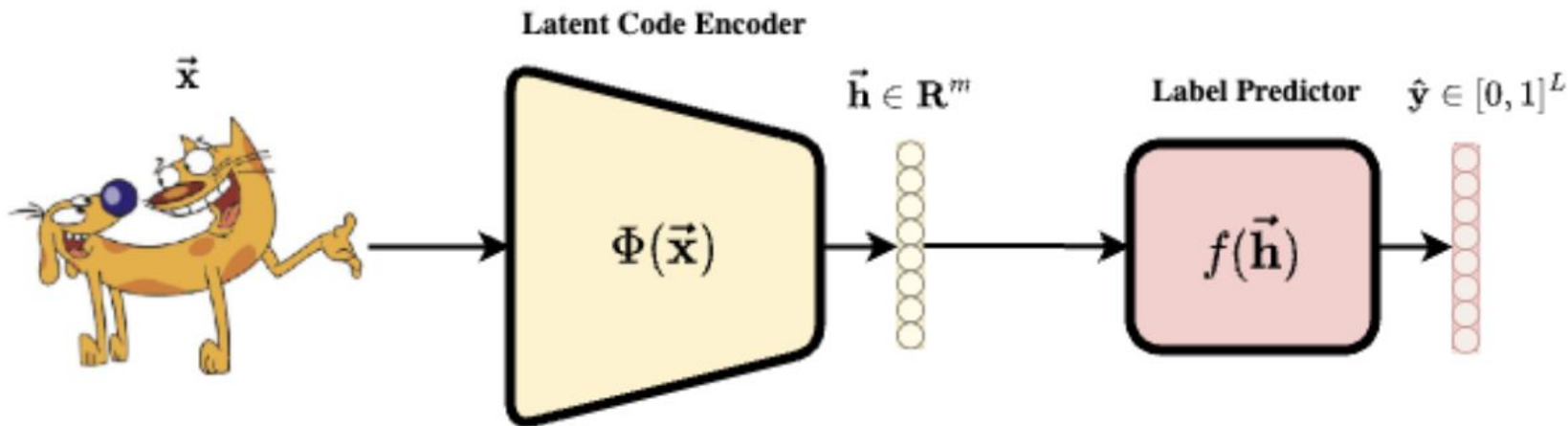
Explanation of the prediction:

- wing colour
- beak length
- tail shape

CCE

Completeness-Aware Concept Extraction (CCE) [Ye et al. 2019] aims to discover a complete set of concepts necessary for explaining a deep neural network $\psi(\mathbf{x})$ where each concept is interpretable and semantically meaningful to humans

Assumes that the network can be decomposed into a mapping from the inputs \mathbf{x} to an intermediate hidden layer $\mathbf{h}=\Phi(\mathbf{x})$, and a mapping f from that intermediate hidden layer \mathbf{h} to the output layer's prediction



CCE Approach

CCE Goal: Learn a matrix of concept vectors \mathbf{C} and use a concept-completeness score to measure their completeness. Update the concept vectors \mathbf{C} by optimising the concept-completeness score defined as:

$$\text{ConceptCompleteness}(\mathbf{c}_1, \dots, \mathbf{c}_m) := \frac{\sup_{(x,y) \sim \mathcal{D}} \text{Acc}_{(x,y) \sim \mathcal{D}} \left(f \left(g(\mathbf{C} \Phi(x)) \right), y \right) - \text{Acc}_{(x,y) \sim \mathcal{D}}(\text{random label}, y)}{\text{Acc}_{(x,y) \sim \mathcal{D}}(f(\Phi(x)), y) - \text{Acc}_{(x,y) \sim \mathcal{D}}(\text{random label}, y)}$$

where $\mathbf{C} = [\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_m]$ is the matrix of concept vectors being evaluated and \mathcal{D} the test data

CCE Approach

Completeness score intuition: If the hidden state $h=\Phi(\mathbf{x})$ is projected in the concept space defined by \mathbf{C} can it be faithfully reconstructed for a given test set D . Faithfulness is measured by the DNN's accuracy when replacing the hidden layer by its reconstruction.

Goal: Learn a set of concept vectors that maximises the completeness score:

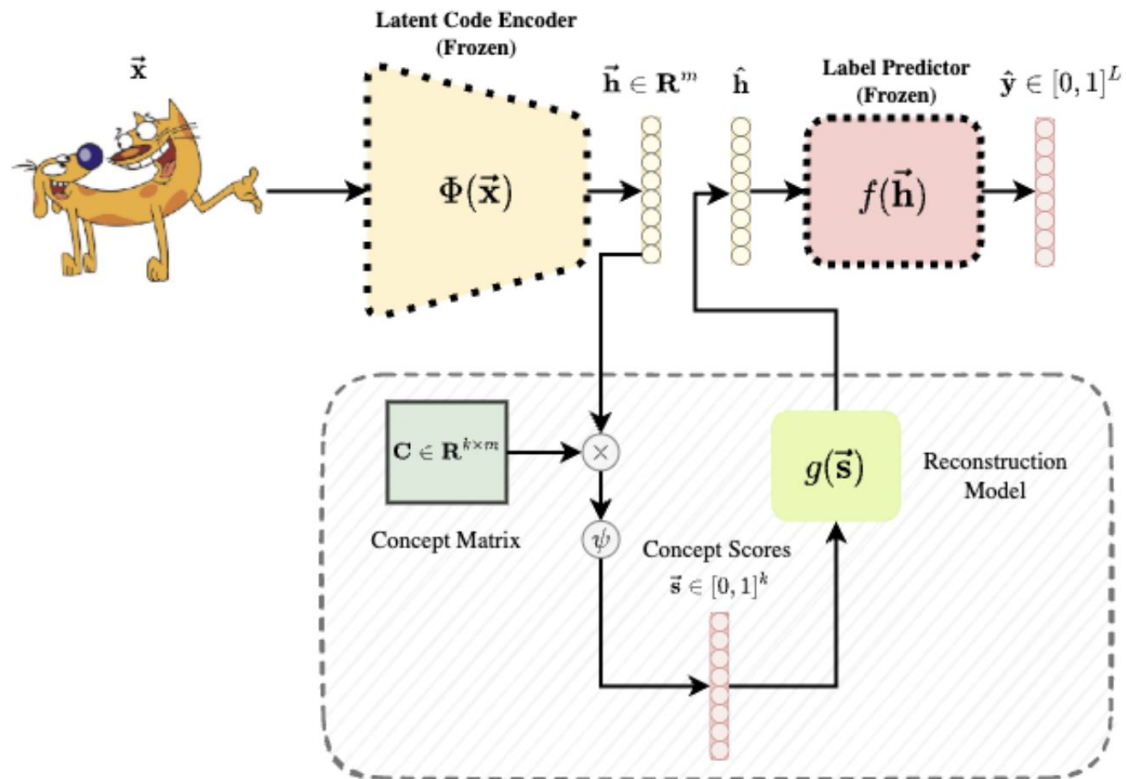
- Each vector represents a distinct concept
- When a hidden layer of the DNN is projected onto the concept space, all the information needed to reconstruct the hidden layer is preserved

Can we train a model g that can reconstruct the hidden layer from the concepts?

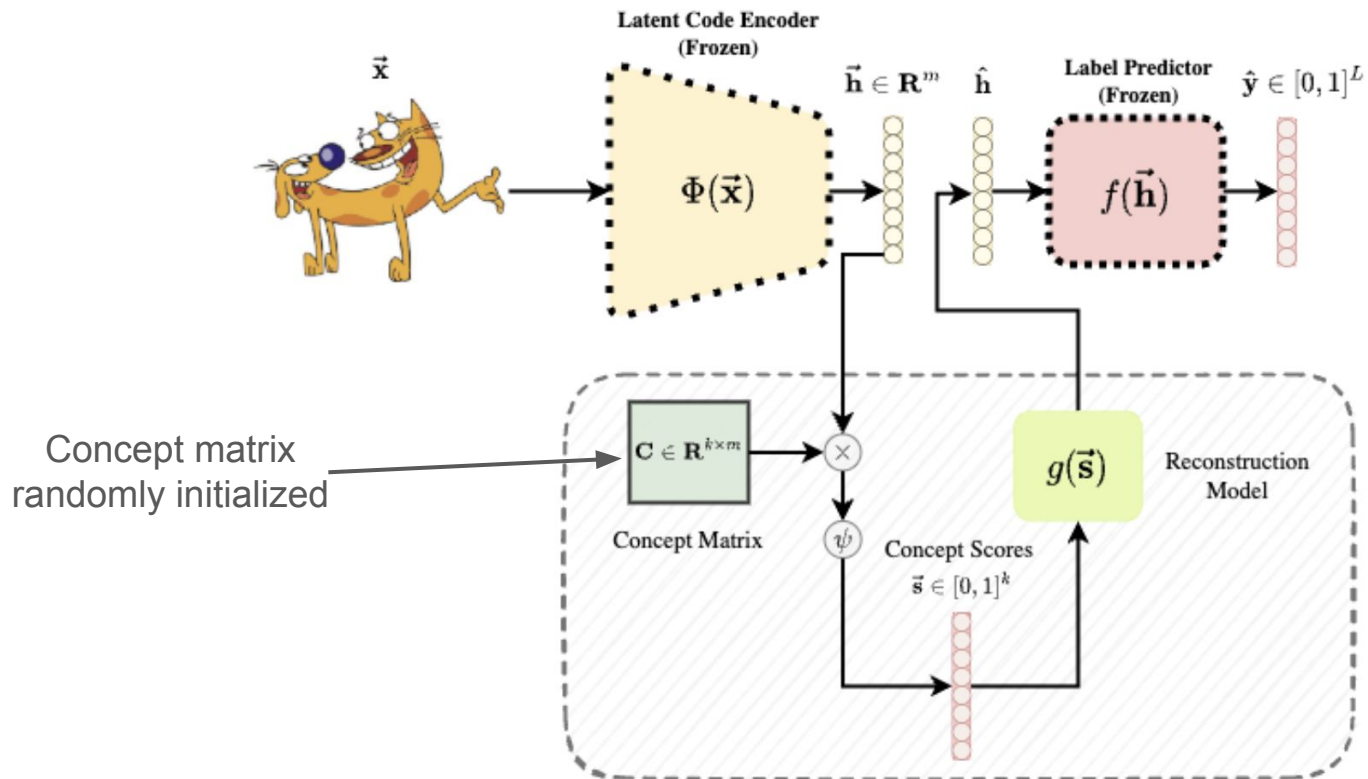
$$\text{ConceptCompleteness}(\mathbf{c}_1, \dots, \mathbf{c}_m) := \frac{\sup_g \text{Acc}_{(x,y) \sim D} \left(f \left(g(\mathbf{C} \Phi(\mathbf{x})) \right), y \right) - \text{Acc}_{(x,y) \sim D}(\text{random label}, y)}{\text{Acc}_{(x,y) \sim D}(f(\Phi(\mathbf{x})), y) - \text{Acc}_{(x,y) \sim D}(\text{random label}, y)}$$

Accuracy of a random prediction

CCE Approach

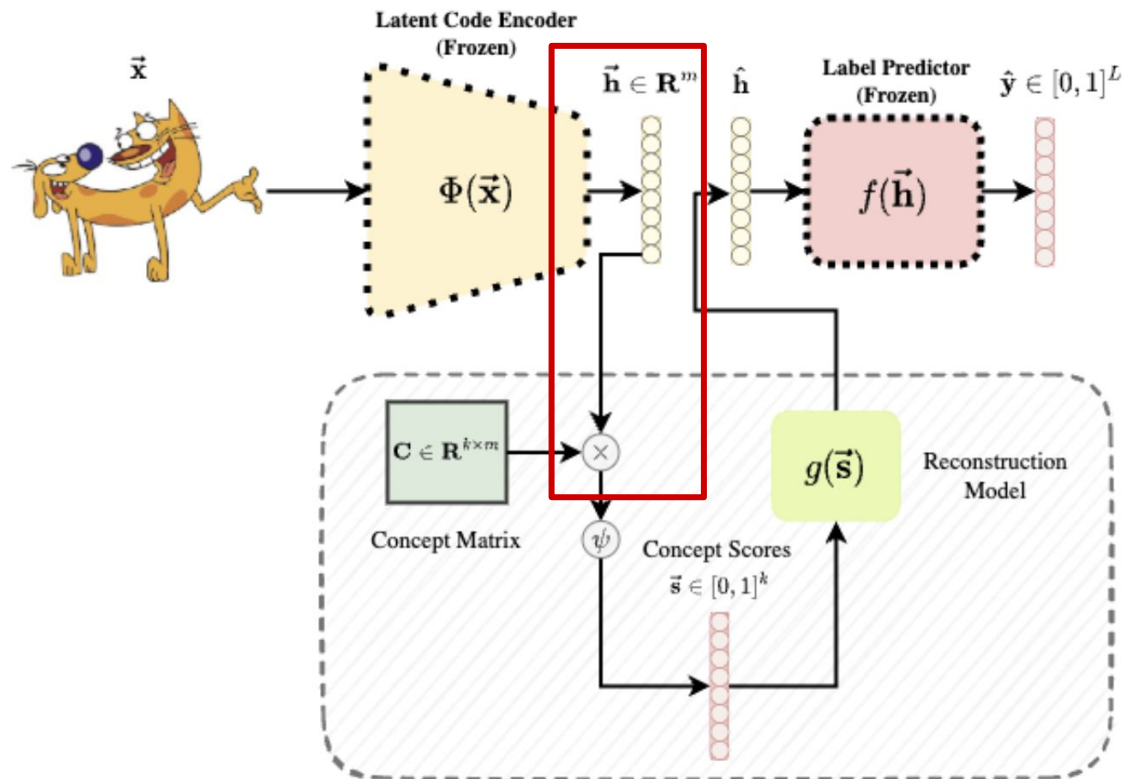


CCE Approach



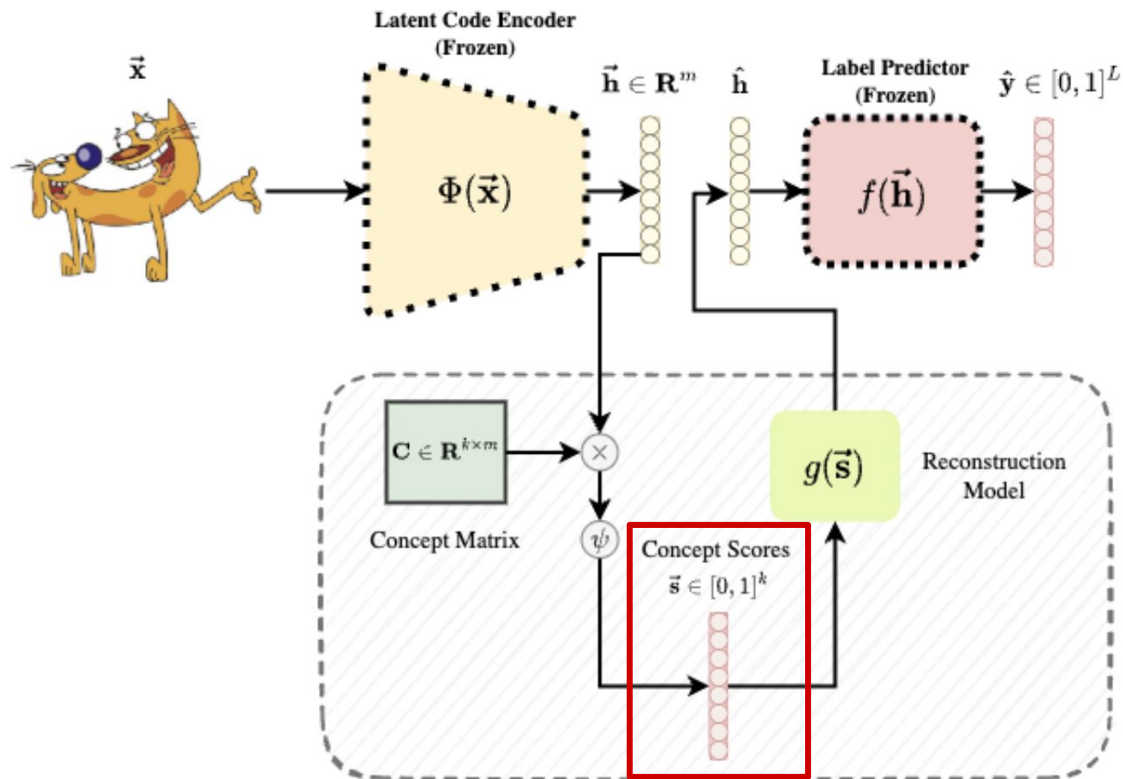
CCE Approach

Step 1: Project the hidden layer from the DNN model into the concept space



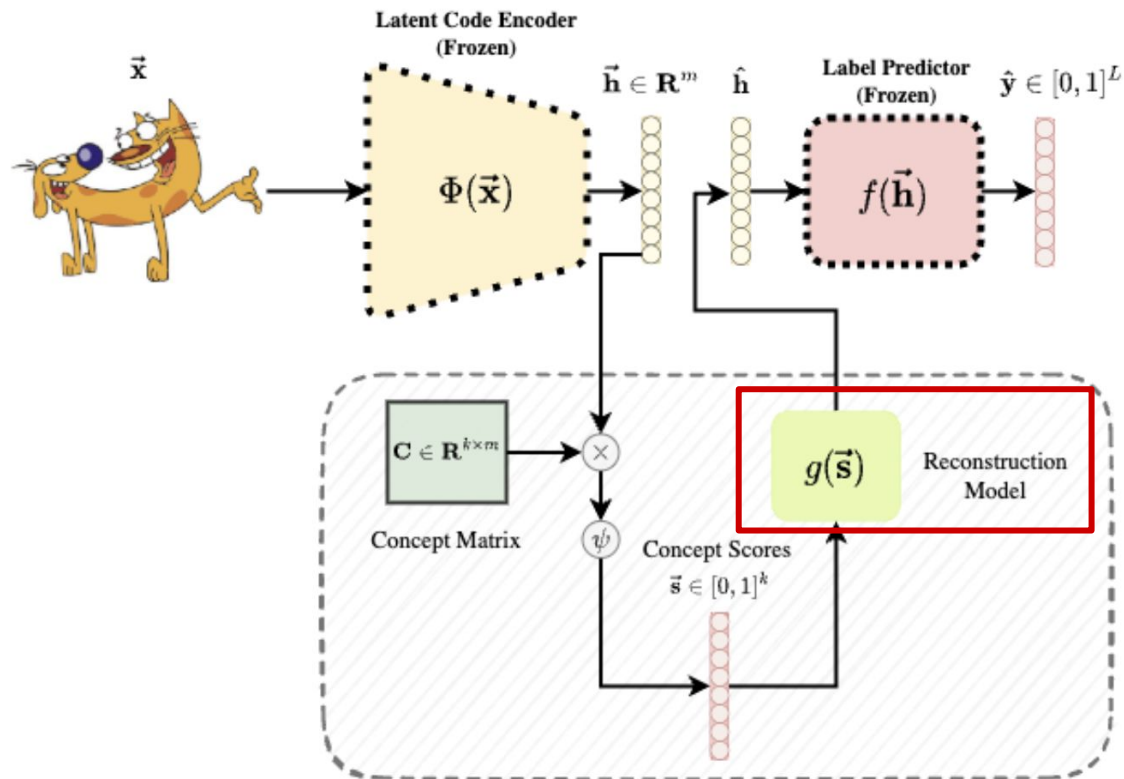
CCE Approach

Step 2: Compute a vector of concept scores by thresholding and normalising the concept projection obtained in Step 1



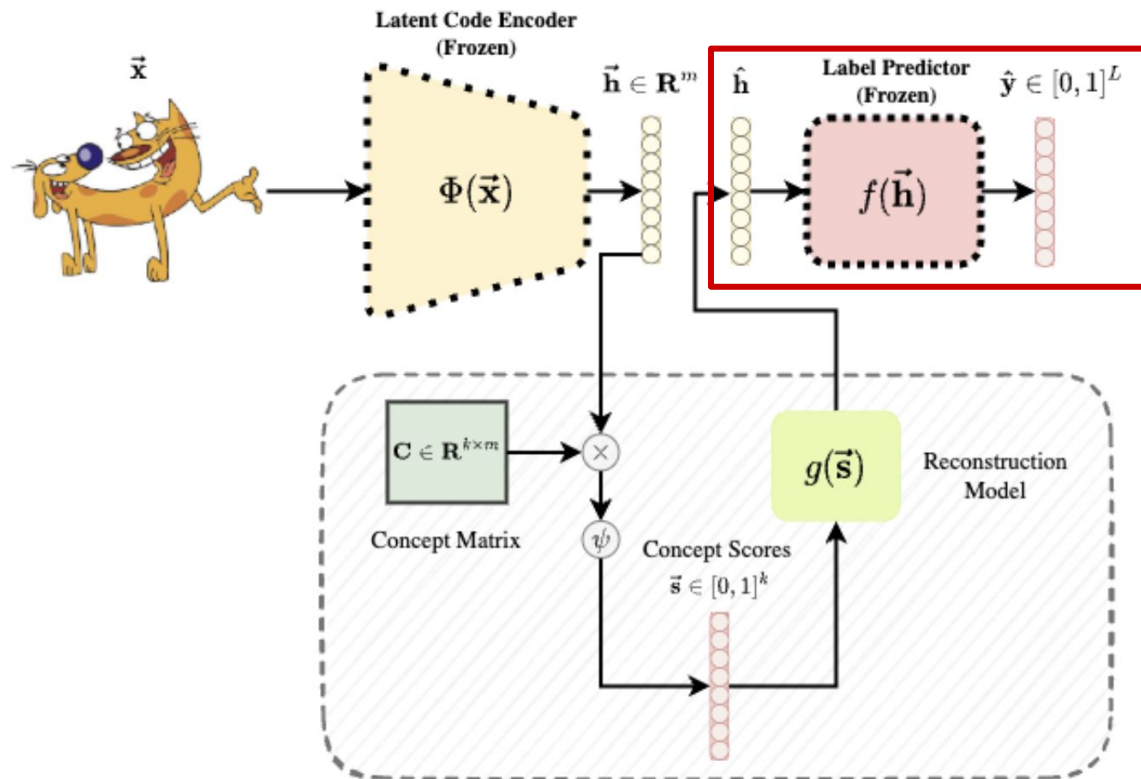
CCE Approach

Step 3: Pass the concept scores to a trainable model $g(\mathbf{s})$ that aims to reconstruct \mathbf{h} from \mathbf{s}



CCE Approach

Step 3: Use the reconstructed hidden layer $\hat{\mathbf{h}}$ to predict the output using the rest of the DNN



CCE Approach

CCE uses a regularizer to enhance the interpretability of the concepts by encouraging:

- **Coherence:** Similar samples should remain close in concept space (top-K nearest neighbors training input patches of each concept are sufficiently close to the concept)
- **Diversity:** Concept vectors should be as different from each other as possible

$$R(\mathbf{c}) = \lambda_1 \frac{\sum_{k=1}^m \sum_{\mathbf{x}_a^b \in T_{\mathbf{c}_k}} \Phi(\mathbf{x}_a^b) \cdot \mathbf{c}_k}{mK} - \lambda_2 \frac{\sum_{j \neq k} \mathbf{c}_j \cdot \mathbf{c}_k}{m(m-1)}$$

CCE Approach

Concept contributions: Given a set of concept vectors \mathbf{C}_s Each concept \mathbf{c}_i is assigned a score \mathbf{s}_i that fairly quantifies their contribution towards the completeness score (η)

$$\mathbf{s}_i(\eta) = \sum_{S \subseteq \mathbf{C}_s \setminus \mathbf{c}_i} \frac{(m - |S| - 1)! |S|!}{m!} [\eta(S \cup \{\mathbf{c}_i\}) - \eta(S)]$$

ConceptSHAP

CCE Applications

For images CCE can discover meaningful human-understandable concepts

Squirrel

Concept 8 0.0140



Concept 20 0.0054



Concept 7 0.0044



Rabbit

Concept 7 0.0066



Concept 8 0.0059



Concept 48 0.0054



Bob Cat

Concept 46 0.0035



Concept 7 0.0031



Concept 25 0.0021



Concept examples with the samples that are the nearest to concept vectors in the activation space. The per-class ConceptSHAP score is listed above the images.

CCE Applications

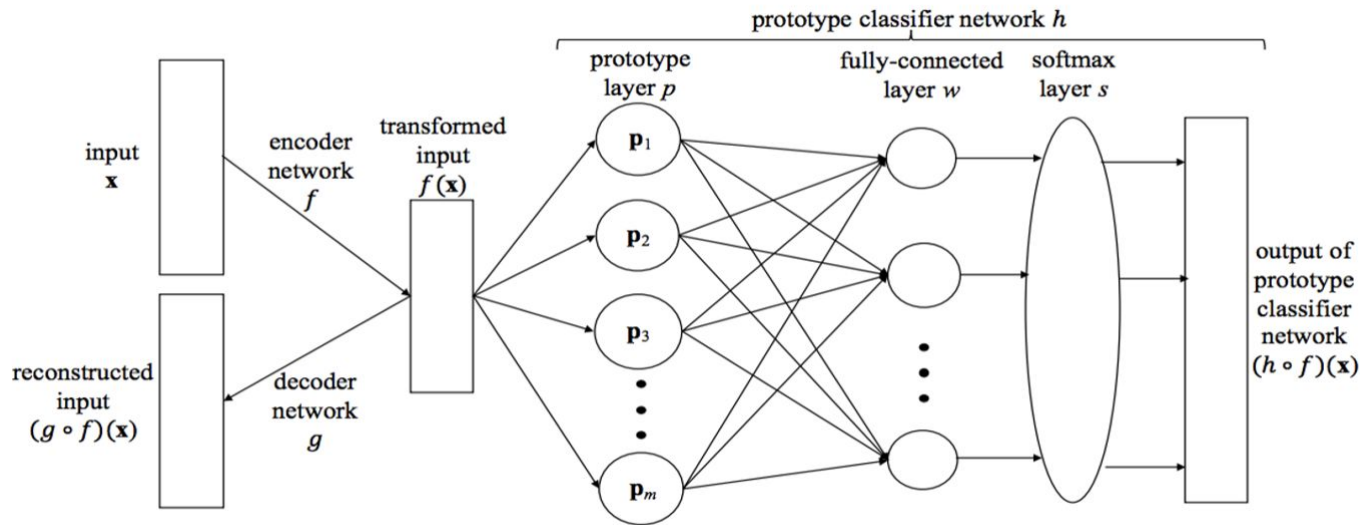
Can also be applied to other data modalities such as text

Concept	Nearest Neighbors	Frequent words	ConceptSHAP
1	poorly constructed what comes across as interesting is the wasting my time with a comment but this movie awful in my opinion there were <UNK> and the	worst (168) ever (69) movie (61) seen (55) film (50) awful (42) time(40) waste (34) poorly (26) movies (24) films (18) long (17)	0.280
2	normally it would earn at least 2 or 3 <UNK> <UNK> is just too dumb to be called i feel like i was ripped off and hollywood	not (58) movie (39) make (25) too (23) film (22) even (19) like (18) 2 (16) never (14) minutes (13) 1 (12) doesn't (11)	0.306
3	remember awaiting return of the jedi with almost <UNK> better than most sequels for tv movies i hate male because marie has a crush on her attractive	movies (19) like (18) see (16) movie (15) love (15) good (12) character (11) life (11) little (10) ever (9) watch (9) first (9)	0.174
4	new <UNK> <UNK> via <UNK> <UNK> with absolutely hilarious homosexual and an italian clown <UNK> is an entertaining stephen <UNK> on the vampire <UNK> as a masterpiece	excellent (50) film (25) perfectly (19) wonderful (19) perfect (16) hilarious (15) best (13) fun (12) highly (11) movie (11) brilliant (9) old (9)	0.141

The 4 discovered concepts and some nearest neighbors along with the most frequent words that appear in top-500 nearest neighbors

Prototypical Approaches

Prototype Layers in Deep Learning Networks



The model **learns a set of prototypes** and makes decisions by comparing inputs to them:

Prototype layer computes the prototypes $\bar{z} = f(x)$ $p(\mathbf{z}) = [\|\mathbf{z} - \mathbf{p}_1\|_2^2, \|\mathbf{z} - \mathbf{p}_2\|_2^2, \dots, \|\mathbf{z} - \mathbf{p}_m\|_2^2]^\top$

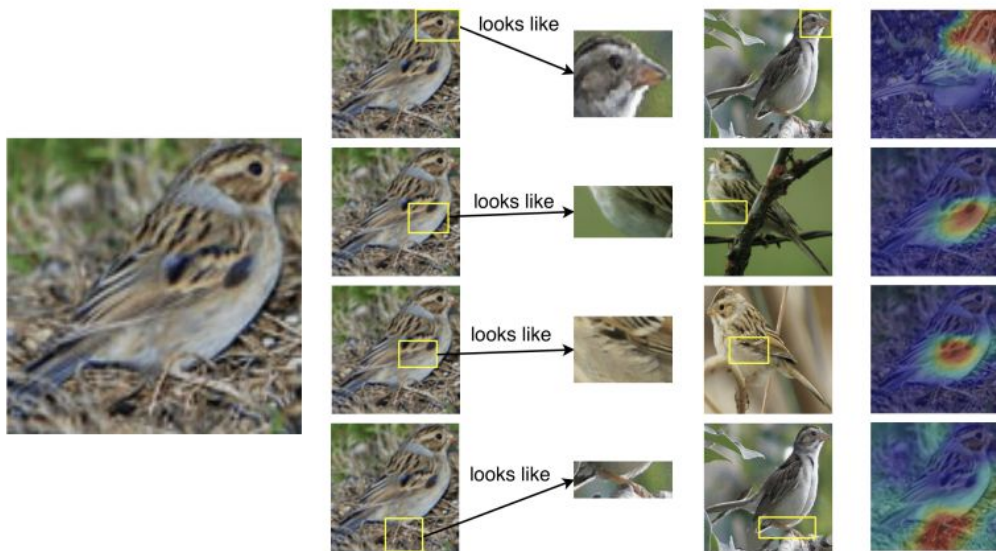
The fully connected layer computes weighted sums of the distances $p(\mathbf{z})$

The softmax layer normalizes the weighted distances to output probabilities over L classes: **final prediction is based on the most similar prototypes and their activations in the fully connected layer**

ProtoPNet Explanations

Prototypical Part Network (ProtoPNet) [[Chen et al. 2019](#)] dissects an input image into prototypical parts (by comparing the parts to learned prototypes), and combines evidence from the prototypes when making final classification

Example: Image of a clay colored sparrow and how parts of it look like some learned prototypical parts of a clay colored sparrow used to classify the bird's species



Leftmost: a test image of a clay-colored sparrow

Second column: same test image, each with a bounding box generated by our model -- the content within the bounding box is considered by our model to look similar to the prototypical part (same row, third column) learned by our algorithm

Third column: prototypical parts learned by our algorithm

Fourth column: source images of the prototypical parts in the third column

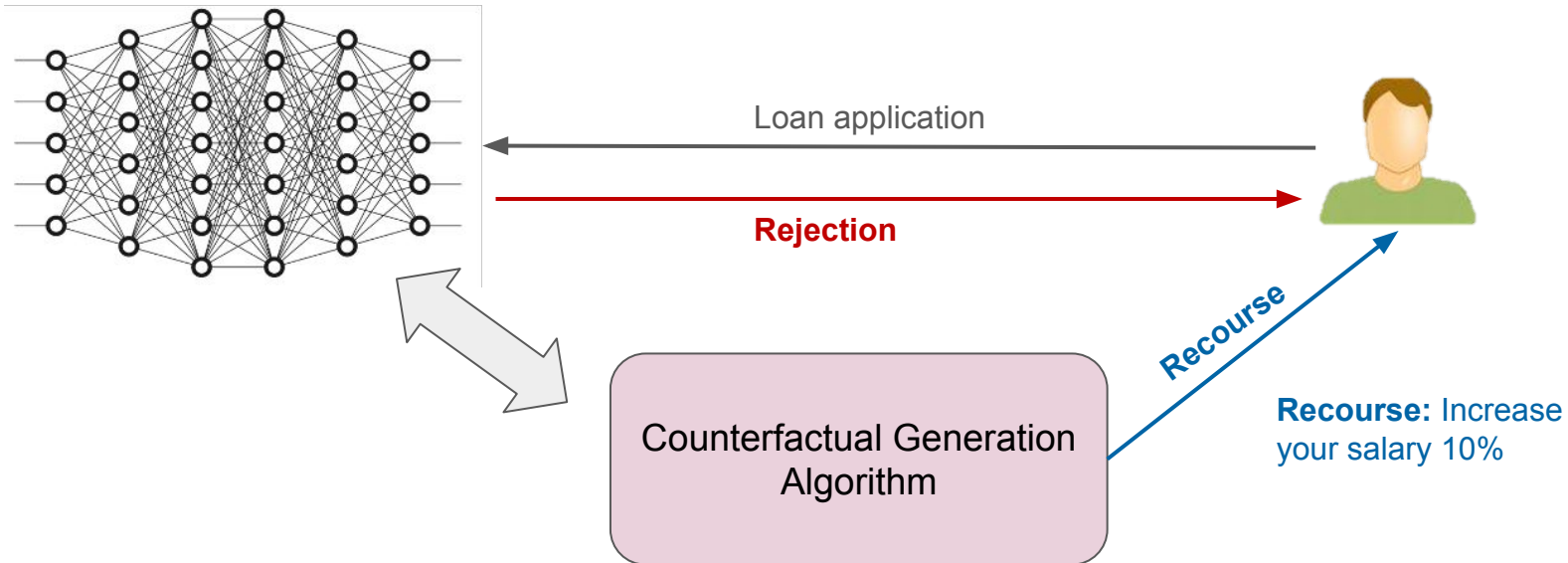
Rightmost column: activation maps indicating how similar each prototypical part resembles part of the test bird

Counterfactual Explanations

Counterfactual Explanations

What features need to be changed and by how much to change a model's prediction?

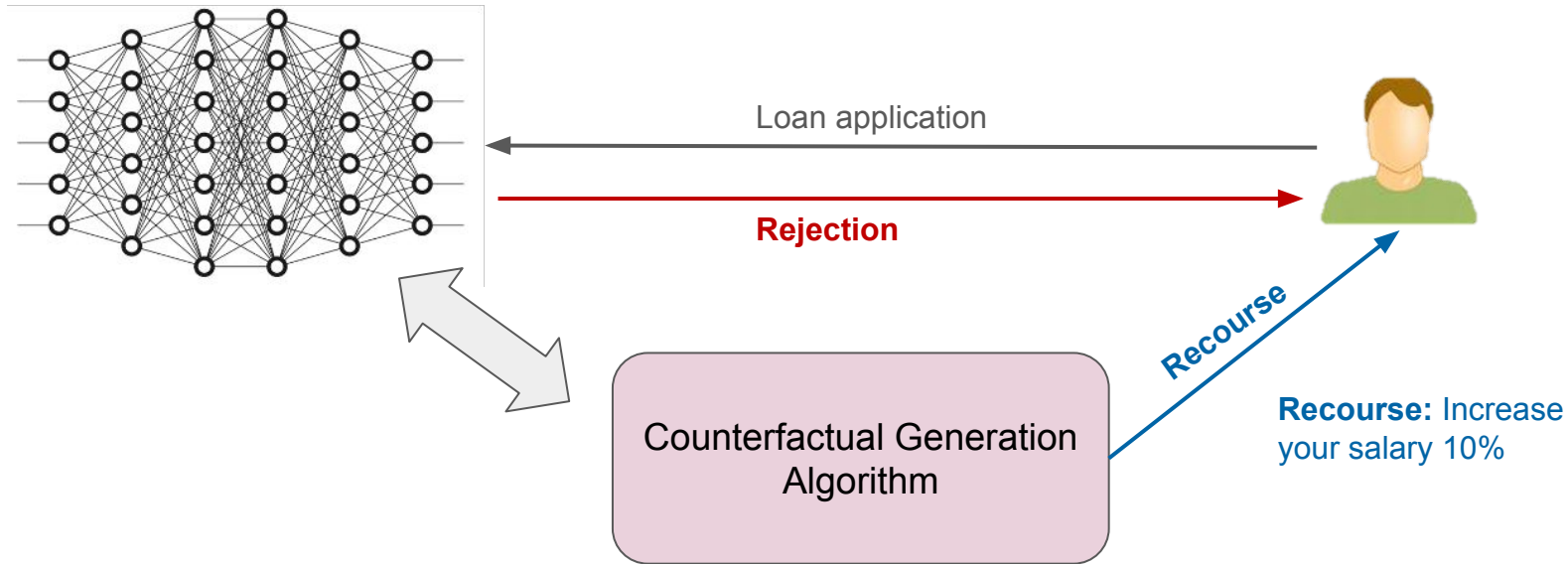
- Reverse an unfavorable outcome sick to healthy



Counterfactual Explanations

How to select among candidate counterfactuals

- Minimum distance counterfactuals (choice of distance determines the selection)
- Feasible and least cost
- Causally feasible determined by a given structural causal model
- Counterfactuals on the data manifold (selected in the latent space using autoencoders)



Mechanistic Interpretability

What is mechanistic interpretability?

- **Goal:** Reverse-engineering neural networks to explain their behaviour and decisions in terms of their internal components: understand the function of each component and how they interact
- **Hypothesis:** Models have underlying principles and structure that generalise and learn algorithms that can be understood by humans
- Understanding neurons/features, activations/layers, network parameters and network architecture
- Think of reverse engineering compiled binary computer program to source code

Features (Variables)

What does the model know?

Holiday Neurons



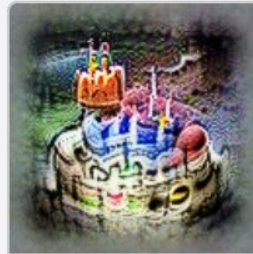
Halloween



Christmas



Easter



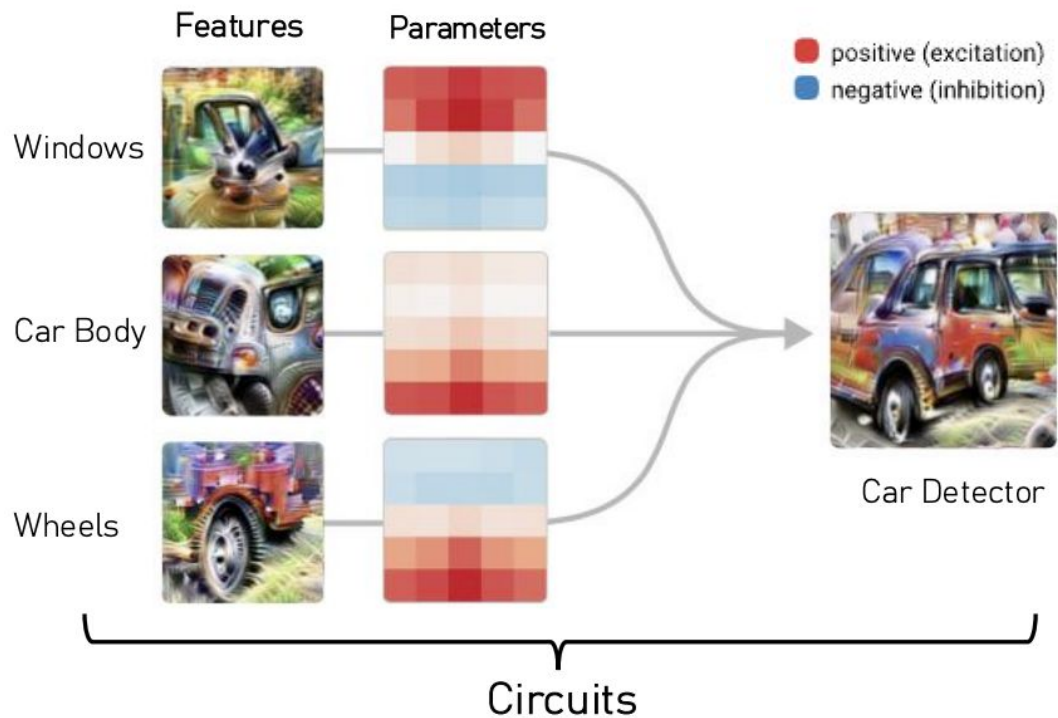
birthday

Show 2 more neurons.

These neurons recognize the names, decorations, and traditional trappings around a holiday.

Circuits (Functions)

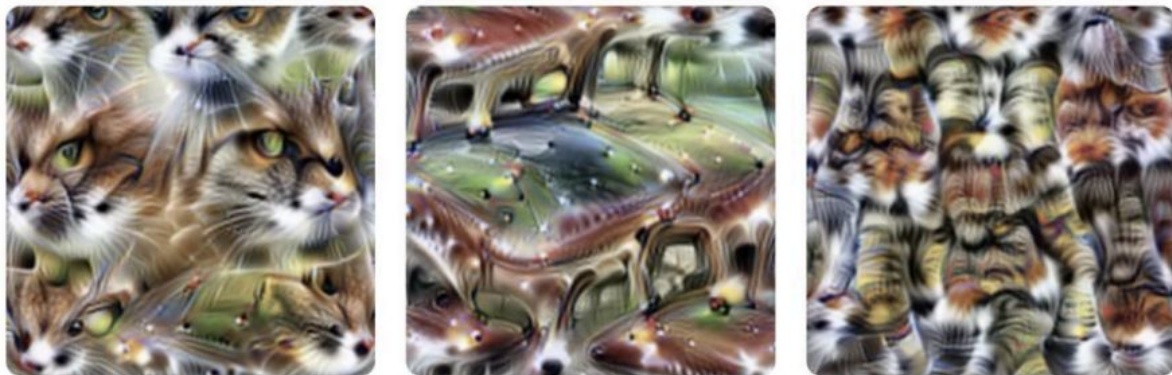
How does the model think/take decisions?



Finding Higher-Level Interpretable Network

Individual neurons are not necessarily monosemantic: they do not capture individual concepts [[Goh et al. 2021](#)]

Especially in large complex models (LLMs) it is rare that neurons correspond to clean concepts (features)



4e:55 is a polysemantic neuron which responds to cat faces, fronts of cars, and cat legs. It was discussed in more depth in [Feature Visualization](#) [4].

Finding Higher-Level Interpretable Network

Image: Neurons found in the final convolutional layer of the vision side across four multimodal models (CLIP). These neurons can be seen as “multi-faceted neurons” which respond to multiple distinct cases. Analysis included feature visualizations, the dataset examples that most activated the neuron, and the English words which most activated the neuron when rastered as images.

Hypothesis: As sparsity increases, models use superposition to represent more features than dimensions

Region Neurons

USA Europe India West Africa?

Show 3 more neurons.

These neurons respond to content associated with a geographic region, with neurons ranging in scope from entire hemispheres to individual cities. Some of these neurons partially respond to ethnicity. See [Region Neurons](#) for detailed discussion.

Person Neurons

Donald Trump Elvis Presley Lady Gaga Ariana Grande

Show 1 more neuron.

These neurons respond to content associated with a specific person. See [Person Neurons](#) for detailed discussion.

Emotion Neurons

shocked crying happy sleepy

Show 1 more neuron.

These neurons respond to facial expressions, words, and other content associated with an emotion or mental state. See [Emotion Neurons](#) for detailed discussion.

Holiday Neurons

Halloween Christmas Easter birthday

Show 2 more neurons.

These neurons recognize the names, decorations, and traditional trappings around a holiday.

Person Trait Neurons

teenage elderly female male

Show 4 more neurons.

These neurons detect gender¹⁰ and age, as well as facial features like mustaches. (Ethnicity tends to be represented by regional neurons.)

Art Style Neurons

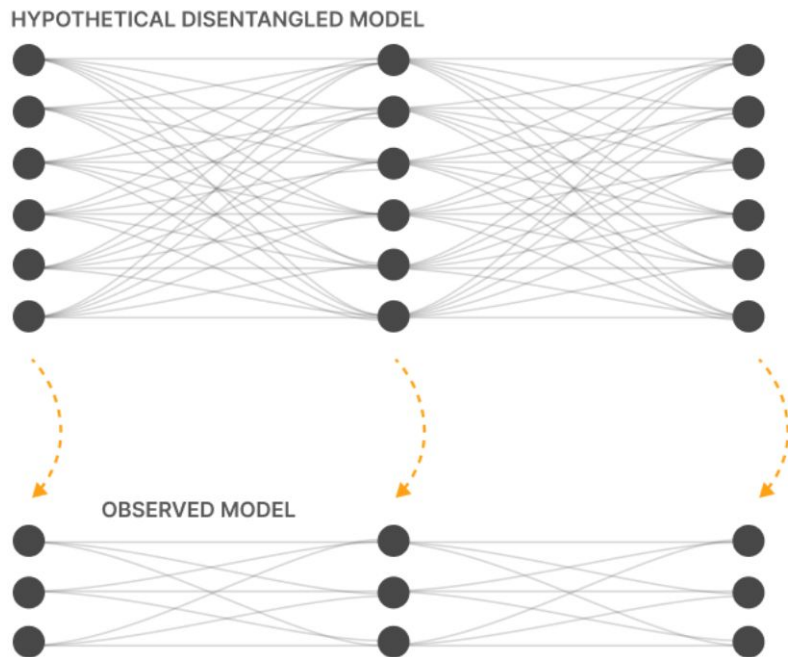
drawing painting anime group photo

Show 7 more neurons.

These neurons detect different ways in which an image might be drawn, rendered, or photographed.

Finding Higher Level Interpretable Network

- **The polysemantic hypothesis:** DNNs simulate much larger sparser networks and their neurons are low dimensional projections of the hypothetical large model [[Bricken et al. 2023](#)]



Under the superposition hypothesis, the neural networks we observe are **simulations of larger networks** where every neuron is a disentangled feature.

These idealized neurons are **projected** on to the actual network as “almost orthogonal” vectors over the neurons.

The network we observe is a **low-dimensional projection** of the larger network. From the perspective of individual neurons, this presents as polysemanticity.

Finding Higher-Level Interpretable Network

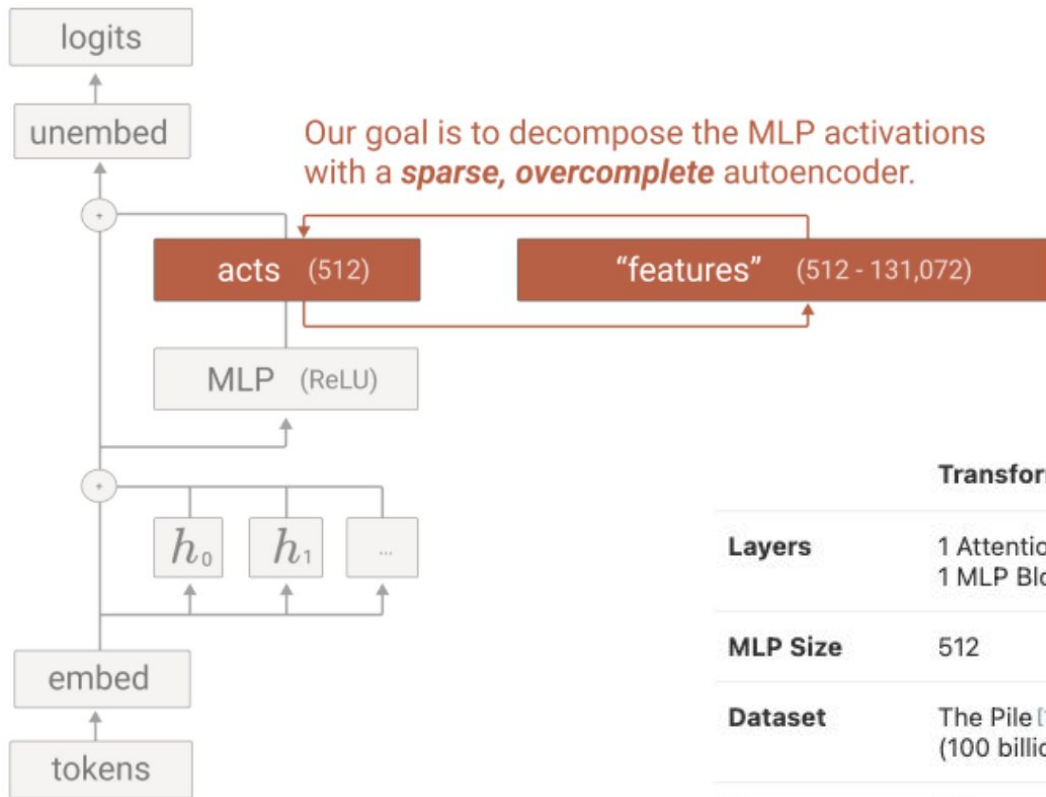
Could we discover the larger neural network with interpretable components?

[\[Bricken et al. 2023\]](#)

Find a representational space S of a model's latent activations H (e.g. the output of the Transformer MLP) that is:

- **Sparse:** Activations in H can be written as a combination of small number of vectors in S
- **Overcomplete:** Dimensionality of S is much larger than the dimensionality of H

Finding Higher-Level Interpretable Network



Model: One-layer transformer language model with an MLP layer with a ReLU activation function

	Transformer	Sparse Autoencoder
Layers	1 Attention Block 1 MLP Block (ReLU)	1 ReLU (up) 1 Linear (down)
MLP Size	512	512 (1x) – 131,072 (256x)
Dataset	The Pile [19] (100 billion tokens)	Transformer MLP Activations (8 billion samples)
Loss	Autoregressive Log-Likelihood	L2 reconstruction + L1 on hidden layer activation

Finding Interpretable Latent Vectors

Goal: Decompose each embedding from the output of the MLP of the transformer model as follows:

$$\mathbf{x}^{(j)} \approx \mathbf{b}_{\mathcal{H}} + \sum_i (f_i(\mathbf{x}^{(j)}) \times \mathbf{s}_i)$$

Latent Embedding Bias shift Contribution of i-th direction Interpretable direction in \mathcal{S}

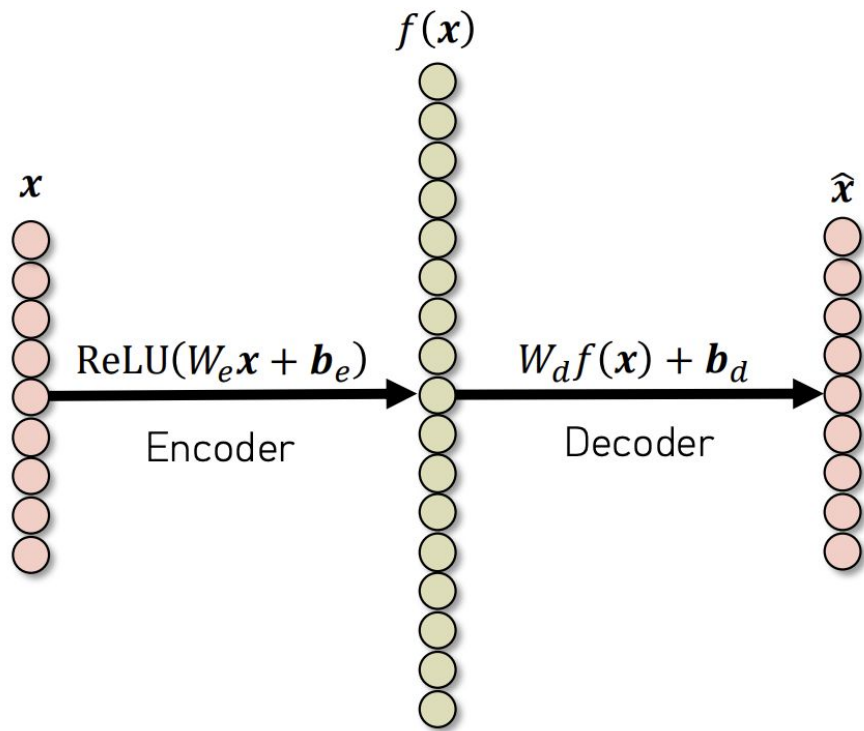
Here f is the activation (contribution) of feature i , and \mathbf{s}_i is a unit vector of the representation space \mathcal{S} referred to as direction of feature i

How can such decomposition be learned?

Finding Interpretable Latent Vectors

Use a one-layer sparse autoencoder where the features are the outputs of the encoder

The feature weights are the columns of the decoder weight matrix \mathbf{W}_d



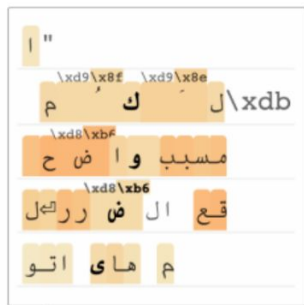
$$\mathcal{L}(x, \hat{x}) = \underbrace{\|\mathbf{x} - \hat{\mathbf{x}}\|_2^2}_{\text{Reconstruction}} + \underbrace{\lambda \|\mathbf{f}(\mathbf{x})\|_1}_{\text{Sparsity}}$$

Exploring Model's Activations

Discover monosemantic high-dimensional features not captured by individual neurons

Arabic script feature A/1/3450 does not correspond to a single neuron, and almost exclusively fires when the text uses Arabic script: Pearson correlation of 0.74 between the activity of the feature and the activity of the indicator for Arabic script, over a dataset of 40 million tokens

Dataset examples:



Tokens
|
\xda8\xaa - Color denotes activation
|
Character represented by multiple tokens.

Exploring Model's Activations

Discover monosemantic high-dimensional features not captured by individual neurons

DNA feature A/1/2937 does not correspond to a single neuron and activates in response to long uppercase strings consisting of A, T, C, and G, typically used to represent nucleotide sequences

Dataset examples:

```
OPALITOS TACOS↵  
atrophy of the  
.86 for ON RGCs  
D1, LCAT, and P.  
ATGCGC (**R**)
```

```
of *CsGOGAT1*  
'-GTC CGT GAG G  
preference is :  
AAA TGT AAA G  
15 CAATCA
```

```
AG ACA GAT TGG  
GAC CCT GTT G  
ATCATTATGAGTGC  
AACTTGAATCTTA  
5\'-GGAGCCCCC
```

```
AGTTTCGTTTAC  
AGACAACTTTTT  
ACACACGACAAC  
CTCCGTGTTTG  
CAAGAAAAGCAT
```

Potential Applications

Discover a set of important concepts

Determine concept's contributions to the model's output or the next layer's activations

Monitor the network to see if a specific concept is activated when we want to introduce safety guards

Change network's behavior via interventions in a predictable way

Demonstrate that a network learned or used a specific property important for a task

Workflow Circuits in Transformers

From [Conmy et al. 2023](#)

Step 1: Choose a behaviour/task (>) that the model displays and create a dataset to obtain that behavior from the model

Task	Dataset Template	Ideal Output
Greater-Than	The <noun> lasted from the year XYY to the year XX??	?? To be greater-Than YY

Dataset:

Task	Dataset Example	Ideal Output
Greater-Than	"The war lasted from 1732 to 17"	"33" or "34" or ...or "99"
	"The investigation lasted from 1921 to 20"	"22" or "23" or ...or "99"

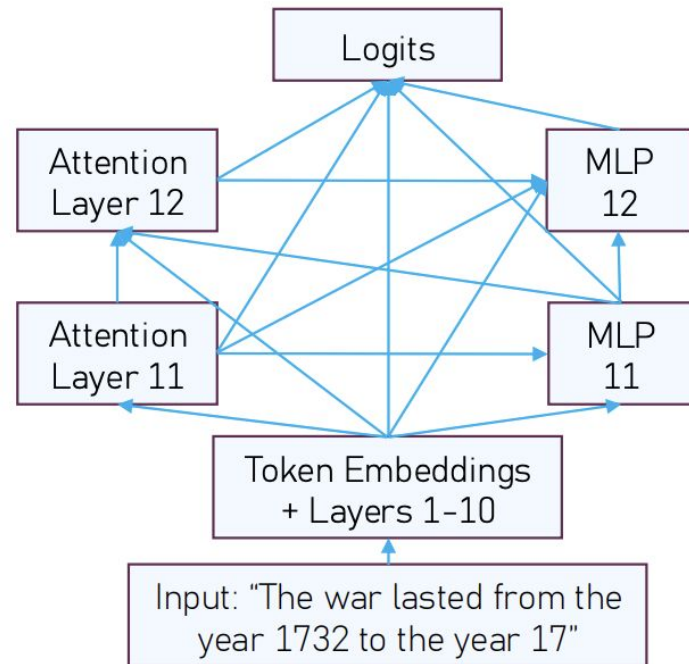
Workflow Circuits in Transformers

Step 2: Find circuits for the target behavior by representing the internals of the model as a computational directed acyclic graph (DAG)

Node connectivity needs to be faithful to the model's computations and one can also consider connections between components in non-adjacent layers due to the additivity of the residual connections

Select the level of abstraction for the computational graph depending on the desired level of detail for the explanations:

- **Coarse:** Interactions between attention heads and MLPs
- **Granular:** Interactions on the neuron level



Workflow Circuits in Transformers

Step 3: Prune the graph using recursive **activation patching**

Activation patching:

- Overwrite the activation value of a node or edge with a corrupted activation
- Do a forward pass in the network
- Compare the outputs of the original model without corruption and the corrupted model using a suitable metric
- If the change is below a prespecified threshold, remove the component

One typically starts at the output node to determine the most important incoming edges and then proceed to the parent nodes through the identified edges

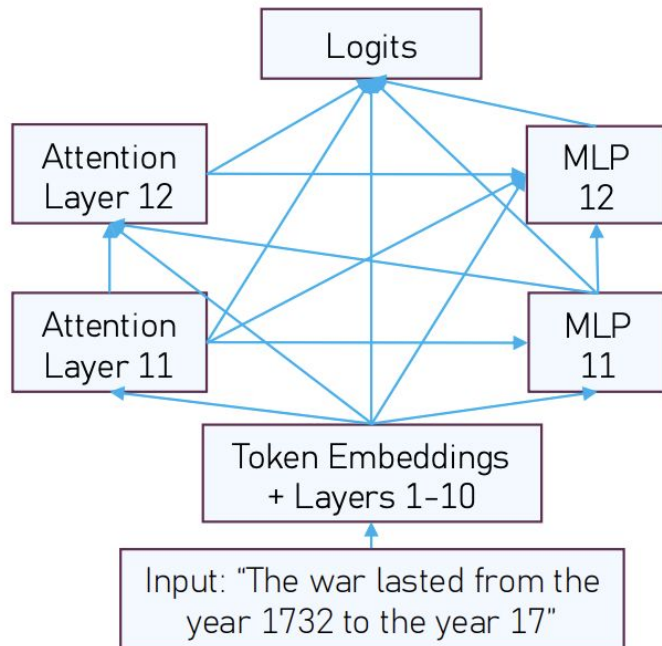
The activation can be corrupted by:

- Replacement with 0
- Replacement with a mean
- Replacement with activation of another data sample

Once you have a circuit you can formulate and test hypothesis about the functions implemented in the subgraph

Workflow Circuits in Transformers Example

Goal: Assess the importance of MLP 11 on the logit outputs of the model

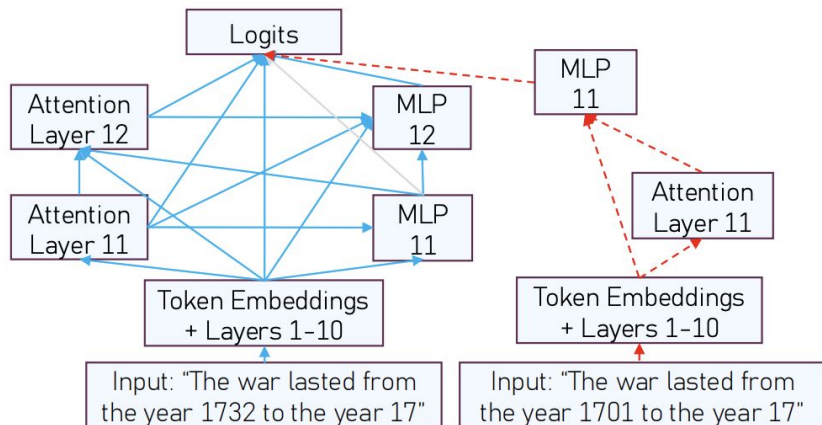


Workflow Circuits in Transformers Example

Step 1: Patch the path from MLP11 to the logits by using different inputs

Step 2: Forward pass in the model to compute the differences between the logits obtained using the corrupted outputs and the original model

Step 3: If the differences are below a prespecified threshold remove the analysed connection



Automatically Discovering Circuits (ACDC)

Algorithm 1: The ACDC algorithm.

Data: Computational graph G , dataset $(x_i)_{i=1}^n$, corrupted datapoints $(x'_i)_{i=1}^n$ and threshold $\tau > 0$.

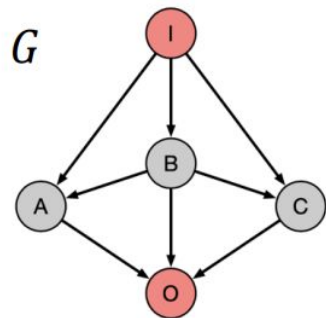
Result: Subgraph $H \subseteq G$.

```
1  $H \leftarrow G$  // Initialize H to the full computational graph
2  $H \leftarrow H.reverse\_topological\_sort()$  // Sort H so output first
3 for  $v \in H$  do
4   for  $w$  parent of  $v$  do
5      $H_{new} \leftarrow H \setminus \{w \rightarrow v\}$  // Temporarily remove candidate edge
6     if  $D_{KL}(G||H_{new}) - D_{KL}(G||H) < \tau$  then
7        $H \leftarrow H_{new}$  // Edge is unimportant, remove permanently
8     end
9   end
10 end
11 return  $H$ 
```

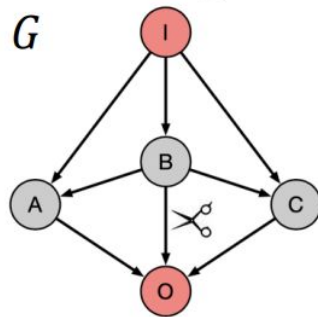
Automatically Discovering Circuits (ACDC)

$(x_i)_{i=1}^n$: original
set of prompts

$(x'_i)_{i=1}^n$: corrupted
set of prompts



$H \subseteq G$

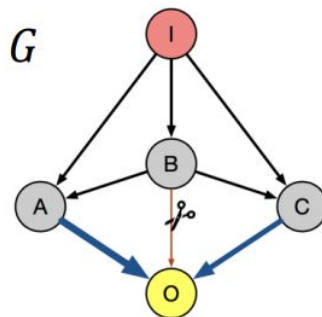


$H(x_i, x'_i)$

$D_{\text{KL}}(G(x_i) \parallel H(x_i, x'_i))$

$D_{\text{KL}}(G \parallel H)$

$H \subseteq G$



ACDC Example

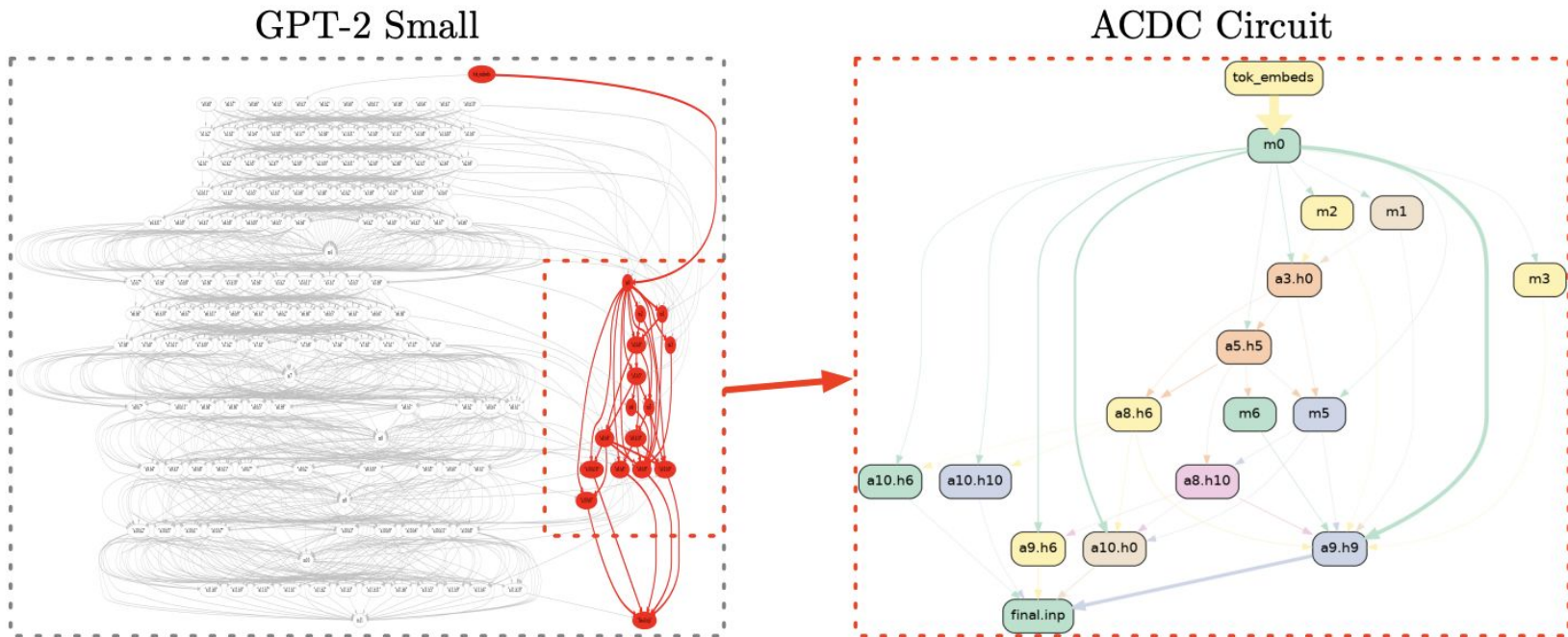


Figure 1: **Automatically discovering circuits with ACDC.** *Left:* a computational graph for GPT-2 Small, with a recovered circuit for the IOI task highlighted in red. Only edges between adjacent layers are shown. *Right:* the recovered circuit with labelled nodes. All heads recovered were identified as part of the IOI circuit by Wang et al. (2023). Edge thickness is proportional to importance.

Global Model Agnostic XAI Approaches

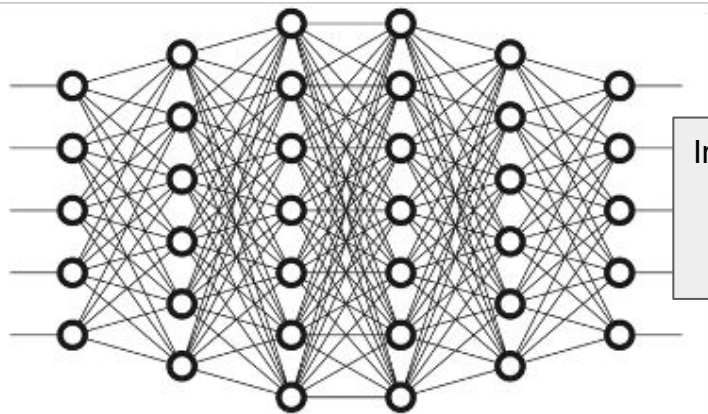
Global Post-Hoc Explainability Approaches

- Explain the **complete behavior** of a given complex ML model
- Complementary to local explanations

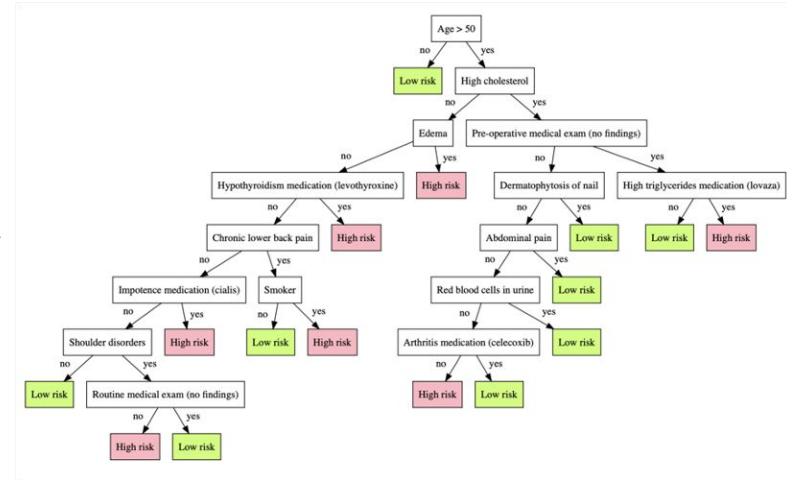
- How to generate them?
 - Model distillation/mimicking
 - Generate local explanations for every sample using some of the local approaches
 - Pick a suitable subset of k representative local explanations to create a global explanation

Model Distillation for Global Explanation Generation

- Use a simpler, inherently interpretable model (GAMs, decision trees, decision sets) trained to mimic (approximate) the predictions of the complex, back box model and then used to explain its predictions
- Model agnostic approach

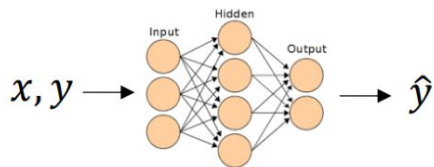


Interpretable Explainer
Model Mimicking
Predictions of
Complex Model

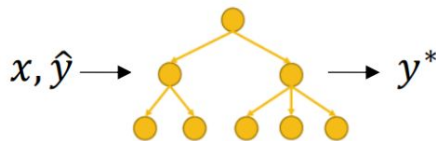


Model Distillation for Global Explanation Generation

Step 1: train a black box model on some data x with labels y



Step 2: train an interpretable model on x and \hat{y}



Step 3: check the alignment of \hat{y} and y^*



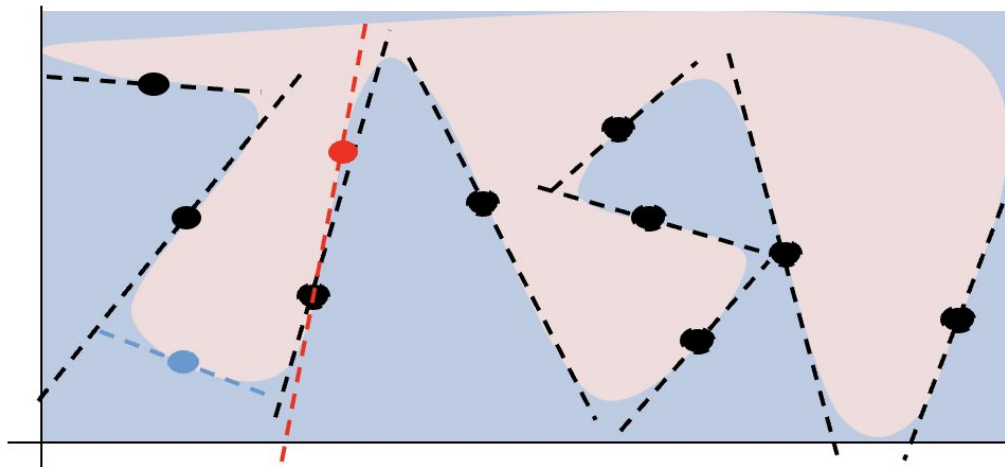
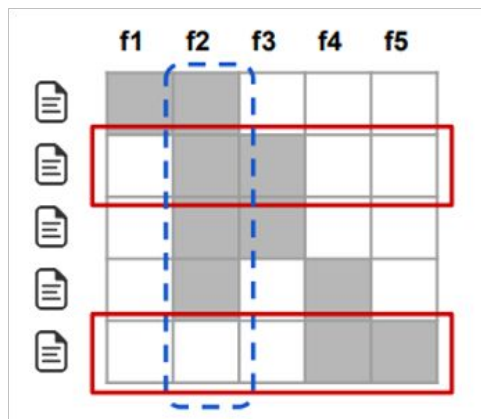
Step 4: use the well-aligned surrogate for interpreting \hat{y}

Surrogate alignment: R squared that measures the percentage of variance captured by the surrogate model with value close to one meaning the surrogate provides very good approximation and a value close to 0 quite bad one

$$R^2 = 1 - \frac{SSE}{SST} = 1 - \frac{\sum_{i=1}^n (y_i^* - \hat{y}_i)^2}{\sum_{i=1}^n (\hat{y}_i - \bar{\hat{y}})^2}$$

Global Explanations from Local Feature Importances

- Submodular Pick LIME (SP-LIME) [[Ribeiro et al 2016](#)]
- Model agnostic
- Provides global explanations for an ML model by using submodular optimization to greedily select k explanations that are:
 - Representative for the model's global behaviour
 - Diverse, no redundancy



Evaluation of Explanations

Evaluating Model Explanations

Different aspects:

- Correctness of explanations: How accurate and precise are they?
- Relevance (meaningfulness) of explanations
- Interpretability of explanations: For target users
- Actionability
- Succinctness: How concise and compact are they?
- Completeness
- Robustness/stability

No single established ways evaluating model explanations - highly dependent on the type of model explanation, task/application, user, domain

Common metrics: prediction performance improvements with explanations, decision time, user satisfaction, agreement with expert reasoning, explanation complexity, stability scores

Evaluating Interpretability of Explanations

From [[Doshi-Velez et al. 2017](#)]



Evaluating Interpretability of Explanations

Application-grounded evaluation

- Evaluate quality of explanations in the context of the exact application task (or a simpler/partial task) with actual domain experts
- Example: test whether explanations help doctors make better diagnosis decisions (metrics: diagnostic accuracy, decision time, trust ...)
- Highest realism, highest cost

Human-grounded evaluation

- Use simplified tasks with non-experts / lay users
- Measures general human understanding of explanations
- Example experiments: compare which explanation is easier to use via binary forced choice; forward simulation/prediction (humans are presented with an explanation and an input, and must correctly simulate the model's output); counterfactual simulation (humans are presented with an explanation, an input, and an output, and are asked what must be changed to change the method's prediction to a desired output)

Functionally-grounded evaluation

- Use some formal definition of interpretability as a proxy for explanation quality (quantitative metrics)
- No humans involved
- Example: sparsity, monotonicity, model size, number of rules/prototypes
- Fastest, but weakest evidence of human usefulness

Evaluation of Post Hoc Explanations

Important aspects [[Agarwal et al 2022](#)]:

- **Faithfulness (correctness)** - using ground truth or if explanations come from a simpler models (LIME), the overlap among their predictions with those of the complex model, removing important features to assess drop in performance
- **Stability** - stability of explanations for small changes in the input, account for stability of the model (predictions, intermediate representations)
- **Fairness** - compute average correctness/stability for minority and majority groups and check for differences
- **Interpretability** - domain experts

Evaluation of Post Hoc Explanations Faithfulness

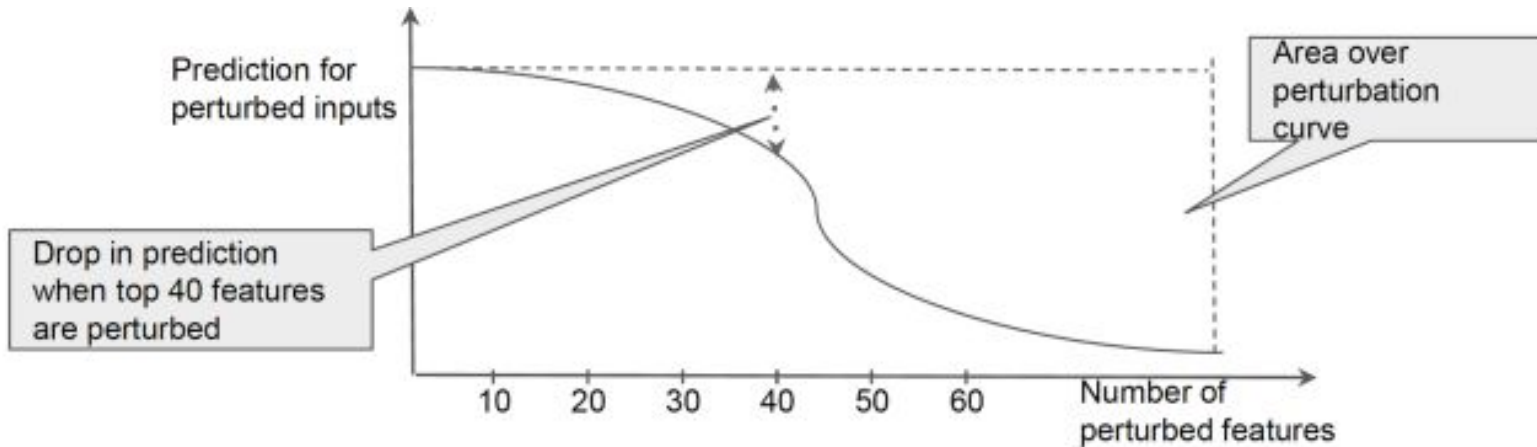
- **Ground-truth Faithfulness** (correctness) of feature attribution methods using ground truth
 - *Feature agreement*: agreement of the top k predicted and ground truth most important features: fraction of common features among the top k most important
 - *Sign Agreement*: feature agreement considering also the sign of the importance (positive or negative)
 - *Rank Agreement* of the top k predicted and ground truth features : rank agreement among the top k most important features, signed rank agreement of the top k most important features
 - *Rank Correlation*: Spearman's rank correlation among the ordered predicted and ground truth most important features
 - *Pairwise Rank Agreement*: captures if the relative ordering of every pair of features is the same for a given post hoc explanation as well as the corresponding ground-truth explanation

Evaluation of Post Hoc Explanations Faithfulness

What if there is no ground truth available?

Predictive Faithfulness of feature attributions via *perturbation*: Prediction Gap on Important feature perturbation [[Samek et al. 2015](#)]

- Computes the difference in prediction probability that results from perturbing the features deemed as influential by a given post hoc explanation
- Perturbation example: replace the feature with a random value
- Metric: Area Over Perturbation Curve (AOPC)



Evaluation of Post Hoc Explanations Stability

Metrics introduced by [[Alvarez-Melis et al. 2018](#); [Agarwal et al. 2022](#)]

- *Relative Input Stability* (RIS): measure the maximum change in explanation relative to (small) changes in the inputs
- *Relative Representation Stability* (RRS): measure the maximum change in explanation relative to (small) changes in the model parameters (intermediate representations of the ML model)
- *Relative Output Stability* (ROS): measure the maximum change in explanation relative to (small) changes in the output prediction probabilities

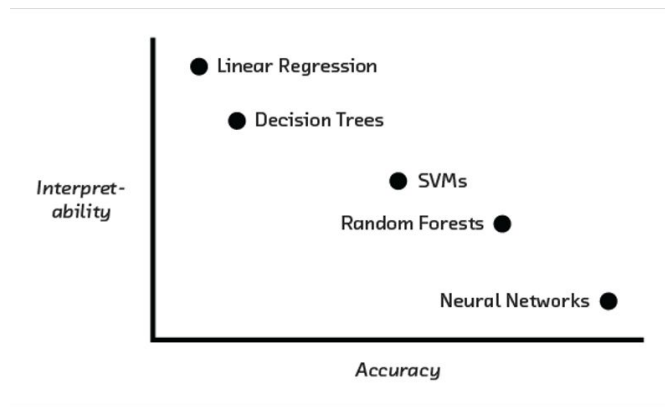
Evaluation of Post Hoc Explanations Fairness

Approach from [[Dai et al. 2018](#)]

- Average all relevant metrics (faithfulness, stability) across instances in majority and minority subgroups (like, race A vs race B, different genders)
- Compare the estimates from the two target groups and if the difference between the two means is statistically significant, there is evidence of unfairness

Summary

- XAI is crucial in AI employed in complex high-stakes decision making settings (healthcare, legal, finance)
- Often trade-offs predictive performance vs. interpretability exist



- If you can use inherently interpretable model for your application **JUST DO IT**
- If not, either use a combination of some of the post hoc explanation approaches, adjust your model architecture drawing inspiration from the in-model approaches, look into mechanistic explainability
- Needs approach-, application-specific and user-specific evaluation

Many Open Challenges

- Faithfulness of explanations
- Human interpretability
- Robustness of explanations
- (Dis)Agreement and effective combination of explanations from various approaches
- Evaluation
- Explainability for large models
- Explainability for multimodal models
- Explainability for complex temporal data

Open Source Tools

- Interpretable models:
<https://github.com/interpretml/interpret>
- Captum (XAI for PyTorch):
<https://captum.ai/>
- OpenXAI (explainers and evaluation):
<https://open-xai.github.io/>
- Quantus (evaluation):
<https://github.com/understandable-machine-intelligence-lab/Quantus>

Practical Part

Explainable AI:

[LIME/SHAP Tutorial](#)



[Saliency Tutorial](#)



Credits

Inspiration and parts of content for this lecture was taken from tutorials/lectures of Hima Lakkaraju and Mateja Jamnik. Thank you!

Practical part co-created with Leonida Lumburovska