Explainable AI

Winter School on
Deep Learning For Vision and Language Modelling
Indian Institute of Technology Guwahati
6th - 13th January, 2025

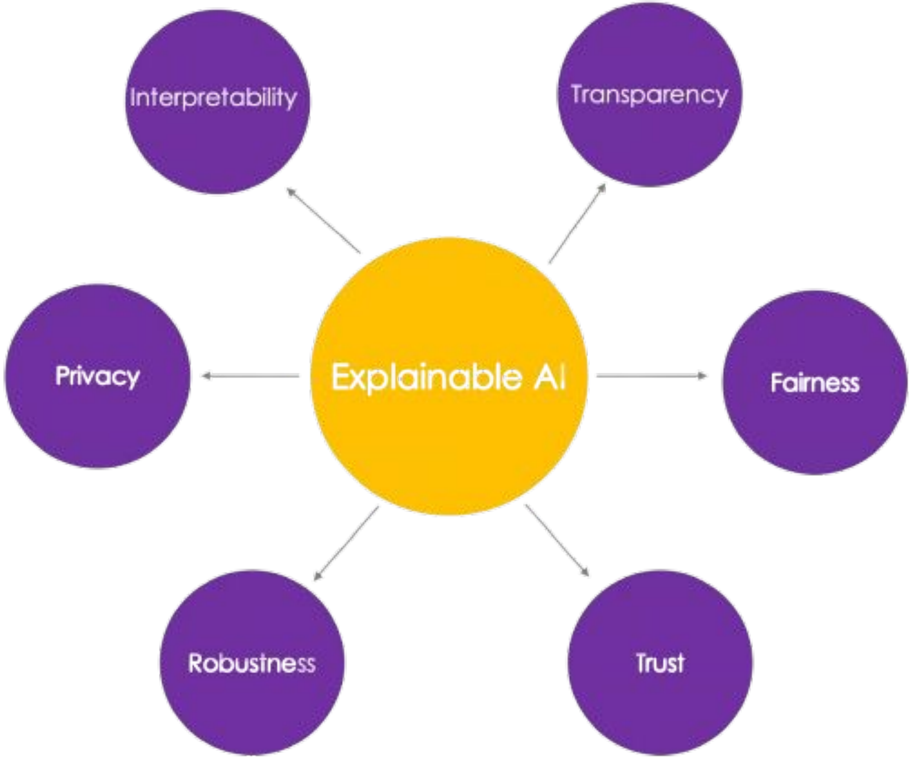
Challenges in Black-box AI

- *If a model fails due to a hidden bias, who is responsible — developers or the model itself?*

- *Would you trust an AI system to make critical decisions without understanding how it works?*

- *Can interpretability compromise AI's performance, or can they coexist?*

What is Explainable AI (XAI)?



Interpretability:
What does the model learn, and how can we understand its decision-making process?

Transparency:
Does the model reveal its internal workings clearly, or is it a black box?

Fairness:
Are the model's predictions free from bias and equitable across all demographic groups?

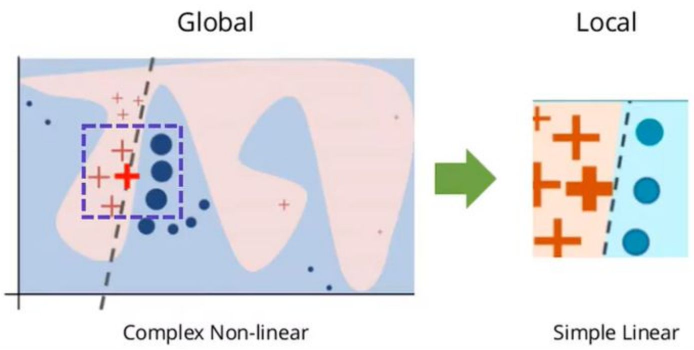
Trust:
Can users rely on the AI's outputs, and how does understanding the model foster confidence in its decisions?

Robustness:
How well does the model maintain performance and explainability under adversarial attacks or input perturbations?

Privacy:
Does the model protect sensitive information and ensure explainability without exposing private data?

Techniques of Explainable AI

- **Local vs. Global**
- **Model-Specific vs. Model-Agnostic**



Types of Explanations:

- **Feature Importance:** Identifying which input features influenced the decision most.
- **Counterfactuals:** Showing how small changes in the input could alter the output.
- **Visualizations:** Representing activations, heatmaps, or decision boundaries for interpretability.

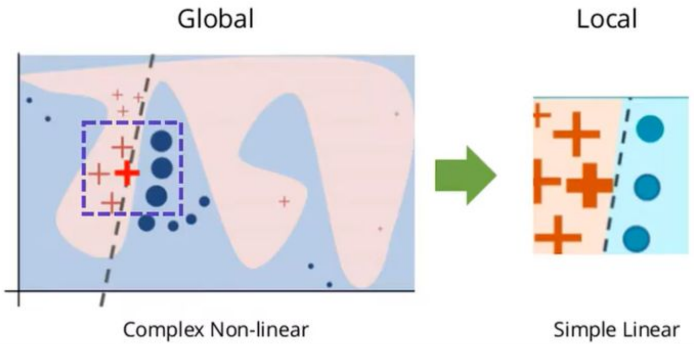
XAI Techniques

Algorithm	Type	Description
LIME (Local Interpretable Model-agnostic Explanations)	Model-Agnostic	Approximates local behavior of any model with an interpretable surrogate model.
SHAP (SHapley Additive ExPlanations)	Model-Agnostic	Uses Shapley values to explain predictions for any type of model.
Grad-CAM (Gradient-weighted Class Activation Mapping)	Model-Specific (CNN)	Visualizes important regions for predictions by analyzing CNN

LIME

Work Principle:

- Input instance: x_0
- Perturbations: Generate $Z=\{ x_1, x_2, \ldots, x_n \}$ by sampling around x_0
- Black-box model predictions: $f(Z)=\{ y_1, y_2, \ldots, y_n \}$



- Define locality weights:

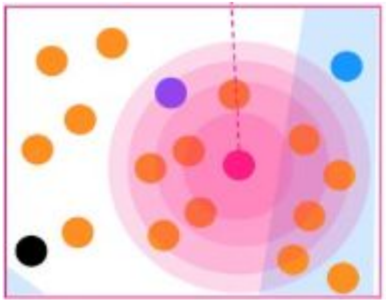
$$w(x, x_0) = \exp\left(-\frac{\|x - x_0\|^2}{2\sigma^2}\right)$$

where σ controls the width of the neighborhood.

- Surrogate model: Fit a simple model $g(x)$ (e.g., linear regression) to minimize:

$$L(g, f, w) = \sum_{x \in Z} w(x, x_0) \cdot (f(x) - g(x))^2$$

subject to g being interpretable.



Output:

Coefficients of $g(x)$ are used as feature attributions for the prediction of x_0

A Case-Study : LIME on 'UCI adult income' dataset with logistic regression Classifier

https://drive.google.com/file/d/1A4fsMSy9miM5W9Q_Gu9z1E-1LkBUHbt_/view?usp=sharing

	age	workclass	fnlwgt	education	educational-num	marital-status	occupation	relationship	race	gender	capital-gain	capital-loss	hours-per-week	native-country	income
0	25	Private	226802	11th	7	Never-married	Machine-op-inspct	Own-child	Black	Male	0	0	40	United-States	<=50K
1	38	Private	89814	HS-grad	9	Married-civ-spouse	Farming-fishing	Husband	White	Male	0	0	50	United-States	<=50K
2	28	Local-gov	336951	Assoc-acdm	12	Married-civ-spouse	Protective-serv	Husband	White	Male	0	0	40	United-States	>50K
3	44	Private	160323	Some-college	10	Married-civ-spouse	Machine-op-inspct	Husband	Black	Male	7688	0	40	United-States	>50K
4	18	?	103497	Some-college	10	Never-married	?	Own-child	White	Female	0	0	30	United-States	<=50K

GRAD-CAM

Work Principle:

- **Input image**: $\boldsymbol{I}$
- **Feature map** from the final convolutional layer: $\boldsymbol{A^k} \in \boldsymbol{R^{H \times W \times D A}}$, where $\boldsymbol{H}$ is height, $\boldsymbol{W}$ is width, and $\boldsymbol{D}$ is depth (number of filters).
- **Class score**: $\boldsymbol{y_c}$ for class $\boldsymbol{c}$ from the final softmax layer.
- **Gradient of the class score** with respect to the feature map activations at the last convolutional layer: $\frac{\partial y_c}{\partial A^k}$

  This gradient measures how much the class score $\boldsymbol{y_c}$ changes with respect to the activations in each feature map.
- **Global average pooling**: Compute the average gradient across spatial dimensions (height and width):

$$\alpha_k^c = \frac{1}{H \times W} \sum_{i,j} \frac{\partial y_c}{\partial A_{ij}^k}$$

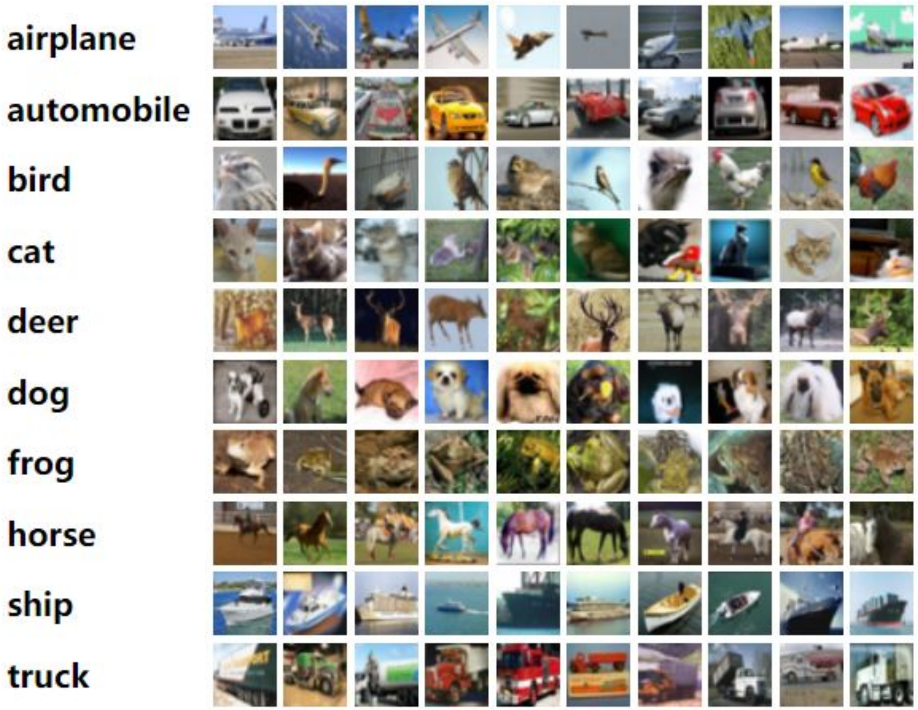
  This is a scalar weight for each filter $\boldsymbol{k}$.
- **Relu over Weighted sum of feature maps**:

$$\text{Grad-CAM}^c(\mathbf{I}) = \text{ReLU}\left(\sum_k \alpha_k^c A^k\right)$$

  The sum of the weighted feature maps gives the class activation map for class $\boldsymbol{c}$. The ReLU operation ensures that only positive activations are considered, focusing on the regions that contribute to the class prediction.

A Case-Study : GRAD-CAM on ‘cifar-10’ dataset with CNN

https://colab.research.google.com/drive/1XizujmYpMoiJRCW87geIVAYOJCJ7qfsl?usp=sharing



A Case-Study : Improving classifier's performance using discriminator with GRAD-CAM based loss

GRAD-CAM Heatmap Generation:

- Generate heatmaps of the classifier predictions
- Normalize the heatmaps

Discriminator Role:

- A lightweight neural network trained to classify heatmaps as **correct or incorrect (if classified correctly or incorrectly)**.
- Provides feedback to guide the classifier via discriminator loss.

Combined Loss Function:

- Total Loss = Classifier Loss + λ * Discriminator Loss.

https://colab.research.google.com/drive/1XizujmYpMoiJRCW87geIVAYOJCJ7qfsl?usp=sharing