Advanced Topic in Deep Learning, Assignment 1

Amir El-Ghoussani, M.Sc.
amir.el-ghoussani@fau.de
Chair of Multimedia Communications and Signal Processing

Friedrich-Alexander-Universität
Technische Fakultät

Agenda

1. Background knowledge
2. Coding task 1: LIME
3. Coding task 2: SHAP

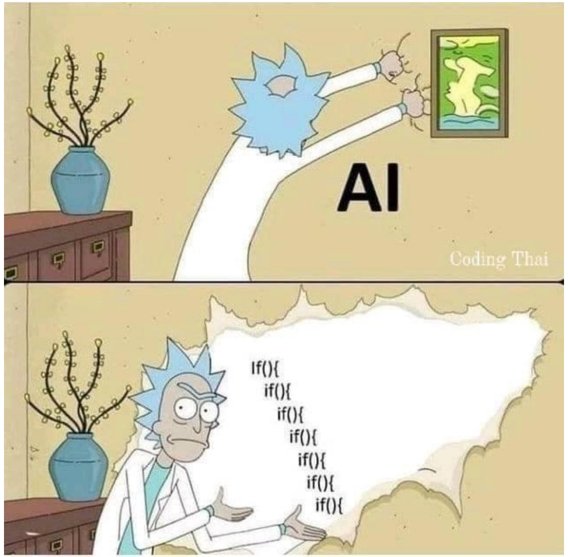
Exercise plan

- Will be updated regularly
- If assignment not uploaded in time,
  Please remind me
- If presentation is scheduled then tutorial will
  be held immediately after the pres-
  entation

Advanced Topics in Deep Learning	Day	Date	Time	Room	Exercise topic	Submissions
~~Holiday / Teaching Free~~	~~Monday~~	~~4/21/2025~~	~~12:00~~	~~(05.025 Seminarraum)~~		
Lecture	Wednesday	4/23/2025	8:00	H15		
Exercise	Monday	4/28/2025	12:00	(05.025 Seminarraum)	Intro	Ex. 1 Upload
Exercise	Wednesday	4/30/2025	8:00	H15	Ex. 1 presentation	
Exercise	Monday	5/5/2025	12:00	(05.025 Seminarraum)	Tutorial	
Lecture	Wednesday	5/7/2025	8:00	H15		
Lecture	Monday	5/12/2025	12:00	(05.025 Seminarraum)		Ex. 1 Deadline
Lecture	Wednesday	5/14/2025	8:00	H15		
Lecture	Monday	5/19/2025	12:00	(05.025 Seminarraum)		
Lecture	Wednesday	5/21/2025	8:00	H15		Ex. 2 Upload
Exercise	Monday	5/26/2025	12:00	(05.025 Seminarraum)	Ex. 2 presentation	
Lecture	Wednesday	5/28/2025	8:00	H15		
Exercise	Monday	6/2/2025	12:00	(05.025 Seminarraum)	Tutorial	
Lecture	Wednesday	6/4/2025	8:00	H15		
~~Holiday / Teaching Free~~	~~Monday~~	~~6/9/2025~~	~~12:00~~	~~(05.025 Seminarraum)~~		
Lecture	Wednesday	6/11/2025	8:00	H15		Ex. 2 Deadline
Lecture	Monday	6/16/2025	12:00	(05.025 Seminarraum)		Ex. 3 Upload
Exercise	Wednesday	6/18/2025	8:00	H15	Ex. 3 presentation	
Exercise	Monday	6/23/2025	12:00	(05.025 Seminarraum)	Tutorial	
Lecture	Wednesday	25.06.2025	8:00	H15		Ex. 4 Upload
Exercise	Monday	6/30/2025	12:00	(05.025 Seminarraum)	Ex. 4 presentation	Ex. 3 Deadline
Lecture	Wednesday	7/2/2025	8:00	H15		
Exercise	Monday	7/7/2025	12:00	(05.025 Seminarraum)	Tutorial	
Lecture	Wednesday	7/9/2025	8:00	H15		Ex. 5 Upload
Exercise	Monday	7/14/2025	12:00	(05.025 Seminarraum)	Ex. 5 presentation	Ex. 4 Deadline
Lecture	Wednesday	7/16/2025	8:00	H15		
Exercise	Monday	7/21/2025	12:00	(05.025 Seminarraum)	Tutorial	
Exercise	Wednesday	7/23/2025	8:00	H15	Exam Q&A	Ex. 5 Deadline

Background knowledge

- In the first lecture, you covered the topic interpretability in supervised learning. Interpretable machine learning model provides information on why certain decisions have been made.



Background knowledge

- In the first lecture, you covered the topic interpretability in supervised learning. Interpretable machine learning model provides information on why certain decisions have been made.
- Categories of Interpretability Methods

Methods	Model training		Class of models		Predictions	
	Pre-hoc / intrinsic	Post-hoc	Model-specific	Model-agnostic	Local	Global
Properties	Simple structure	Trained model	Particular class of models (NN)	Any (trained) model	Individual prediction	Entire model

Background knowledge

Interpretable Models:

- Linear Regression, Logistic Regression and Decision Trees
- Local Model-Agnostic Approaches:
  - Individual Conditional Expectation (ICE): a plot shows how the output changes when changing a feature
  - *Local interpretable model-agnostic explanations (LIME):* explain individual predictions of black box machine learning models.

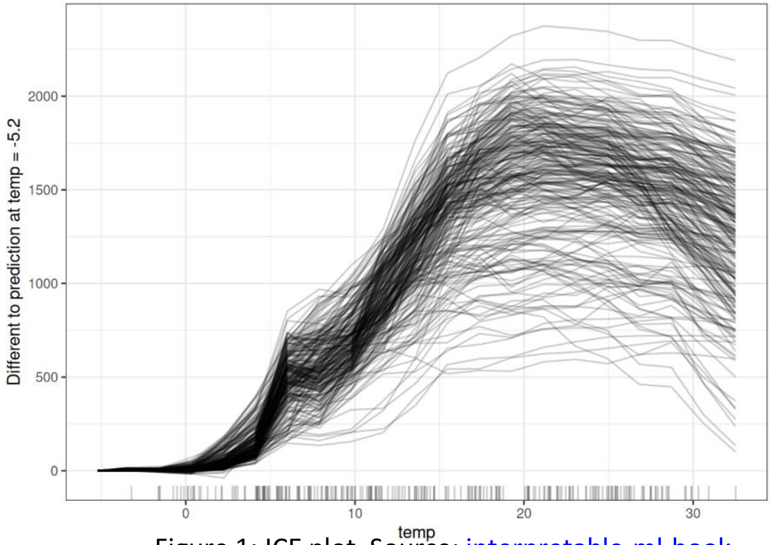


Figure 1: ICE plot. Source: interpretable-ml-book

Background knowledge

Interpretable Models:

- Linear Regression, Logistic Regression and Decision Trees
- Local Model-Agnostic Approaches (cont.):
  - Counterfactual Explanations: smallest change to the feature values that changes the prediction to a predefined output
  - Shapley value: assume that each feature value of the instance is a "player" in a game where the prediction is the payout → fairly distribute the "payout" among the features
  - *SHAP:* explain individual predictions, based-on the game-theoretically optimal Shapley values

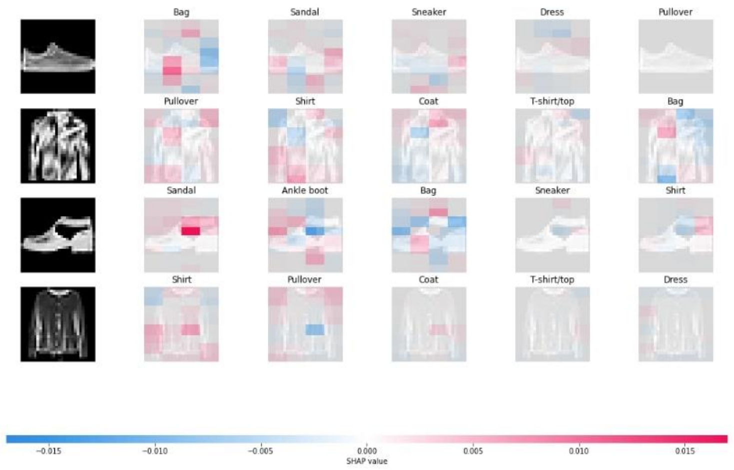
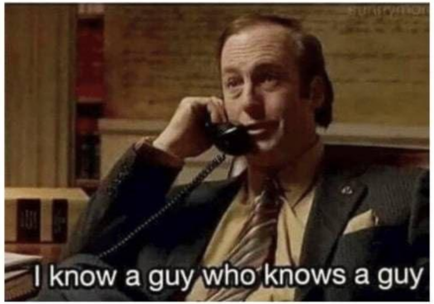


Image source: shap-for-image-classification

Coding task 1: **L**ocal **I**nterpretable **M**odel-agnostic **E**xplanations

How Neural Networks work?
Neurons:

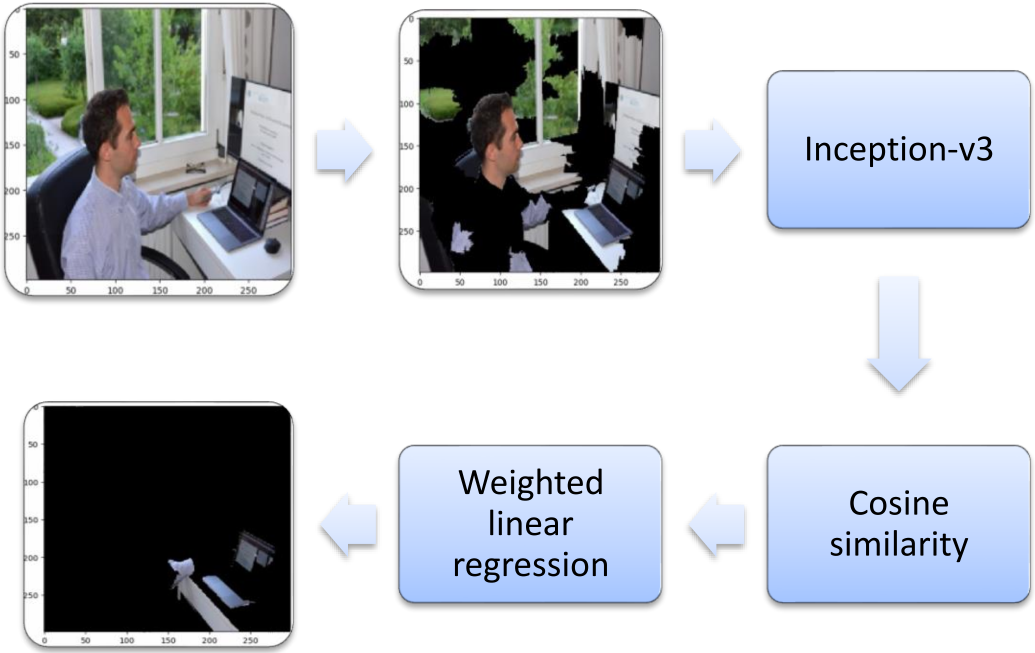


- LIME focuses on explaining the model's prediction for individual instances.
- Use **surrogate** model – a simple interpretable model.
- Variations of the images are created by segmenting the image into "super pixels" and turning super pixels off or on.
- LIME is one of the few methods that works for tabular data, text, and images.
- LIME is implemented in Python and R and is easy to use. To fully understand the algorithm, you are NOT allowed to use the library.
- Problems of LIME: instability of the explanations → difficult to trust the explanations

Reference: interpretable-ml-book

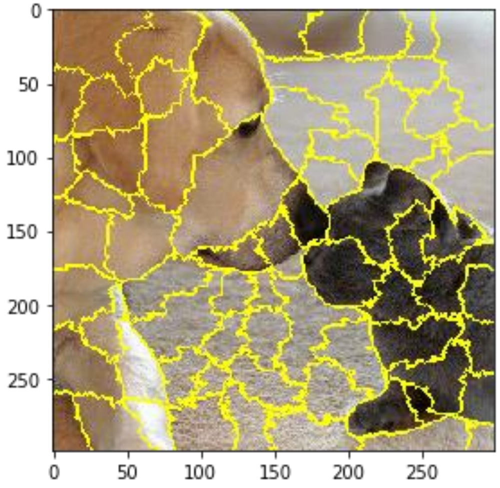
Coding task 1: LIME

- Pipeline of LIME
  1. Pick a sample + a black box ML model
  2. Perturb sample multiple times → new dataset
  3. Get model predictions for each perturbation
  4. Weight the prediction: distance to the original image
  5. Train an interpretable model on perturbed data
  6. Get explanation



Coding task 1: LIME – super pixels

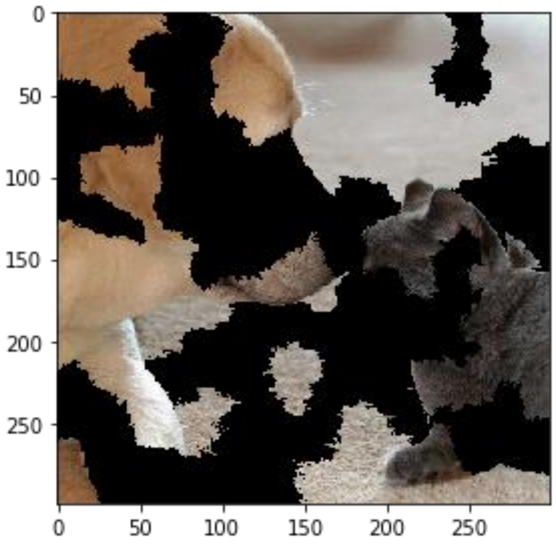
- Compute super pixels:
  - Super pixels are contiguous patches of the image that share color and / or brightness similarities
  - Implementation: use **quickshift()** from skimage
  - Why quickshift? → quickshift is a mode-seeking algorithm that considers the pixels as samples over a 5-dimensional space (3 color dimensions and 2 space dimensions)
  - Some inputs of quickshift()
    - Image: input image of shape (M, N, C)
    - Ratio: between color-space and image-space proximity. Higher → more weight for color-space
    - Kernel_size: size of Gaussian kernel for sample density smoothing. Higher → fewer clusters
    - Max_dist: Cut-off point for data distances. Higher means fewer clusters



Sample output

Coding task 1: LIME – perturbation

- Perturbs the image
  - Randomly replacing some super pixels of the image
  - Perform using mean replacement: mean color of the super pixels
  - Usually, black out the super pixels
  - Repeat this step multiple times → new dataset
  - Implementation hints: use binomial distribution to select which super pixels to be turned off



Sample output

Coding task 1: LIME – weights

- Calculate weights:
  - Compute the cosine distance between original image and the perturbed image using pairwise_distances from sklearn.metrics
  - Compute weight by a kernel function:

$$\pi_i = \sqrt{\exp\left(-\frac{\mathrm{d}_{\cos}^2}{\mathrm{v}^2}\right)}$$

Where $v > 0$ is a bandwidth parameters, default = 0.25.

Coding task 1: LIME – surrogate model

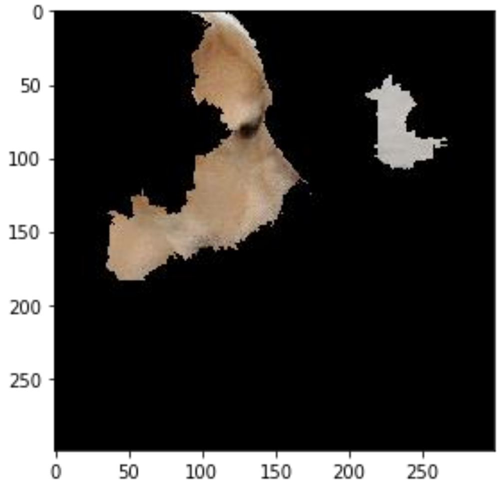
- Surrogate model:
  - Weighted linear model: [LinearRegression()]
  - Remember to set the weight when using [fit()] using the previously computed weights
  - Input data is the perturbed images
  - Each coefficients in the linear model corresponds to one super pixel in the segmented image
    → the importance of each super pixel for the prediction of the top class

```
array([ 0.0199833 , -0.01601374,  0.10354327, -0.04821644,  0.08925877,
        0.07826848,  0.02714029,  0.07659395,  0.18122355, -0.05638588,
        0.03509676,  0.00470357,  0.02208912,  0.10356663,  0.07223697,
        0.0034734 ,  0.08162887,  0.03907232,  0.00769051,  0.02527205,
       -0.0100494 ,  0.02130284, -0.07029254, -0.02555164,  0.52121138,
        0.0205534 ,  0.0013183 , -0.17025011, -0.03082538,  0.14881233,
        0.05691062,  0.1011255 , -0.01224566, -0.04081408, -0.03864275,
       -0.02153394, -0.05745923,  0.02746975,  0.03796638,  0.03152467,
        0.03358099,  0.00733296,  0.04806797, -0.02303122, -0.0145786 ,
        0.08431814,  0.008036  , -0.01945883, -0.09000518,  0.05641921,
        0.02874261,  0.01926118, -0.03653446,  0.03901715, -0.05825456,
        0.03474161, -0.102688  ,  0.00780907, -0.03470868,  0.03349195,
        0.06900843, -0.05142001,  0.02219387,  0.05436448,  0.01072274,
       -0.03208548,  0.09252425, -0.0057378 ])
```

Sample output

Coding task 1: LIME – explanation

- Explain the results of black box model
  - Sort the coefficient from the surrogate model → get the super pixel that contribute the most to the black box output
  - Visualize
- In the example, Labrador Retriever is the class with highest confidence and this is its LIME interpretation



Sample output

Coding task 2: SHAP

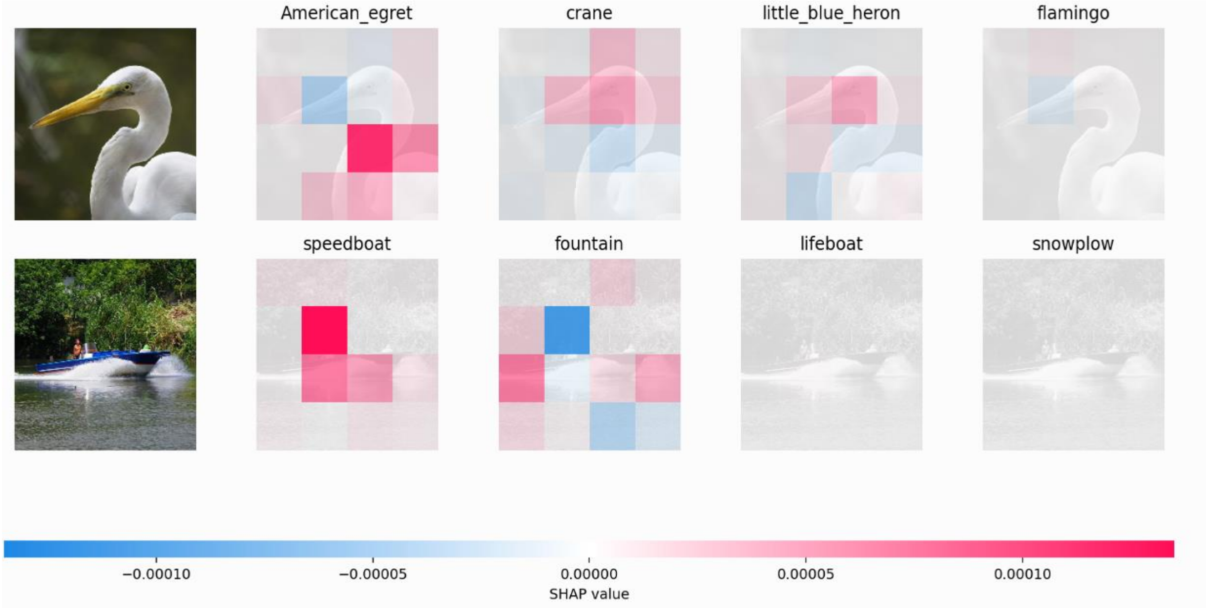
- A method to explain individual predictions, based on Shapley Values. SHAP brought Shapley values to <u>text and image</u> models.
- Combination of LIME and Shapley Values.
- Satisfies properties of Efficiency, Symmetry and Additivity.
- Strengths:
  - Solid theoretical foundation
  - Connects LIME and Shapley values
  - Fast implementation for tree-based models
  - Global interpretations are consistent with the local explanations
- Limitations:
  - Possible to create intentionally misleading interpretations
  - Can be slow

Coding task 2: SHAP

- SHAP value explanation after 100 evaluations:
  Red: indicates positive influence, Blue: indicates negative influence

False positive

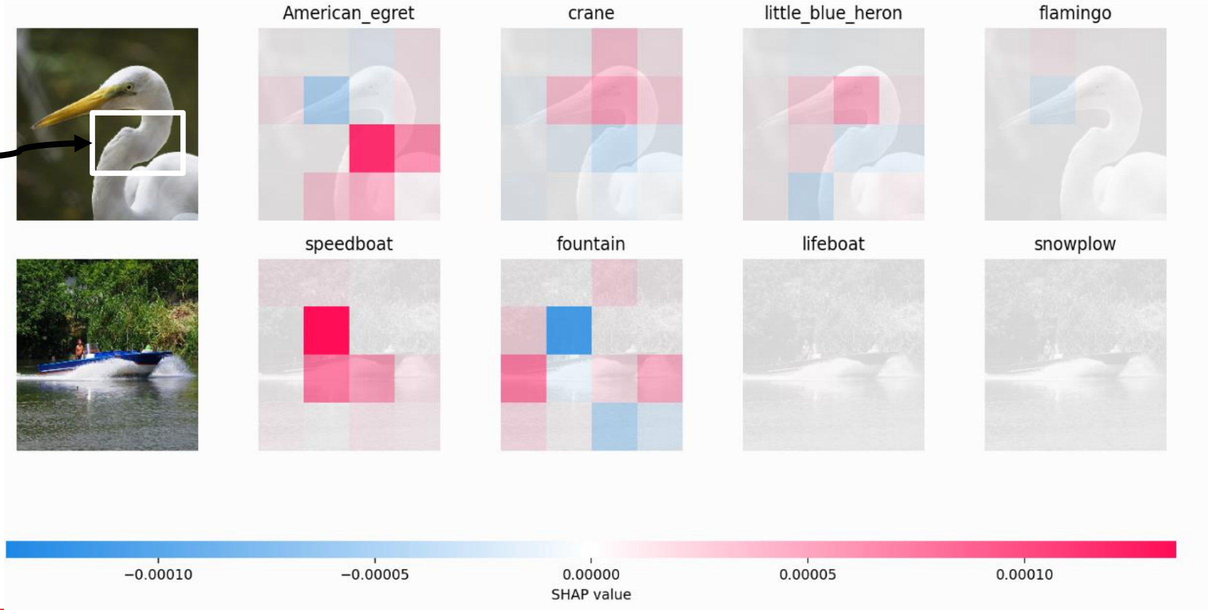
True positive



Coding task 2: SHAP

- SHAP value explanation after 100 evaluations:
  Red: indicates positive influence, Blue: indicates negative influence

‘bump’ causes misclassification



Questions?