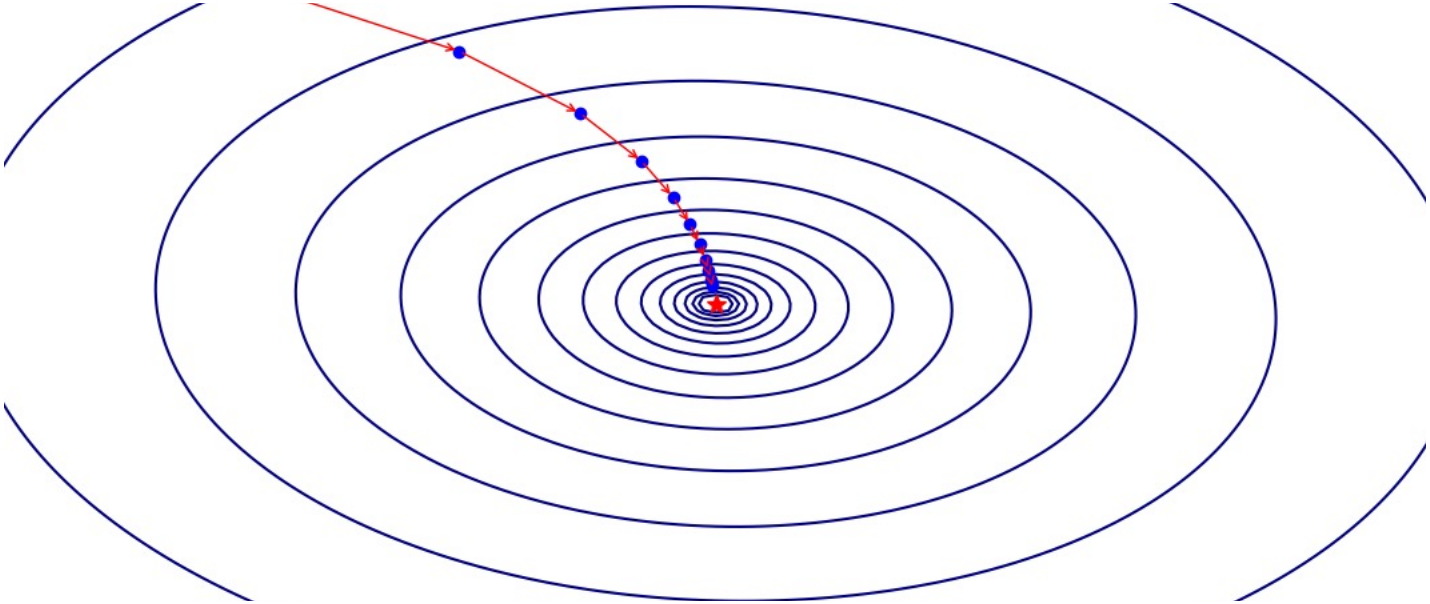


Advanced Topics in Deep Learning

Summer Semester 2024

1. Interpretability

15.04.2024

Prof. Dr. Vasileios Belagiannis

Chair of Multimedia Communications and Signal Processing

Friedrich-Alexander-Universität
Technische Fakultät

Team

- Lectures: Vasileios Belagiannis.
- Exercises: Amir El-Ghoussani.
- Special thanks to Arij Bouazizi, Adrian Holzbock, Julia Hornauer, Julian Wiederer, Youssef Dawoud, Amir El-Ghoussani for the support in creating the lectures material.

About the instructor

Not for sharing (LMS, Friedrich-Alexander-Universität Erlangen-Nürnberg)

Studies

- Democritus University of Thrace (Dipl. Eng.), 2004 – 2009
- Technische Universität München (M.Sc.), 2009 – 2011
- Technische Universität München (Dr. rer. nat.), 2011 – 2015

Experience

- VGG, University of Oxford (Post-doc), 2015 – 2017
- Vision, OSRAM (Senior Researcher / Post-doc), 2017– 2018
- Universität Ulm (Assistant Professor), 2018 – 2022
- Universität Magdeburg (Professor), 2022
- FAU (Professor), 2022 -

Machine Learning and Perception Group @ FAU

- Chair of Multimedia Communications and Signal Processing.

Course Objectives

- Study advanced topics in deep learning.
- Develop algorithms related to the course topics.
- Read publications related to the course topics.

Prerequisites

- There are no formal prerequisites to join the lecture.
  - Check if the lecture is offered to your study programme or if you are allowed to choose it.
- Basics topics will not presented here. One needs to have the background from *Introduction to Deep Learning* and *Machine Learning in Signal Processing*.
- It is not recommended to follow the lecture without background on machine learning and deep learning.
- Being familiar with Python is important for the exercises.
- There will be literature provided at the end of each lecture.

Logistics & Grading

- Lectures: Mondays at 12:15
  - Room: 11501.00.155 (**H15 Hans-Wilhelm-Schüßler-Hörsaal**).
- Exercises: Wednesdays at 16:15.
  - Room: 11501.05.025 (**05.025 Seminarraum**).
- Wednesday **17.04 & 24.04** there will be additional lectures online in Zoom (no exercise).
- The slides and other course material are available on Studon.
- There is a bonus 0.3 if one submits all assignments.
- The final exam will be written.

Course Topics

1. **Interpretability.**
2. Attention and Transformers.
3. Geometric Deep Learning.
4. Similarity Learning.
5. Generative Models.
6. Self-supervised Learning.
7. Model Compression.
8. Transfer learning, domain adaptation, few-shot learning.
9. Uncertainty Estimation.
10. Recap and Q&A.

- The exam will be written.
- We will have an exam preparation test.

Acknowledgements

- Special thanks Arij Bouazizi, Julia Hornauer, Julian Wiederer, Adrian Holzbock and Youssef Dawoud for contributing to the lecture preparation.

Today's Agenda and Objectives

- Interpretability definition.
- Categories of interpretability.
- Interpretable Models.
- Local Model-Agnostic Approaches.
- Local interpretable model-agnostic explanations (LIME).

Machine Learning in Everyday Life

- Example: Build a machine learning algorithm for loan eligibility prediction.
- Input features: age, income, loan amount, credit history, loan usage, …
- Algorithm challenges:
  - Fairness and ethics in the way the decision is made.
  - Privacy preserving algorithm since it deals with personal data.
  - Robust algorithm where small changes in the features do not alter the decision.
  - Provide explanations for the decision to trust the algorithm.

Source: https://www.flaticon.com Freepik

Goal: build a house.

Trustworthy Machine Learning

- Machine learning systems are increasingly being used in real-world applications.
- Our reference machine learning systems is composed of a deep neural network.
- Can we trust the output of the system?
- Can we interpret how the output decision was made?
- Why is interpretability important?

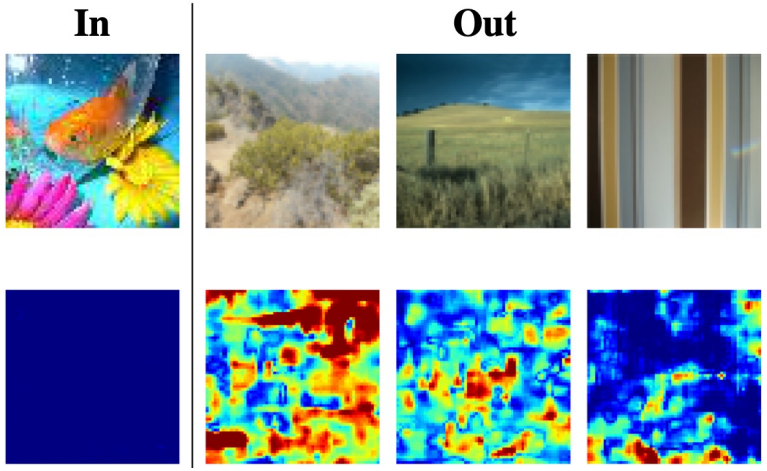


Image Source: https://arxiv.org/abs/2211.08115

Goal: Detect image regions* that the model can be uncertain about its decisions.

*Hornauer, Julia, and Vasileios Belagiannis. "Heatmap-based Out-of-Distribution Detection." Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. 2023.

Interpretability

- Interpretability refers to the degree to which we can understand the cause of a decision (as referred by Miller*, Biran and Cotton**).
- A high interpretable machine learning model provides information on why certain decisions have been made.
- In particular, it is important to know why the model outputs the way it does for high-risk applications. We week for explanations.
- Our definition of interpretability is in the context of supervised learning.

Source: https://www.flaticon.com Freepik

Automated Driving

Source: https://www.flaticon.com orvipixel

Medical Image Analysis

*Miller, Tim. "Explanation in artificial intelligence: Insights from the social sciences." Artificial intelligence 267 (2019): 1-38.
**Biran, Or, and Courtenay Cotton. "Explanation and justification in machine learning: A survey." IJCAI-17 workshop on explainable AI (XAI). Vol. 8. No. 1. 2017.

Categories of Interpretability Methods

- Categories based on the model training:
  - Pre-hoc / Intrinsic: design a machine learning model that is interpretable because of its simple structure, e.g. a decision tree or a linear regression model.
  - Post-hoc: Interpretability can be performed to an already trained machine learning model, e.g. a deep neural network.
- Categories based on the class of models:
  - Model-specific: the approach is applicable to a particular class of models, e.g. neural networks.
  - Model-agnostic: the approach is applicable to any machine learning model. This type of methods are normally applied on already trained models. Furthermore, there is no direct access to the model internals, e.g. neural network parameters. Instead there is access to the input and output of the model.
- Categories based on the prediction:
  - Local: explain individual predictions.
  - Global: explain the entire model.
- The above categories are based on Chapter 3.2 Taxonomy of Interpretability Methods*.

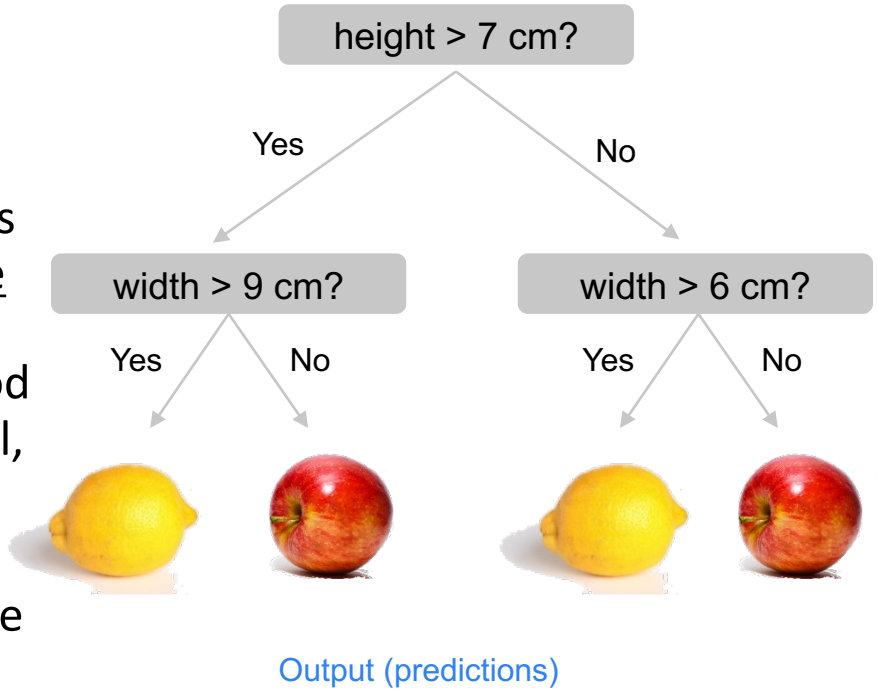
*3.2 Taxonomy of Interpretability Methods, Molnar, Christoph. Interpretable machine learning. Lulu. com, 2020.

Output of Interpretability Methods

- Developing an interpretability algorithm assumes to define the input and output to the approach.
- The following can serve as input to the interpretability method:
  - Model features.
  - Model parameters.
  - Data samples.
- The following can form the output of the interpretability method:
  - Feature summary statistics and visualisation, e.g., t-SNE on the features.
  - Model parameter analysis and visualisation, e.g., kernel of neural networks.
  - Data samples from the training set or newly created that provide explanation, e.g. adversarial samples.

Aspects of Interpretability Methods

- Global Interpretability.
  - Complete model: Understand how the model works at once*. It is assumed to have a trained model, access to the machine learning algorithm and the training data. Modern models are complex and this kind of interpretability is not feasible in practice. For example, a deep neural network cannot be understood as a whole given a pre-trained model, the training algorithm and the data.
  - Model parts (modules): Understanding parts / modules of the model in a global basis is easier. For instance, consider a decision tree where the goal is to understand a particular splitting function.



Aspects of Interpretability Methods (Cont.)

- Local Interpretability.
  - Single prediction: Examine the behaviour of the model for a single sample. For a given input sample, we examine the output. For the sample under investigation, we can, for example, perturb a feature and observe the prediction. In this way, we can understand the behaviour of the features based on the predictions.
  - Batch of predictions: The idea of the single prediction can be applied for a group of samples. In addition, we can consider different batches of predictions to generalise in a global interpretability level.
- Among the interpretability methods, sample-based (local) explanations are the most common.
- Moreover, perturbation-based methods are popular because they can be model agnostic.

Explanation Properties

- Based on the perturbation approaches*, we can define the following properties for the explanations, mainly focused on classification tasks**:
  - Expressive Power: It refers to the explanations language. For instance, the language can be propositional calculus (e.g. if, or, and), Non-classical logic (e.g. many-valued logic such as fuzzy logic), decision tree, linear model.
  - Translucency: The extent to which the explanation method has access to the machine learning model internals. It can be either decompositional (decompose the model modules) , pedagogical (treat the model as a black box) or eclectic that is combination of both.
  - Portability: This refers to the range of models in which the interpretability approach can be used.
  - Algorithmic complexity: It refers to the computational complexity of the explanation algorithm.

*Ribeiro, Marco Tulio, Sameer Singh, and Carlos Guestrin. "" Why should i trust you?" Explaining the predictions of any classifier." Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining. 2016.
**Robnik-Šikonja, Marko, and Marko Bohanec. "Perturbation-based explanations of prediction models." Human and Machine Learning: Visible, Explainable, Trustworthy and Transparent (2018): 159-175.

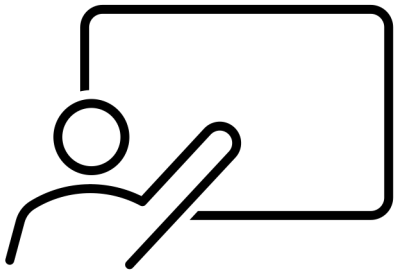
Explanation Properties (Cont.)

- Based on the quality of the explanation*, the following further properties can be defined:
  - Accuracy: the ability of the interpretability approach to generalise to unseen samples.
  - Fidelity: how well the output explanations reflect the behaviour of the model. The local fidelity, i.e. per sample, cannot necessarily generalise to the global level.
  - Consistency: it refers to the similarity between the explanations of two different models for the same sample.
  - Stability: it refers to the similarity of the explanations for similar samples.
  - Comprehensibility: how well do people understand the explanation.
  - Certainty: it reflects the confidence of the output explanation.
  - Representativeness: it refers to the number of samples that the explanation covers.

*Robnik-Šikonja, Marko, and Marko Bohanec. "Perturbation-based explanations of prediction models." Human and Machine Learning: Visible, Explainable, Trustworthy and Transparent (2018): 159-175.

“Good” Explanations

- *Miller studied how a ”good” explanation should be to be human-friendly.
- Why was my loan declined?
- We need an explanation that includes the reasons that led to the result (contrastive explanation**).
- So you need an explanation based on a reference. For example, another person with an accepted loan so that we understand why my loan was rejected.
- Moreover, ones to two causes are sufficient for a human-friendly explanation.
- A good explanation includes the unexpected causes that led to the result.



*Miller, Tim. "Explanation in artificial intelligence: Insights from the social sciences." Artificial intelligence 267 (2019): 1-38.
**Lipton, Peter. "Contrastive explanation." Royal Institute of Philosophy Supplements 27 (1990): 247-266.

Evaluation of Interpretability

- There is no clear protocol on how to measure the interpretability of a method and, more generally, how to evaluate interpretability methods.
- There can be though three types of evaluation*:
  - Application-grounded evaluation: Provide the interpretability output, alongside the model prediction, to a human expert for evaluation. The expert can access the interpretability output.
  - Human-grounded evaluation: Alternatively, an experiment can be conducted without experts. This approach would be useful when it is complex to involve a large number of experts. In this case, the human is given pairs of explanations and has to choose the one with the highest quality.
  - Functionally-grounded Evaluation: There is no human involved in this evaluation. A proxy is defined and then the interpretability output is evaluated with respect to the proxy. A human is required though prior to evaluation to access the validation of the experiment. For example, check the improvement in prediction of a model that has already been shown to be interpretable by a human expert. In this evaluation, it is not easy to define a proxy.
- *Which type of the above evaluation is the most complicated?*

*Doshi-Velez, Finale, and Been Kim. "Towards a rigorous science of interpretable machine learning." arXiv preprint arXiv:1702.08608 (2017).

Interpretable Models

- Commonly used interpretable algorithms include linear regression, logistic regression and decision trees.
- Monotone: Increasing the feature value always increases (or decreases) the output.
- Interactions: It refers to features interactions.

Algorithm*	Linear	Monotone	Interaction	Module Explanation	Task
Linear regression	✓	✓	✗	✓	Regression
Logistic regression	✗	✓	✗	✓	Classification
Decision Tress	✗	-	✓	✓	Regression & Classification
Naive Bayes	✗	✓	✗	✓	Classification
K-Nearest Neighbours	✗	✗	✗	✗	Regression & Classification

*3.2 Taxonomy of Interpretability Methods, Molnar, Christoph. Interpretable machine learning. Lulu. com, 2020.

Linear Regression

- The linear regression is an approach that computes weighted sum of the input features.
- Consider an input feature vector $\mathbf{x} \in \mathbb{R}^D$, where D is the dimension of $\mathbf{x}$.
- An output scalar value (target) $\mathrm{y} \in \mathbb{R}$.
- The linear regression model to predict the the target is defined as $\hat{\mathrm{y}} = \mathrm{f}(\mathbf{x}; \mathbf{w}) = \mathbf{w}^T\mathbf{x} + \mathrm{b}$.
- $\hat{\mathrm{y}}$ is the predicted value, $\mathbf{w} \in \mathbb{R}^n$ is the weights vector, and b is the bias (or intercept).
- This is a linear transformation, and our goal is to predict ($\hat{\mathrm{y}}$) the exact target value ($\hat{\mathrm{y}} = \mathrm{y}$) given the input feature vector $\mathbf{x}$.
- In the context of model fitting, we seek for the parameters $\mathbf{w}$ to best fit the model to our data.

Linear Regression: Learning

- The parameter values ($\mathbf{w}$, b) can be learned with the support of a training set.
- Consider the training dataset $\mathbf{X}^{train} \in \mathbb{R}^{m \times n}$, $\mathbf{y}^{train} \in \mathbb{R}^{m}$, where m denotes the number of samples.
- Similarly, a test dataset is required to measure performance of $\mathrm{f}(\mathbf{x}; \mathbf{w})$ after learning the parameters, $\mathbf{X}^{test} \in \mathbb{R}^{m \times n}$, $\mathbf{y}^{test} \in \mathbb{R}^{m}$.
- The test set can have different dimensions from the training set.
- **Important:** Our learning algorithm is allowed to experience only the training set to learn the parameters ($\mathbf{w}$, b).

Linear Regression: Loss

- Learning the parameter values from the training data set requires the loss function $\mathcal{L}(f\ (x; \boldsymbol{w}), y)$.
- The loss function $\mathcal{L}(f\ (x; \boldsymbol{w}), y)$ is the performance measure of how good is our prediction during learning the parameters on the training set. Also, the same measure is applied on the test set.
- For the task of linear regression, the performance measure, namely, mean squared error (MSE) is used on both training and test sets. Our goal is to minimize the MSE as loss function on the training set such that it results in low MSE on the test set.
  - $MSE = \frac{1}{m} \sum_{i=1}^{m} \left(\hat{y}_i^{train} - y_i^{train}\right)^2 = \frac{1}{m} \sum_{i=1}^{m} \left(\mathbf{w}^T \mathbf{x}_i^{train} - y_i^{train}\right)^2$.
  - For simplicity we assume that b is included in $\mathbf{w}$.

Linear Regression: Loss (Cont.)

- Minimizing the loss function is equivalent to finding the best $\mathbf{w}^*$ such that:
  - $\mathbf{w}^* = \arg\min_{\mathbf{w}} \frac{1}{m} \sum_{i=1}^{m} \left(\mathbf{w}^T \mathbf{x}_i^{train} - \mathrm{y}_i^{train}\right)^2$.
- *How do we obtain the optimal parameter values $\mathbf{w}^*$?*
- Solving for $\mathbf{w}^*$:
  - Normal equation: closed-form solution to find the parameter values. The gradient is set to zero and solve for $\mathbf{w}^*$. This is like an one-step algorithm (analytic approach).
  - Iterative: find iteratively the parameter values that minimize the MSE (loss function). Gradient-based optimization is a common iterative approach.
- *What are the advantages of each approach?*

Linear Regression: Normal Equation

- Loss function:
  - $MSE = \frac{1}{m}\sum_{i=1}^{m}\left(\hat{y}_i^{train} - y_i^{train}\right)^2 = \frac{1}{m}\sum_{i=1}^{m}\left(\mathbf{w}^T\mathbf{x}_i^{train} - y_i^{train}\right)^2$.
- We reformulate our loss function using matrix notation:
  - $MSE = \left(\mathbf{X}^{(train)}\mathbf{w} - \mathbf{y}^{train}\right)^T\left(\mathbf{X}^{(train)}\mathbf{w} - \mathbf{y}^{(train)}\right)$.
  - We skip the term $\frac{1}{m}$ because it has no impact to the optimal parameter values.
- Now we expand the above equation to obtain:
  - $MSE = \left(\mathbf{w}^T\mathbf{X}^{(train)T} - \mathbf{y}^{(train)T}\right)\left(\mathbf{X}^{(train)}\mathbf{w} - \mathbf{y}^{(train)}\right) = \mathbf{w}^T\mathbf{X}^{(train)T}\mathbf{X}^{(train)}\mathbf{w} - \mathbf{w}^T\mathbf{X}^{(train)T}\mathbf{y}^{(train)} - \mathbf{y}^{(train)T}\mathbf{X}^{(train)}\mathbf{w} + \mathbf{y}^{(train)T}\mathbf{y}^{(train)} = \mathbf{w}^T\mathbf{X}^{(train)T}\mathbf{X}^{(train)}\mathbf{w} - 2\mathbf{w}^T\mathbf{X}^{(train)T}\mathbf{y}^{(train)} + \mathbf{y}^{(train)T}\mathbf{y}^{(train)}$.
  - Note that $\mathbf{y}^{(train)T}\mathbf{X}^{(train)}\mathbf{w} = \left(\mathbf{y}^{(train)T}\mathbf{X}^{(train)}\mathbf{w}\right)^T = \mathbf{w}^T\mathbf{X}^{(train)T}\mathbf{y}^{(train)}$.

Linear Regression: Normal Equation (Cont.)

- Next, we can obtain $\boldsymbol{w}^*$ by the setting the gradient of the MSE w.r.t $\mathbf{w}$ to zero and solve for it.
  - $\nabla_{\mathbf{w}} MSE = 0 \Rightarrow$
  - $\nabla_{\mathbf{w}} \left(\mathbf{w}^T \mathbf{X}^{(train)T} \mathbf{X}^{(train)} \mathbf{w} - 2\mathbf{w}^T \mathbf{X}^{(train)T} \mathbf{y}^{(train)} + \mathbf{y}^{(train)T} \mathbf{y}^{(train)} \right) = 0 \Rightarrow$
  - $2\mathbf{X}^{(train)T} \mathbf{X}^{(train)} \mathbf{w} - 2\mathbf{X}^{(train)T} \mathbf{y}^{(train)} = 0 \Rightarrow$
  - $\mathbf{X}^{(train)T} \mathbf{X}^{(train)} \mathbf{w} = \mathbf{X}^{(train)T} \mathbf{y}^{(train)} \Rightarrow$
  - $\mathbf{w} = \left(\mathbf{X}^{(train)T} \mathbf{X}^{(train)} \right)^{-1} \mathbf{X}^{(train)T} \mathbf{y}^{(train)}$.
- Our solution is $\mathbf{w} = \left(\mathbf{X}^{(train)T} \mathbf{X}^{(train)} \right)^{-1} \mathbf{X}^{(train)T} \mathbf{y}^{(train)}$.
- Using the above equation we can obtain the parameter value within <u>one step</u>.

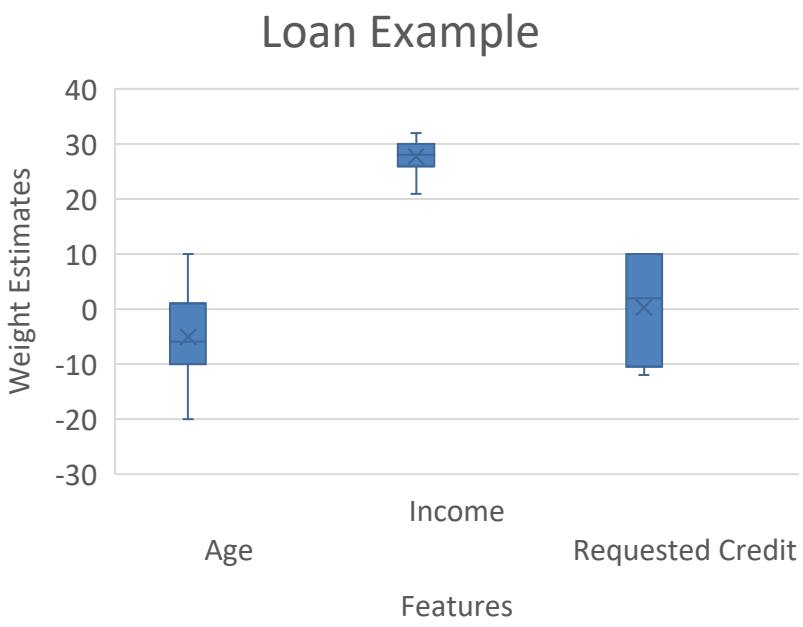
Linear Regression: Interpretation

- *How do we interpret the weights of the linear regression?*
- The linear relationship between input and weights/coefficients makes the mode easy to interpret. Features with high weights have more influence to the output and vice versa.
- Feature importance: It can be measured by the absolute value of the t-statistic. The t-statistic is the ratio between the estimated weight and its standard error.
- A example below:

Feature	Estimated Weight	Error	T-statistic
Age	-10	20	0.5
Income	30	5	6
Requested Credit	5	25	0.2

Linear Regression: Interpretation (Cont.)

- Moreover, visualising linear regression models is easy and intuitive.
- A weight plot can be created to examine their importance. For example, a box plot can be used to visualise the weights.
- Individual predictions can be also plotted, e.g. features and outcome.



Lasso Regression

- Lasso regression can be used for eliminating less important features.
- It is a variation of linear regression with L1-regularization.
- Loss function:
  - $\frac{1}{m}\sum_{i=1}^{m}\left(\mathbf{w}^T\mathbf{x}_i^{train} - y_i^{train}\right)^2 + \lambda\sum_{j=1}^{p}|w_j|$.
  - $\frac{1}{m}$ can be skipped.
- The tuning parameter λ controls the influence of the regularizer.
- The L1-reugalarisation is good at brining some of the input features to 0 (feature elimination).
- Thus, the model can have sparse parameters.

Linear Regression: Advantages & Disadvantages

- The weighted sum output can be easily interpreted.
- The visualization of the parameters is easy, as well as computing statistics.
- However, linear regression cannot mode complex and non-linear relationships between the features and output.

Local Model-Agnostic Approaches

- Theses approaches rely on individual prediction to provide explanation. A few standard approaches are:
  - Individual Conditional Expectation (ICE).
    - They are plots that show how changing a feature affects the output, e.g. predicted loan probability (output) vs person age.
    - *Goldstein, Alex, et al. "Peeking inside the black box: Visualizing statistical learning with plots of individual conditional expectation." journal of Computational and Graphical Statistics 24.1 (2015): 44-65.*
  - Local interpretable model-agnostic explanations (LIME).
    - Explain individual predictions of a black box model. A surrogate model is trained to approximate the predictions of the black box model.
    - *Ribeiro, Marco Tulio, Sameer Singh, and Carlos Guestrin. "" Why should i trust you?" Explaining the predictions of any classifier." Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining. 2016.*

Local Model-Agnostic Approaches (Cont.)

- Theses approaches rely on individual prediction to provide explanation. A few standard approaches are:
  - Counterfactual Explanations.
    - They look for the features to be changed to provide the wished output. The challenge is to have an approach that performs better than trial and error.
    - *Wachter, Sandra, Brent Mittelstadt, and Chris Russell. "Counterfactual explanations without opening the black box: Automated decisions and the GDPR." Harv. JL & Tech. 31 (2017): 841.*
  - Shapley value.
    - Assuming that each feature is a ”player” in a game where the model prediction corresponds to the pay-out. The Shapley values show how to fairly distribute the pay-out among the players (features).
    - *Shapley, Lloyd S. "A value for n-person games." (1953): 307-317.*

Local interpretable model-agnostic explanations (LIME).

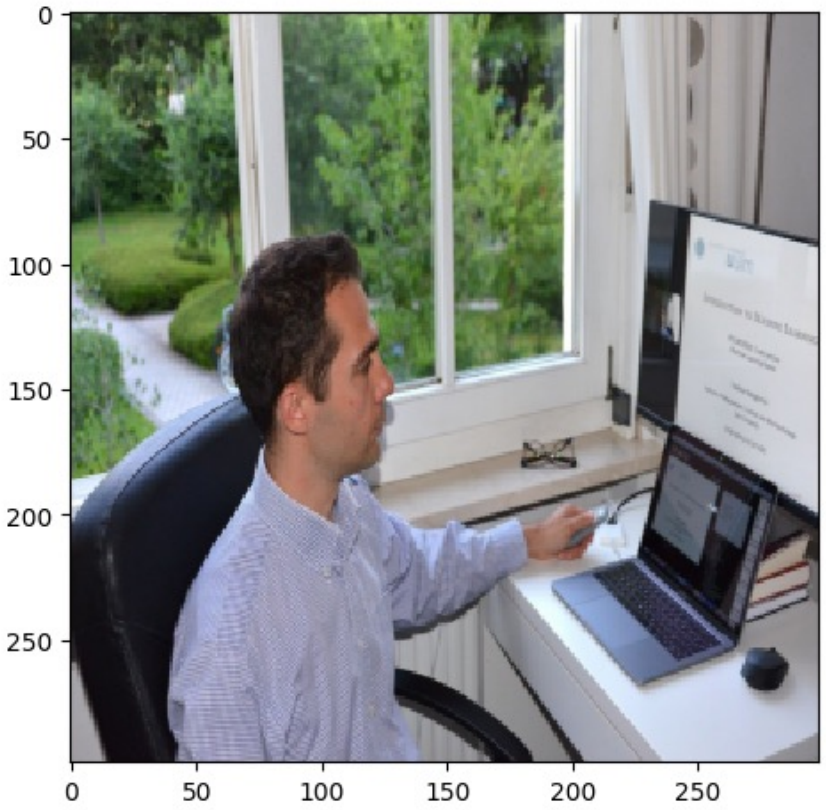
- The main idea of LIME is to approximate the black box machine learning model locally (per sample) using a surrogate model. The surrogate model is a simple interpretable model.
- The approach is based on the idea of perturbing the input features of a data sample, e.g. hiding image pixels. By perturbating the original sample several times, a new dataset is generated with perturbed samples.
- Then for the perturbed training data, the new predictions are computed, while they are also weighted based on their similarity to the original data.
- Using the perturbations, predictions and the computed weights, the surrogate model is approximated by a simple interpretable model, e.g. linear regression or a shallow decision tree. This stage involves training a new simple and interpretable model.

LIME Algorithm

- Pick up a sample (e.g., image) and the black box machine learning model (e.g., deep neural network).
- Perturb the sample multiple times and then get prediction for each perturbation from the black box model.
- Weight each perturbed sample based on each distance from the original sample.
- Train an interpretable model on the perturbations, including the weights and predictions.
- Rely on the interpretable model to provide explanations.

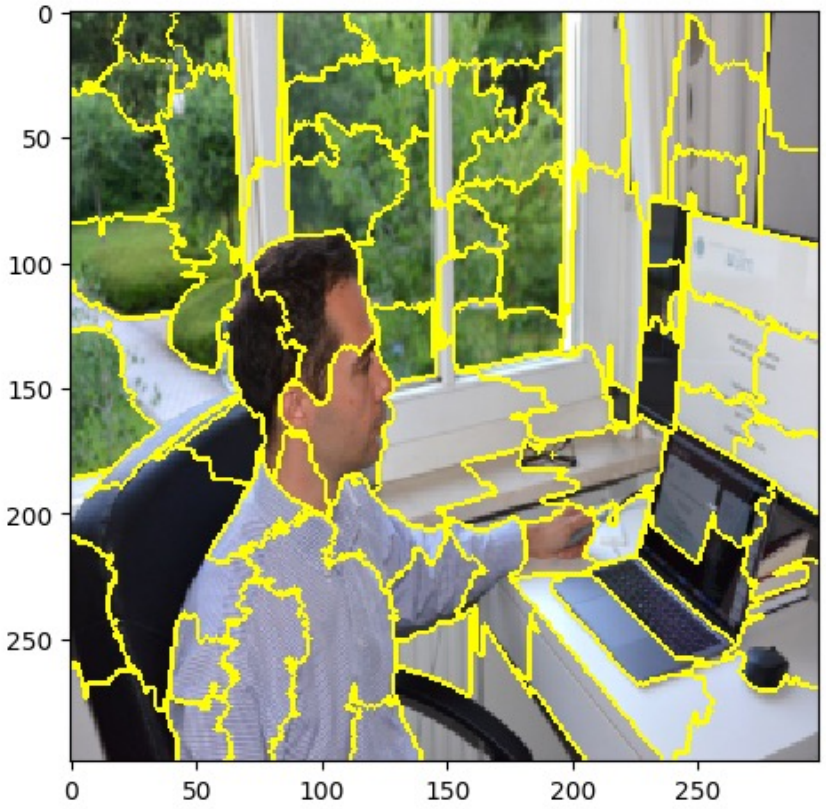
LIME Example

- The task is image classification with a convolutional neural network.
- The top-5 predictions using the Inception-V3 model are:
  - Photocopier
  - Laptop
  - Notebook
  - Cash machine
  - Desktop machine
- We seek for an explanation for the top-1 output.



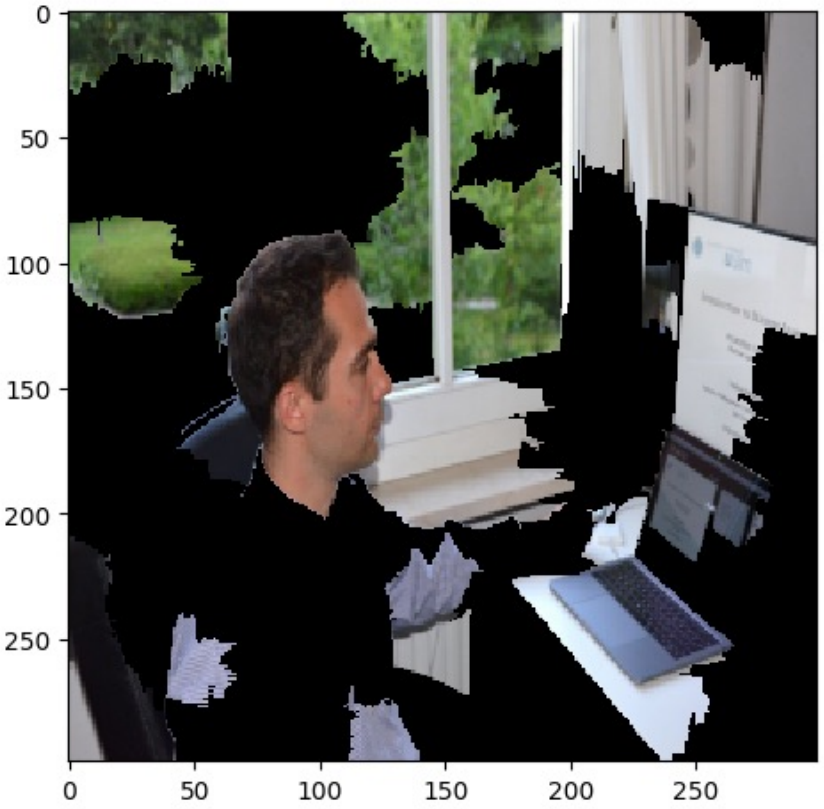
LIME Example (Cont.)

- To perform the input image perturbation, we rely on superpixels instead of removing individual pixels.
- *What is the benefit of the superpixels?*
- We randomly choose to remove a number of superpixels. This is a perturbation. Moreover, each perturbation is a new training sample.
- We perform several perturbations, e.g. 300.



LIME Example (Cont.)

- This a one example of a perturbed image.
- For each perturbed image, we perform a prediction using the Inception-V3 model.
- Next, we compute the distance between every perturbed image and the original image. For instance, we can use the cosine similarity.
- We convert the distance to a weight using some kernel function.

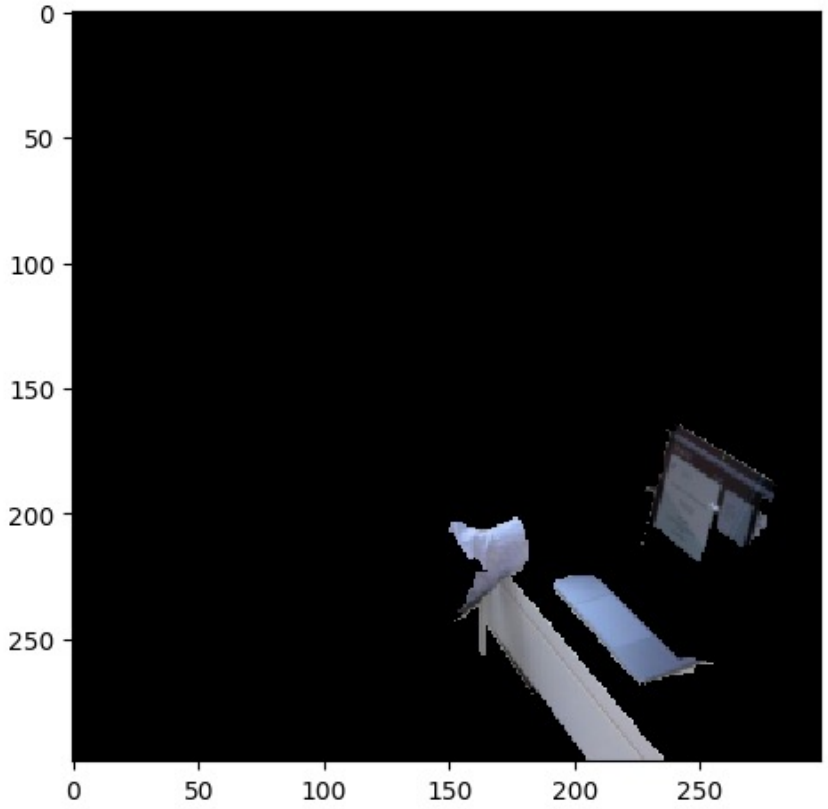


LIME Example (Cont.)

- We fit a weighed linear regression model to the perturbations, top-1 prediction and weights of the perturbed data samples.
- Our model has as many parameters as the number of superpixels. Thus, we expect to the same number of coefficient values.
- Linear regression is an interpretable model. The value of each coefficient tell us the importance of the superpixel.

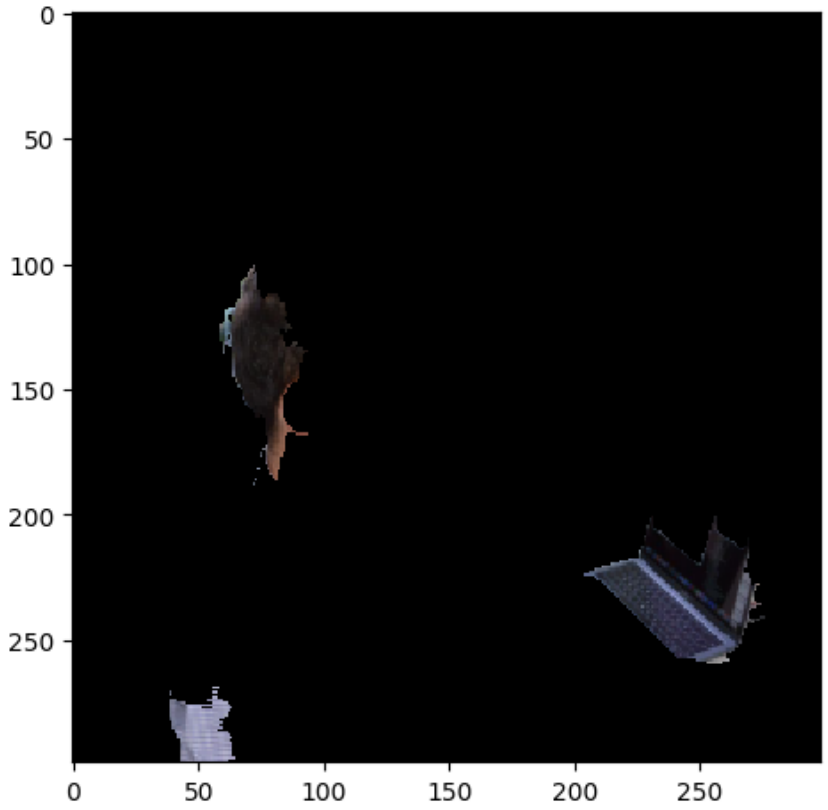
LIME Example (Cont.)

- We select the top-4 coefficients, for example, with the highest value.
- These are the superpixels that lead to the top-1 prediction (photocopier) on the original image.
- Finally, we visualise these superpixels. This is our visual explanation for the original prediction.
- Note that we could even pick more than 4 superpixels.



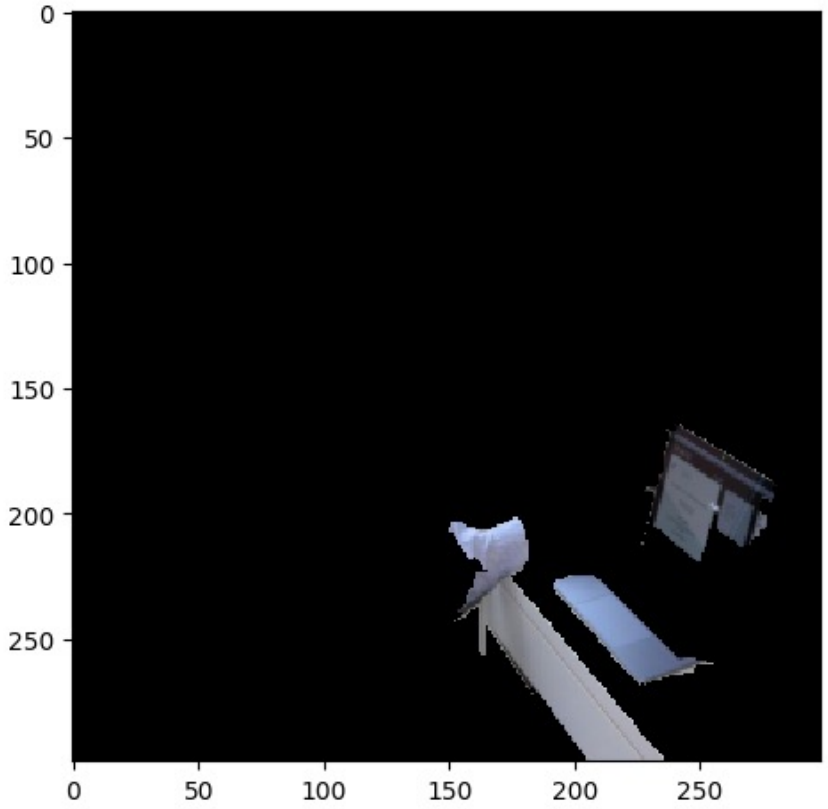
LIME Example (Cont.)

- In the same way, we could examine the pixels that lead to the top-2 prediction (laptop).
- While the top-1 prediction is incorrect, the top-2 is correct. In the visual explanation there are superpixels including the laptop.
- *What is a major limitation on using a linear interpretable model?*



LIME Properties

- The algorithm has been tested for tabular data, text and images. The perturbations for text can be removing words.
- The algorithm can be expensive because of the perturbations. A large number of perturbed samples helps to achieve a better model fit.
- LIME is not always stable. It has shown to produce different results for very similar samples*.
- It can be applied to any kind of machine learning model, not only on neural networks.



*Alvarez-Melis, David, and Tommi S. Jaakkola. "On the robustness of interpretability methods." arXiv preprint arXiv:1806.08049 (2018).

Study Material

Not for sharing (LMS, Friedrich-Alexander-Universität Erlangen-Nürnberg)

- Molnar, Christoph. Interpretable machine learning. Lulu. com, 2020 (Chapter 3 Interpretability, 5 Interpretable Models, 9 Local Model-Agnostic Methods)
- Relevant publications as reported in the corresponding slides.

Next Lecture

Attention and Transformers