Karlsruher Institut für Technologie

Machine Learning 1 – Fundamentals

Summary

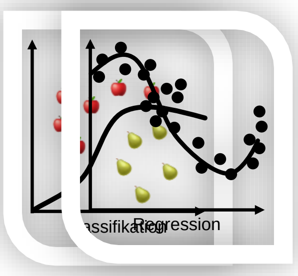
Prof. Dr. J. M. Zöllner, M.Sc. Nikolai Polley, M.Sc. Marcus Fechner

Outline

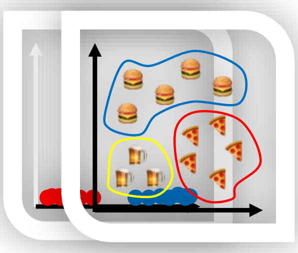
- Recap on ML1
- What‘s Next?
- Evaluation
- Exam

Introduction

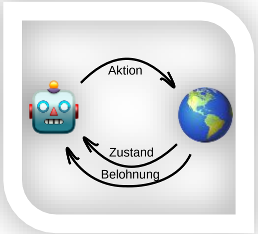
- Supervised Learning



- Unsupervised Learning



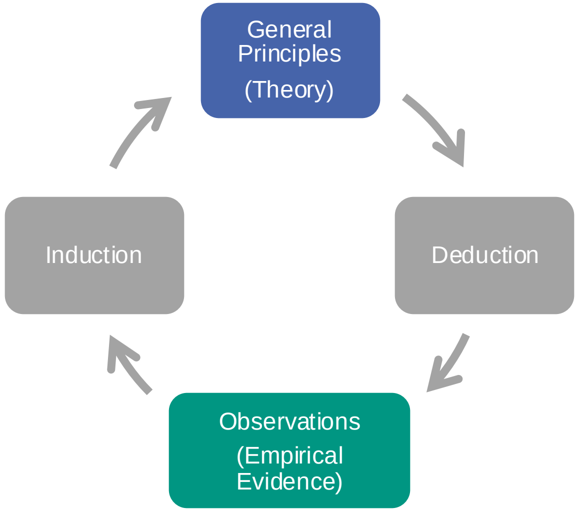
- Reinforcement Learning



Inductive Learning

- **Goal**: The machine learning method should find the best hypothesis $h \approx t$ in the large hypothesis space H, that best fits the observed data

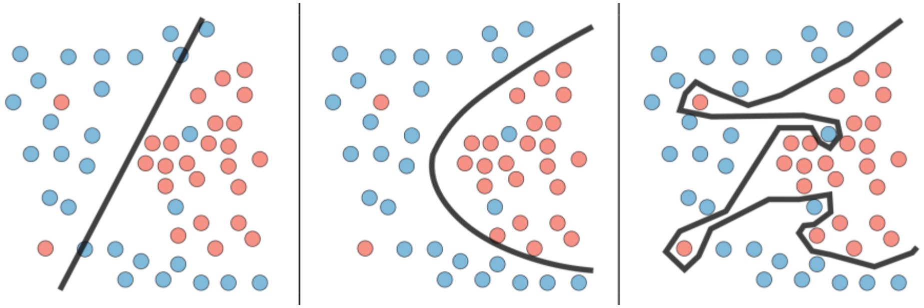
- **Induction**
- **Deduction**



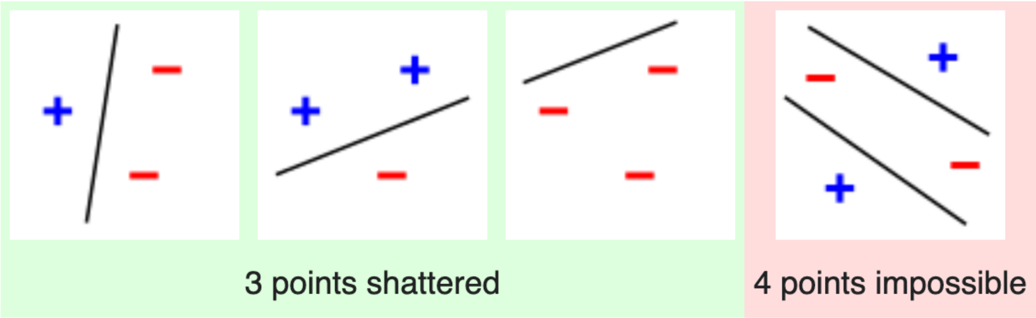
- **Inductive learning hypothesis**
- **Induktive bias:** Certain hypotheses are preferred over other hypotheses in the hypothesis space

Learning Theory

- **Learning system** = Hypothesis space + Learning method
- **Empirical risk minimization**: $\hat{\mathcal{L}}_{\mathrm{D}}(h_\theta) = \mathbb{E}_{(x,y)\sim\hat{p}}[\ell(h_\theta(x),y)] = \frac{1}{|\mathrm{D}|}\sum_{(x,y)\in D}\ell(h_\theta(\mathrm{x}),y)$
- **Generalization gap** → Train Error | Validation Error | Test Error
- **Learning is challenging**
- **Overfitting vs underfitting**

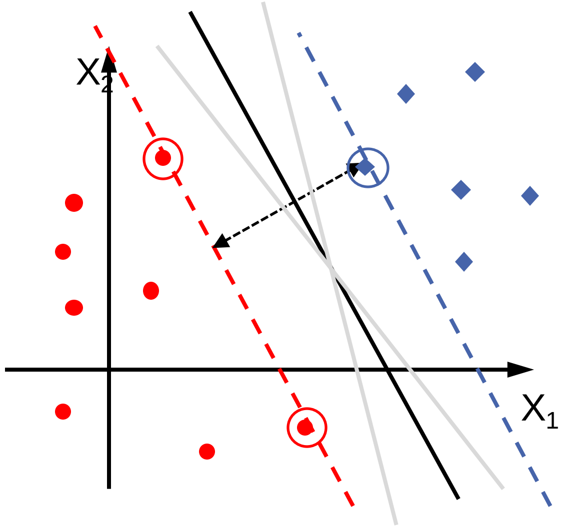


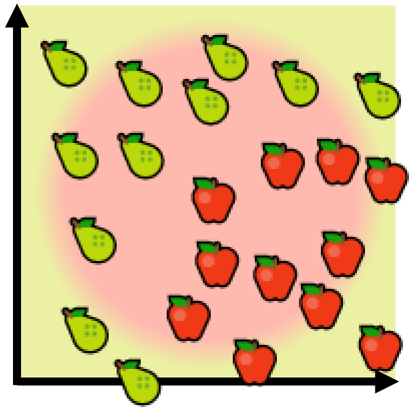
- **Evaluate model**: Metrics, Cross-validation
- **Improve model**: Boosting, Bagging, Adaboost
- **VC dimension**



Support Vector Machine

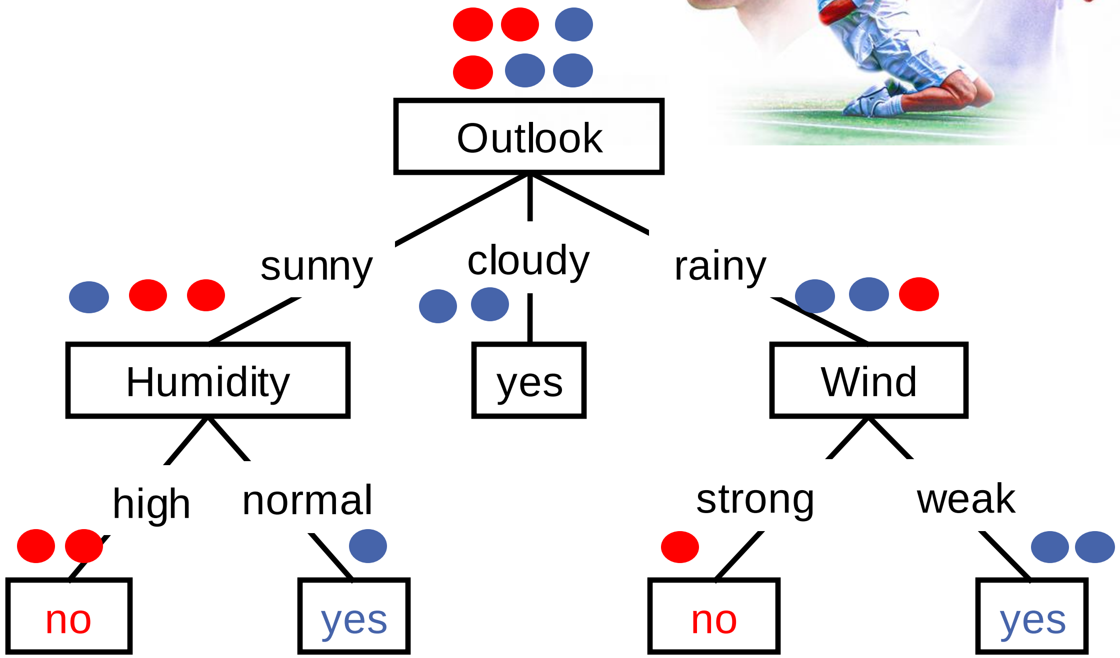
- **Problem**: There are many ways to separate the two sets (red and blue), but what is the optimal solution to the problem?
- **Solution**: Find the best separating line/hyperplane with maximum margin to the classes
- **Support vectors**
- **Soft Margin SVM**
- **Non-linear SVM**
  - **Kernel function**
  - **Kernel-trick**





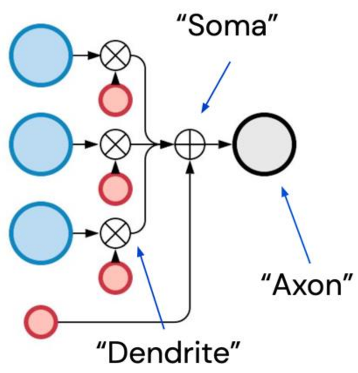
Decision Trees

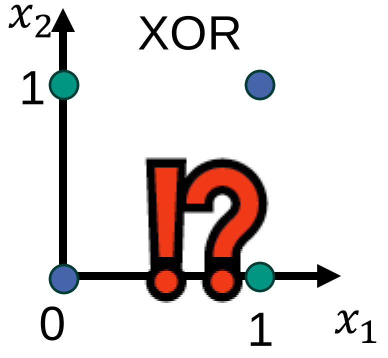
- **Properties**
  - Non-parametric
  - Interpretable
- **Attribute selection**
  - Entropy
  - Information gain
- **ID3 Algorithm**
  - Top-Down
  - Greedy
  - Prone to overfitting
- **Reduce overfitting**
  - Early Stopping
  - Pruning
  - Bagging
  - Random Forests

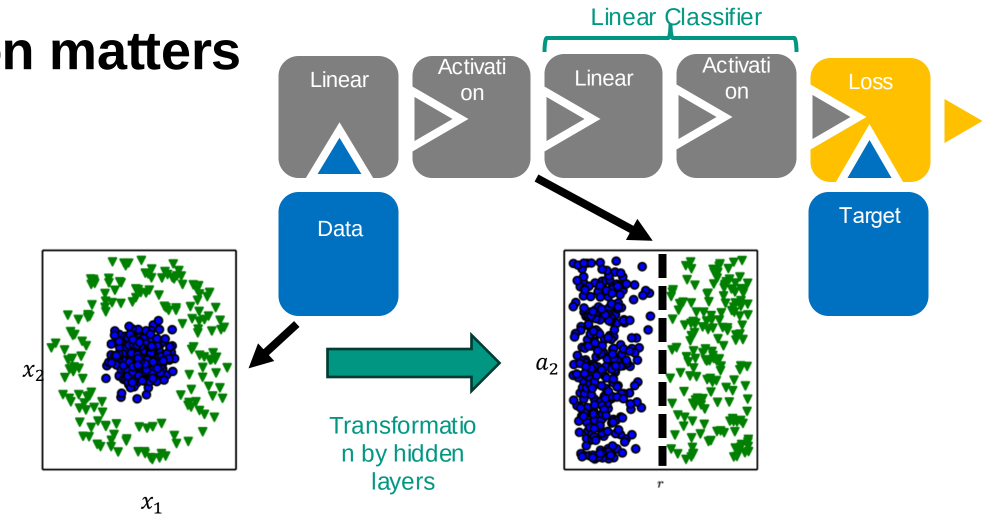


Neural Networks - Basics

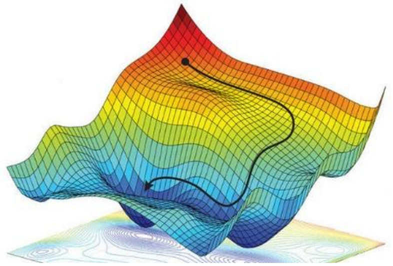
- **Artificial neuron** = Mathematical approximation of a real neuron
- **XOR Problem**
- **Representation matters**





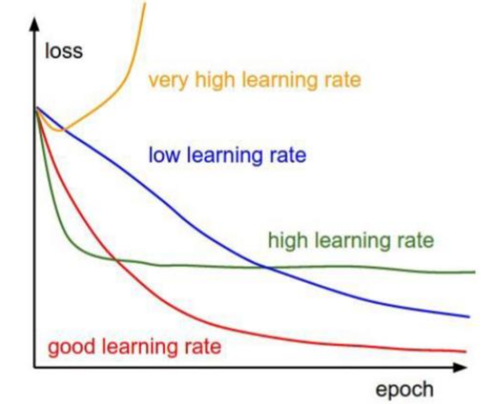


- Multi-layer neural networks are **universal function approximators**
- **Learning** = Optimization = Gradient Descent
- **Backpropagation** = Efficient application of the chain rule



Neural Networks - Hyperparameter

- **Hyperparameter**: Manually defined settings of training algorithm
  - Badly chosen settings prevent successful training



- **Optimization Methods**
  - SGD, Momentum, ADAM,
  - Static and dynamic learning rates

- **Regularization**

$$\tilde{\mathcal{L}}(\boldsymbol{\theta}) = \underbrace{\hat{\mathcal{L}}(\boldsymbol{\theta})}_{\text{Original Loss}} + \underbrace{\lambda R(\boldsymbol{\theta})}_{\text{Regularization Term}}$$

- **Initialization of parameters**
  - Bad intialization can lead to vanishing gradient

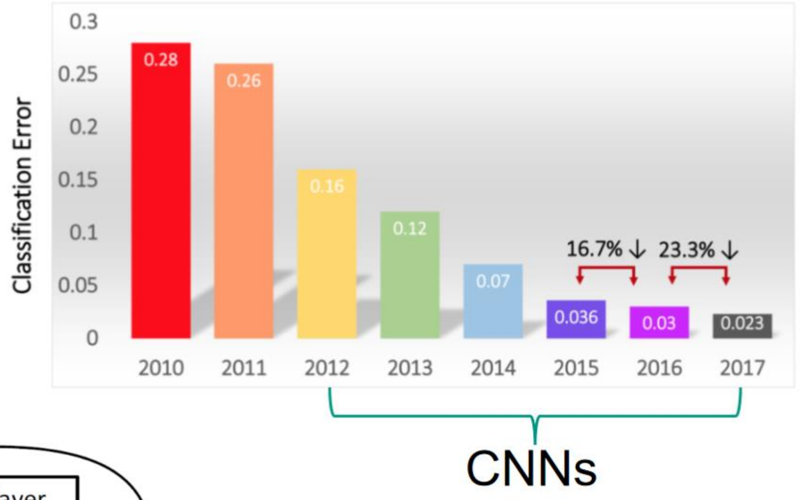
Hyperparam	RoBERTa$_{\text{LARGE}}$
Number of Layers	24
Hidden size	1024
FFN inner hidden size	4096
Attention heads	16
Attention head size	64
Dropout	0.1
Attention Dropout	0.1
Warmup Steps	30k
Peak Learning Rate	4e-4
Batch Size	8k
Weight Decay	0.01
Max Steps	500k
Learning Rate Decay	Linear
Adam ϵ	1e-6
Adam β_1	0.9
Adam β_2	0.98
Gradient Clipping	0.0

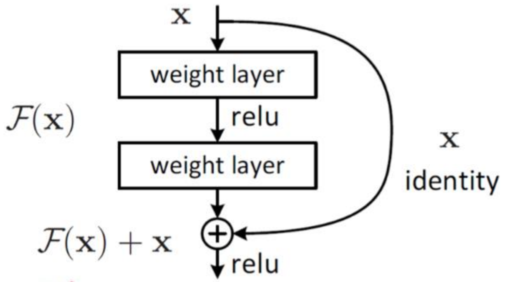
Covered in ML2

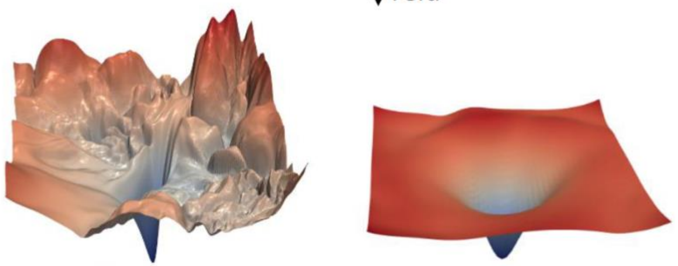
Covered in ML1

Convolutional Neural Networks

- Input with spatial relations
- Base Operator: **Convolution**
  - Small kernel that gradually increase the receptive field layer after layer
- Layers and activation functions
- Architectures (ResNet)





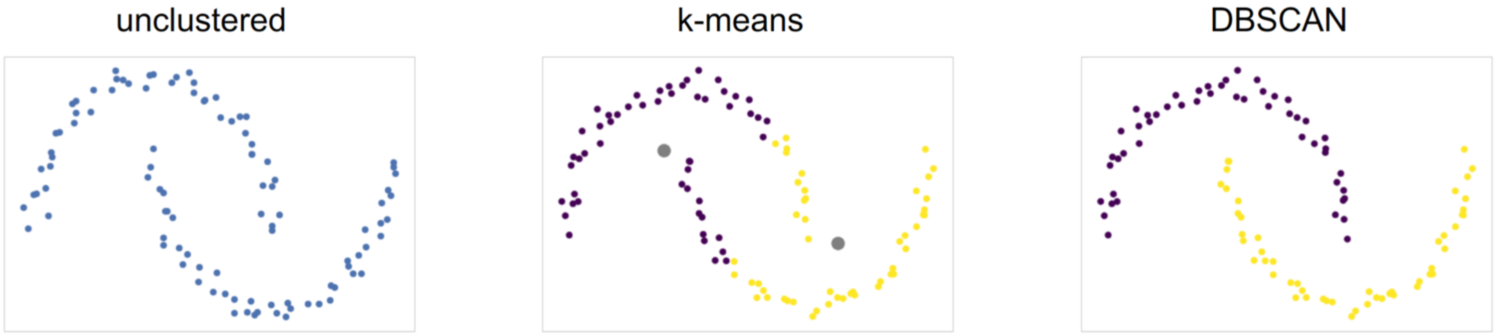


(a) without skip connections

(b) with skip connections

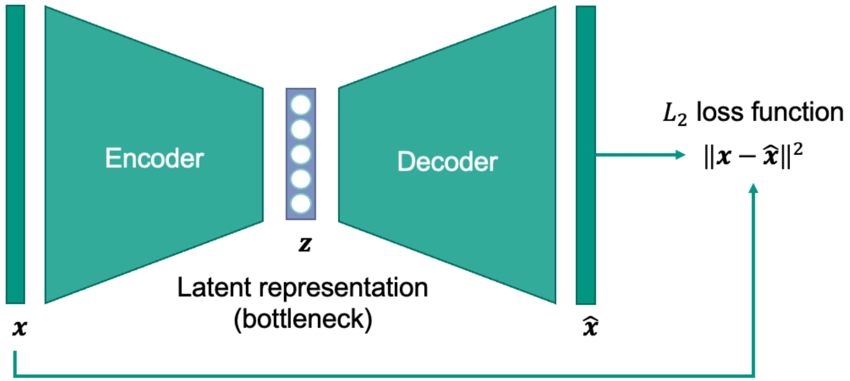
Unsupervised Learning

- Examples only contain input data (**unlabeled data**)
  - Goal: Find the **underlying structure** in the data
  - Clustering or generative



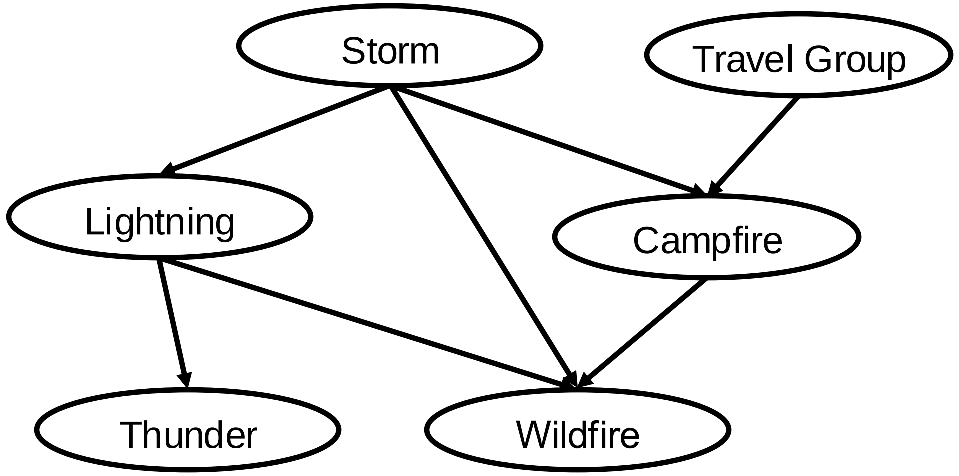
- **K-Means**
  - How it works
  - What if it doesn‘t work
  - Adaptations
- **DBSCAN**

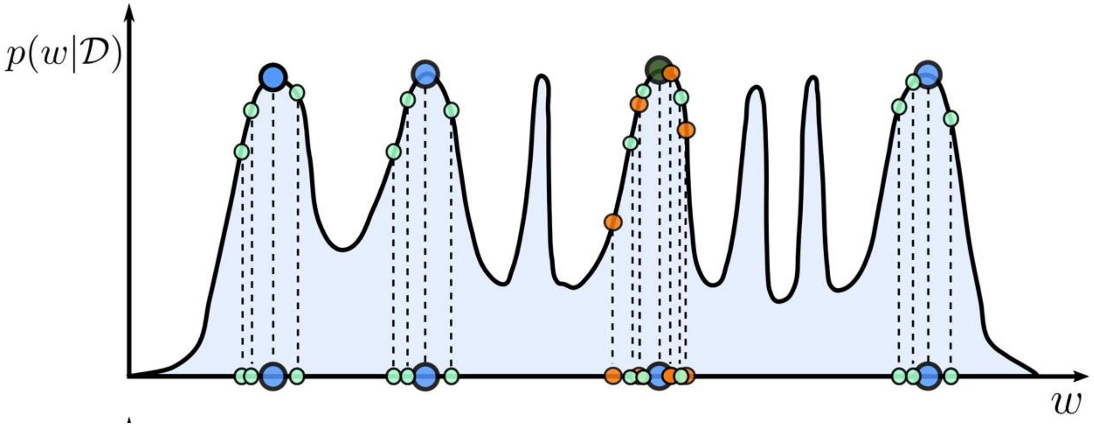
- **Autoencoder**



Bayesian Learning

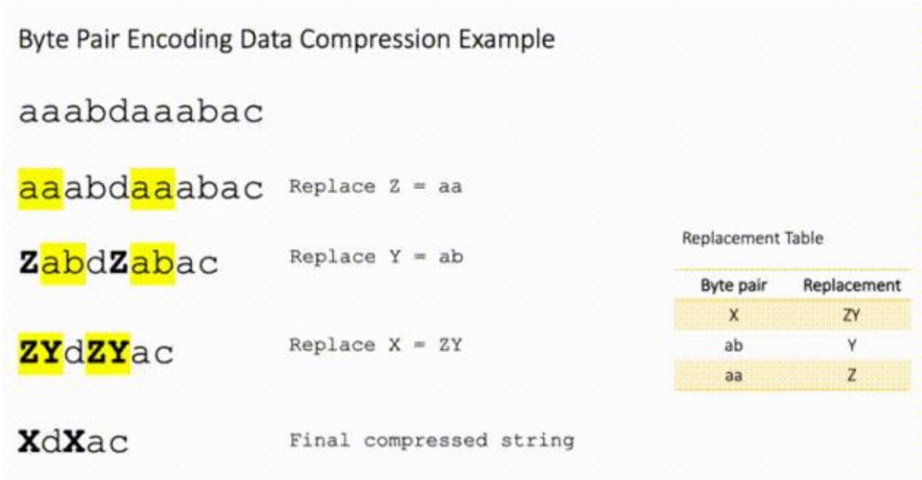
- **Why**?
  - Combining prior knowledge with observed data
- **Bayes‘ theorem**: $P(h|D) = \frac{P(D|h)P(h)}{P(D)}$
- **Estimation**
  - Maximum a posteriori
  - Maximum likelihood
- **Classifier**
  - Optimaler Bayes Klassifikator
  - Naiver Bayes Klassifikator
- **Bayesian networks**
- **Expectation-maximization algorithm**

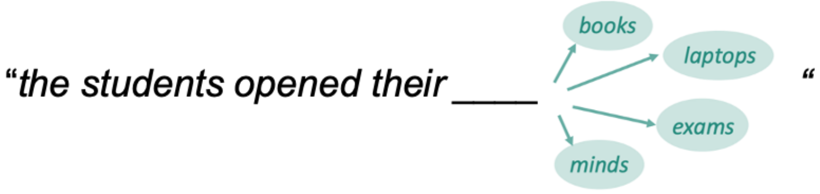


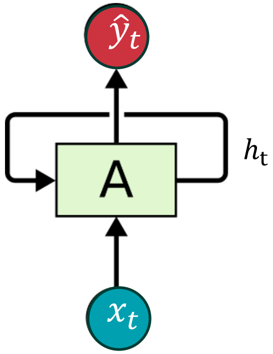


NLP and Sequence Models

- **NLP**: Enable computers to understand, interpret and generate natural language
- **Challenges**: Language ambiguity, syntax, grammar, polysemy, etc.
- **Text Normalization**: Convert text to standardized format
- **Tokenization**: Break text into meaningful units (tokens)
  - Space-based tokenization
  - Byte-pair encoding
- **Text representation**: Word embeddings
- **Language models**: Task of predicting the next token
  - Sequence models
  - RNN, LSTM
- **Decoding**: Convert output of model to text



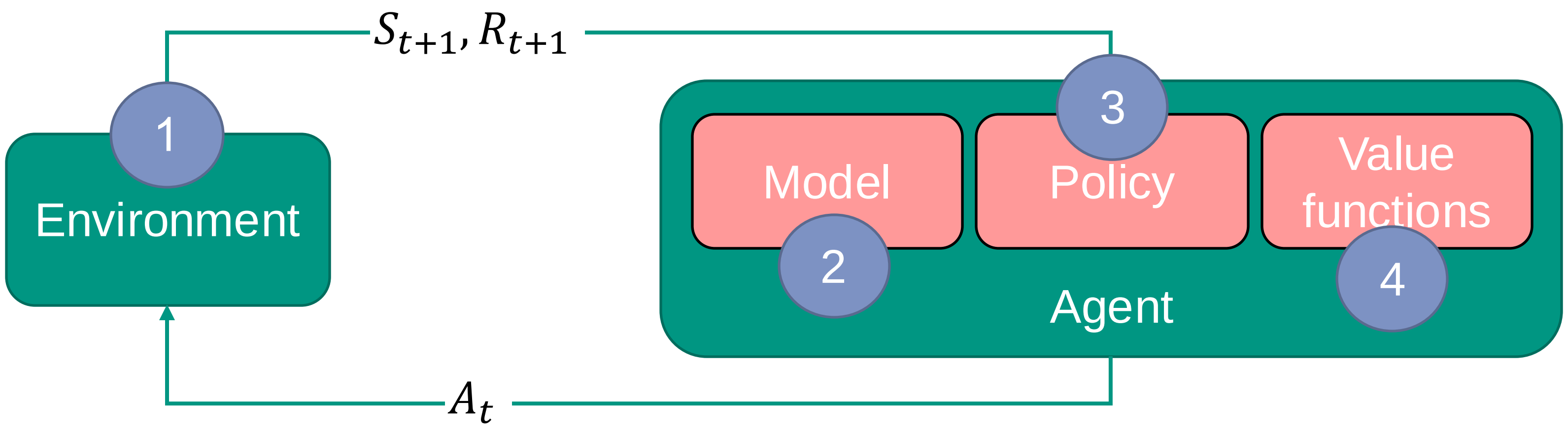




$$\hat{\boldsymbol{x}}_t = g(p(\boldsymbol{x}_t|\boldsymbol{x}_{1:t-1}))$$

Decoding algorithm

Reinforcement Learning (RL)



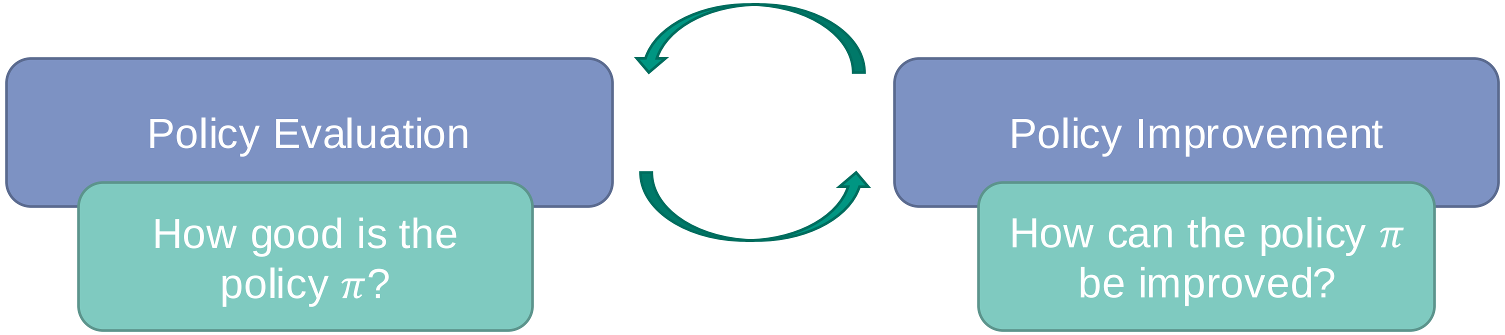
1. Mathematically formulated as MDP
2. Environment behavior predicted by the agent
3. Behavior of the agent for all states
4. Evaluation of states/action for a given policy

- Sequential decision making problem
- Data is actively generated by interacting with the environment
- Basis for RL is the reward hypothesis
- **Reward hypothesis:** Any goal can be formalized as the outcome of maximizing a cumulative reward!

$$\mathrm{E}_{\tau\sim\pi}\left[\sum_{k=0}^{\infty}\gamma^k R_{t+1+k}\right]$$

Reinforcement Learning (RL)

- An MDP is solved if an optimal policy π^* is found.
  - **Policy Iteration:**



- **Dynamic Programming:**
  - Computes the optimal value function and policy **iteratively** if the **true model is known**.
  - **Value Iteration:** Combines the evaluation and improvement step into one update step.

$$V_{k+1}(s) \leftarrow \max_{a \in A} \left(r(s,a) + \gamma \sum_{s' \in S} p(s'|s,a)\, V_k(s') \right)$$

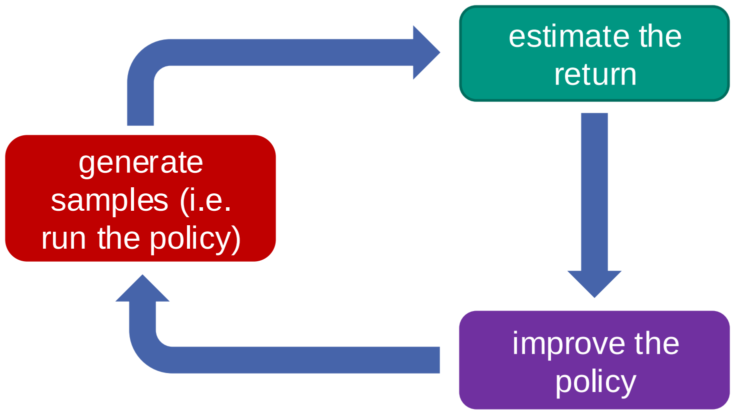
Reinforcement Learning (RL)

Karlsruher Institut für Technologie

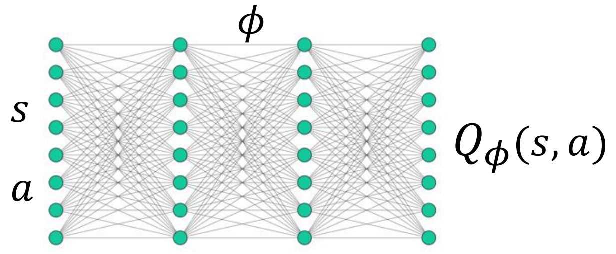
- Reinforcement Learning by interacting with the environment (**sampling**) due to an **unknown model** → **exploration**

- **Value-based RL:** → Learn value function and derive policy
  - **Monte-Carlo Methods:** Estimate by sampling
  - **Temporal-Difference (TD) Learning:** Learn by bootstrapping
    - Reduce the TD error between value functions of successive states
    - SARSA (on-policy TD control) and Q-Learning (off-policy TD control)
  - **n-step bootstrapping:** Mix of Monte-Carlo and Temporal Difference
  - **Deep Q-Learning:** Use neural nets for function approximations
    - Catastrophic forgetting → replay buffer
    - Changing target values → target network
    - Overestimation of Q-values → double Q-Learning

Fit $Q_\pi(s,a)$



$$a^* = \arg\max_{a \in A} Q_\pi(s,a)$$



Outline

- Recap on ML1
- What‘s Next?
- Evaluation
- Exam

Courses

🎓 Lectures

- Machine Learning 1 – Fundamentals ❄️
  - Fundamentals of learning systems
- Machine Learning 2 – Advanced Methods 🌞
- Advanced and current methods from research
  - e.g. Diffusion Models, Foundation Models, Transformer, Object Detection, Deep Reinforcement Learning, Self-Supervised Learning, Generative Neural Networks, Active Learning

🚀 Seminar & Practical Labs

- Seminar: Cognitive Automobiles and Robots 🌞 ❄️
  - Development of a theoretical research task in the field of ML/autonomous driving and the state of the art.
- Practical Lab: Cognitive Automobiles and Robots ❄️
  - Project in the field of ML/autonomous driving, which is to be implemented in practice.
- Practical Lab: Machine Learning 🌞
  - Project in the field of ML/autonomous driving, which is to be implemented in practice.

Explanation: 🌞 Summer term ❄️ Winter term

Courses

Applications are closed

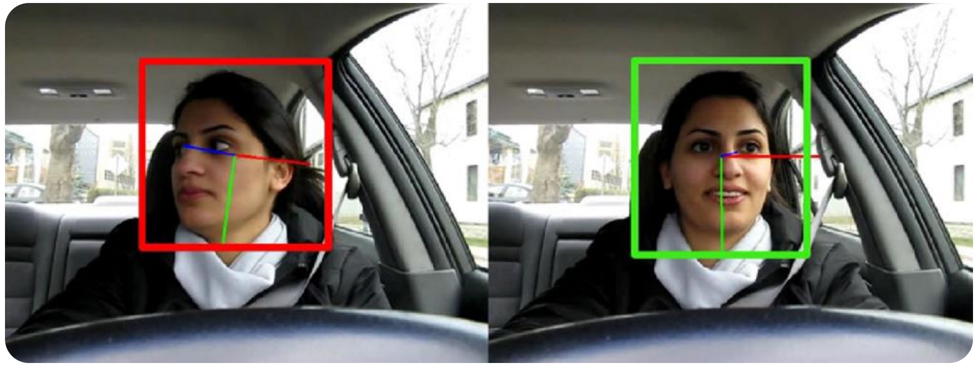
Applications are open

🚀 Seminar & Practical Labs

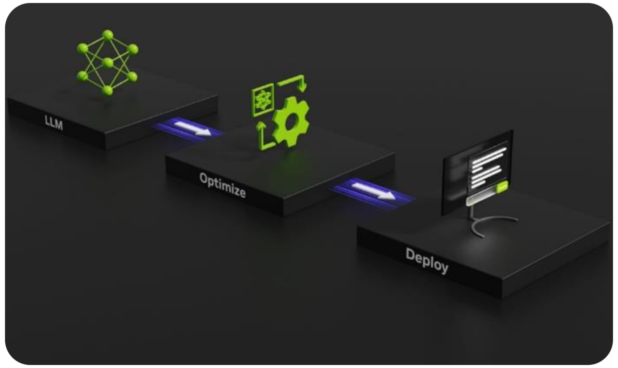
- Seminar: Cognitive Automobiles and Robots 🌞 ❄️
  - Development of a theoretical research task in the field of ML/autonomous driving and the state of the art.
- Practical Lab: Cognitive Automobiles and Robots ❄️
  - Project in the field of ML/autonomous driving, which is to be implemented in practice.
- Practical Lab: Machine Learning 🌞
  - Project in the field of ML/autonomous driving, which is to be implemented in practice.

Explanation: 🌞 Summer term ❄️ Winter term

Practical Lab: Machine Learning

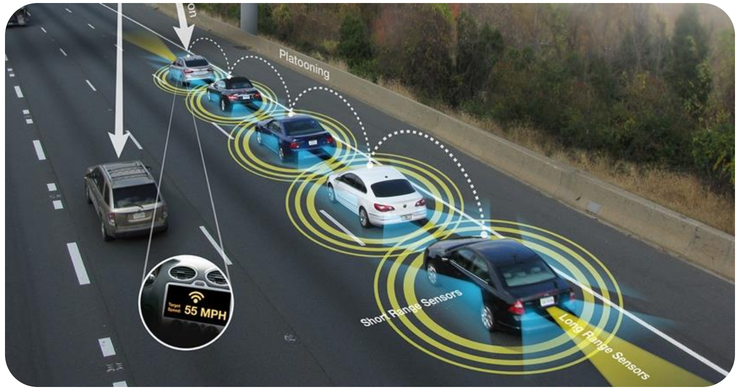


Head-Pose-Estimation
from Mobile Devices



Increasing Network Efficiency

Learning to Walk
with RL



Simulation and V2V Communication for
Platooning

Advertisement

- We are always looking for motivated students!
- **Topics**: Perception | Prediction | Mapping | Planning | Safety and Security | Vehicle-2-X | Simulation | End-to-End Learning | Safe AI | Mixed Reality | Virtual Reality | Reinforcement Learning | UX
- **Where**: Many interesting offers in the field of machine learning can be found on our homepage.
- **Offers**: Bachelor theses, master theses, student assistant positions
- **Nothing there? Feel free to send an email to our doctoral students!**

Outline

- Recap on ML1
- What‘s Next?
- Evaluation
- Exam

Evaluation: Lecture and Exercise

- https://onlineumfrage.kit.edu
- TAN:

Lecture: X8XR2

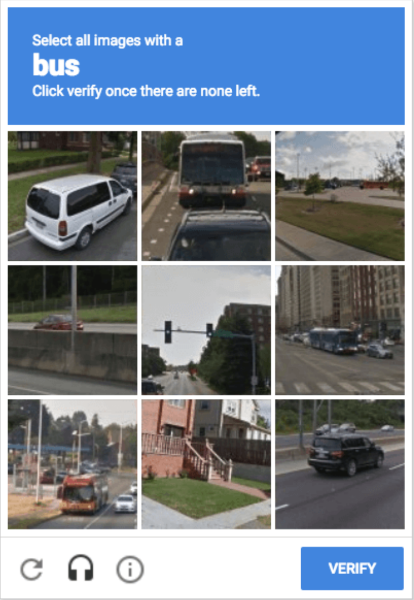
Exercise: MC3G8

Outline

- Recap on ML1
- What‘s Next?
- Evaluation
- Exam

Project Task: CAPTCHA

- The project task is finished
- Top 3:
  - Team Hummingbird 87.92%
  - Team Apple Moth 87.54%
  - Team Cobra 87.29%



Some Statistics:

Number of Uploads: 2327

Number of Teams: 290

Current Leaderboard:

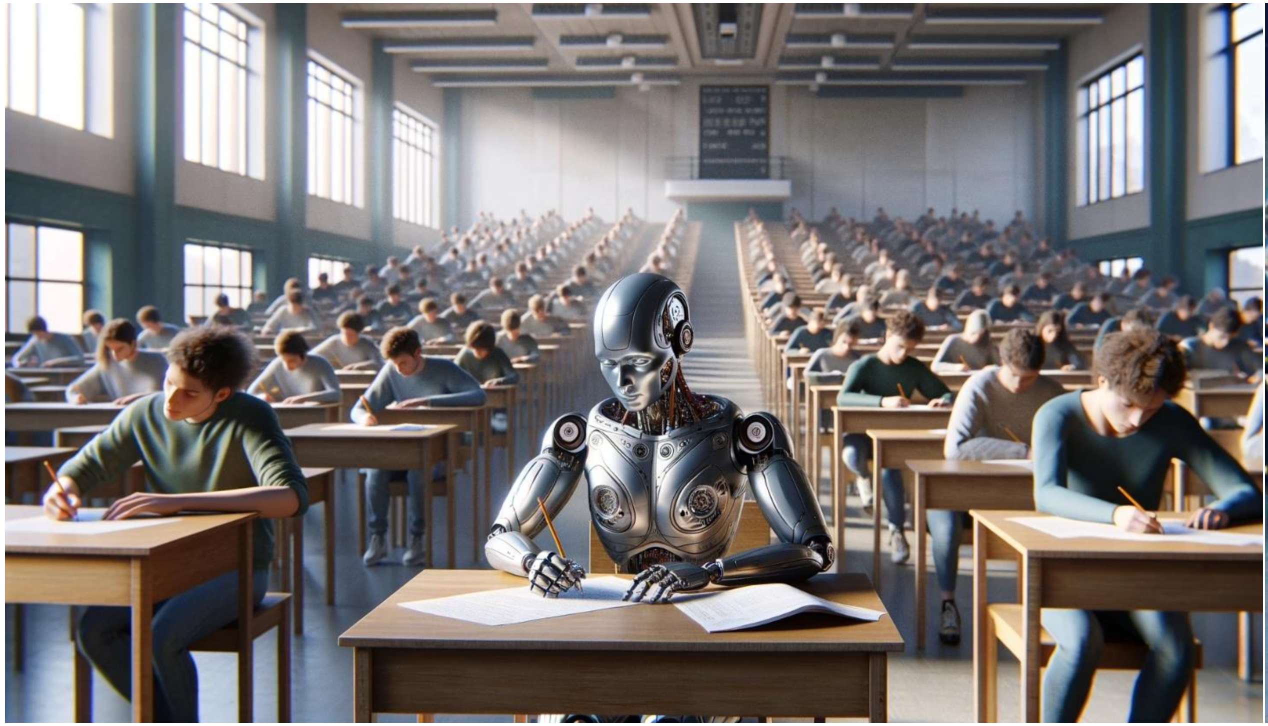
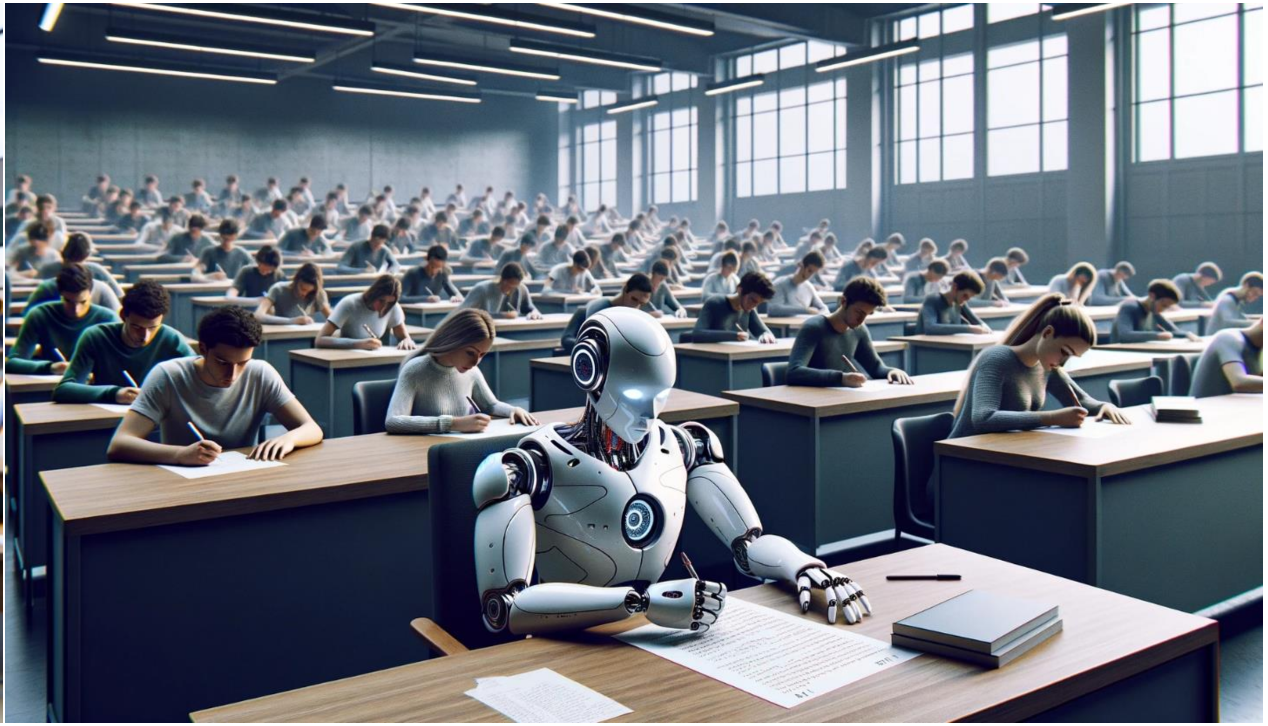
	TeamName	↓ Accuracy
1	Übungsleitung Large-CNN	89.59
2	Team Humming Bird	87.92
3	Team Apple Moth	87.54
6	Team Cobra	87.29
5	Team African Golden Cat	87.29
4	Team Quail	87.29
9	Team Angelfish	87.27
8	Team Emu	87.27
7	Team Rattlesnake	87.27
10	Team Asian Giant Hornet	87.21

Notes for Exam

- Bring your calculator!
- 60 minutes, 60 points
  - One Point → approx. one minute per point
  - Don‘t waste time if you don‘t know the answer
    - You don‘t need all points for very good grades!
- We don’t want long texts
- If you know you won’t write the exam:
  It makes our lives easier if you cancel beforehand in campus-system

ML1 Exam of SS-2023

- We let GPT-4 take the exam
- Let’s see how it did!

Good Luck for the Exam!