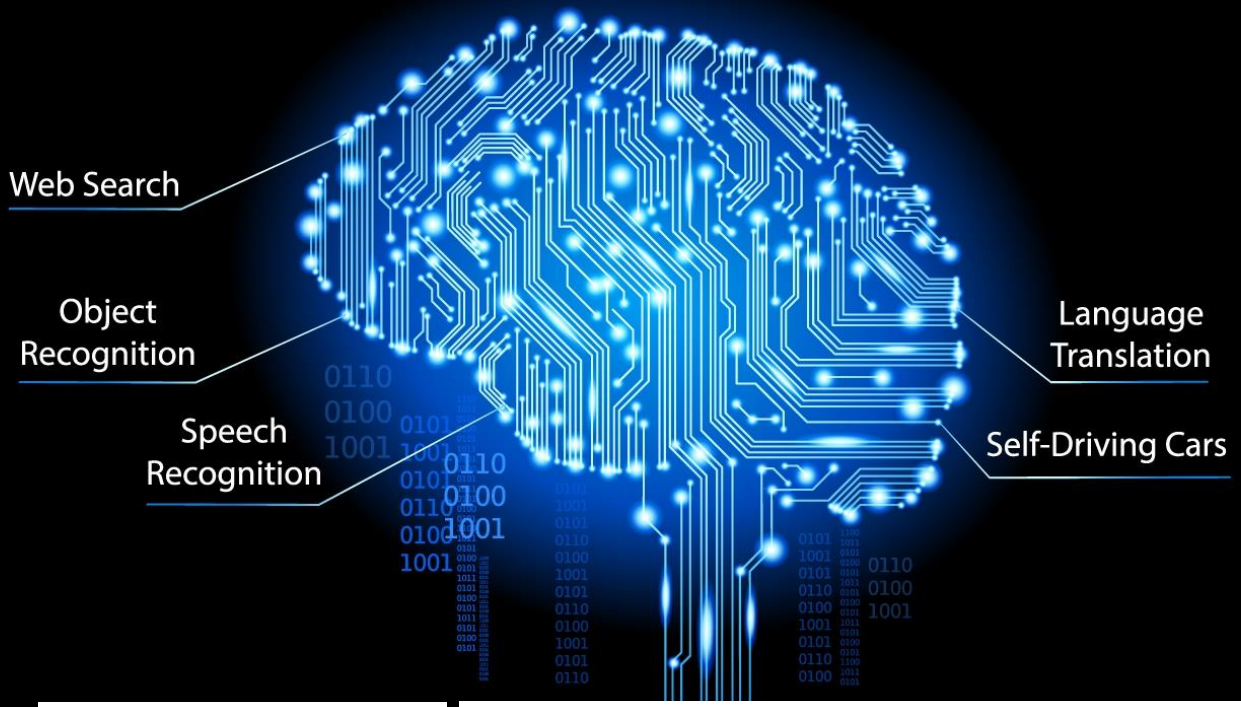


b+m Informatik

KI & ML – Ein Blick unter die Motorhaube

b+m Engineering

Digitale Woche 2020

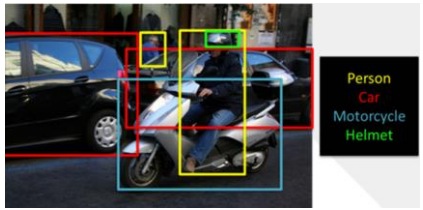
Agenda

Die drei großen Bereiche des Machine Learning –
dazu (Live) Beispiele, sowie Erläuterung der Funktionsweise

1. Supervised Learning: Image Classification mit Convolutional Neural Nets
2. Unsupervised Learning: Anomalieerkennung/Fraud Detection
3. Reinforcement Learning: Deep Q-Learning

AI-Durchbrüche auf einigen Anwendungsfeldern

Image Processing
Convolutional Neural Networks (CNN)



Generative Adverserial Networks (GAN)

Natural Language Processing
Recurrent Neural Networks (RNN/LSTM)

Google Assistant: Haircut Appointment Call

Facebook: Alice & Bob Experiment

DeepL

Intelligent Systems
Reinforcement Learning

Google Deepmind Runner

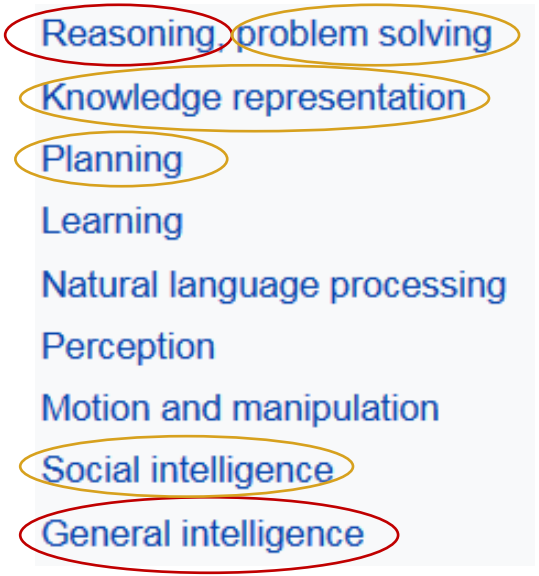
Boston Dynamics Atlas

Games
Reinforcement Learning

Mögliche Go Positionen: $\sim 10^{170}$
Atome im sichtb. Universum: $\sim 10^{80}$

AlphaGo
AlphaGo Zero

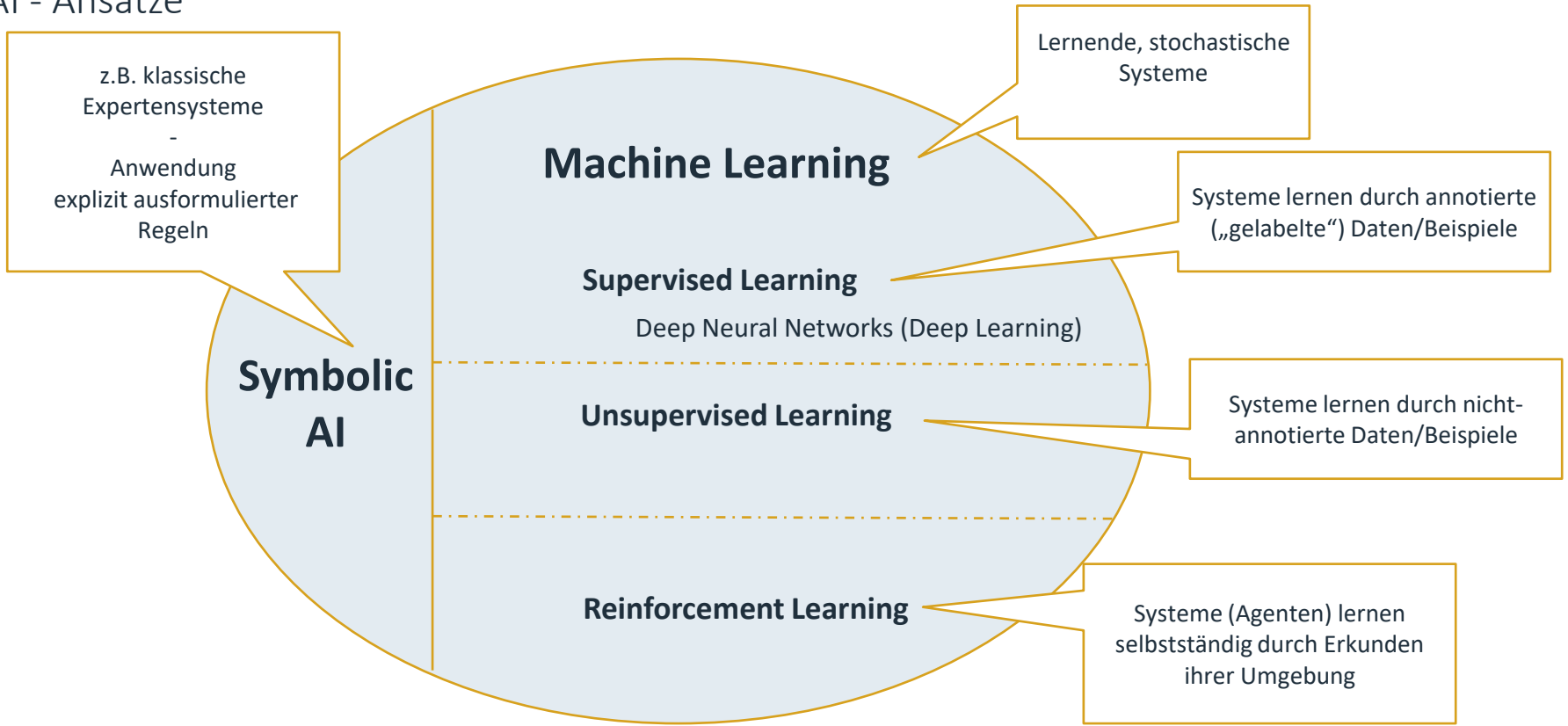
AI – Definition über Forschungs- bzw. Anwendungsfelder



Spitzenforschung

Forschung in den Kinderschuhen

AI - Ansätze



AI Landscape – Big Picture

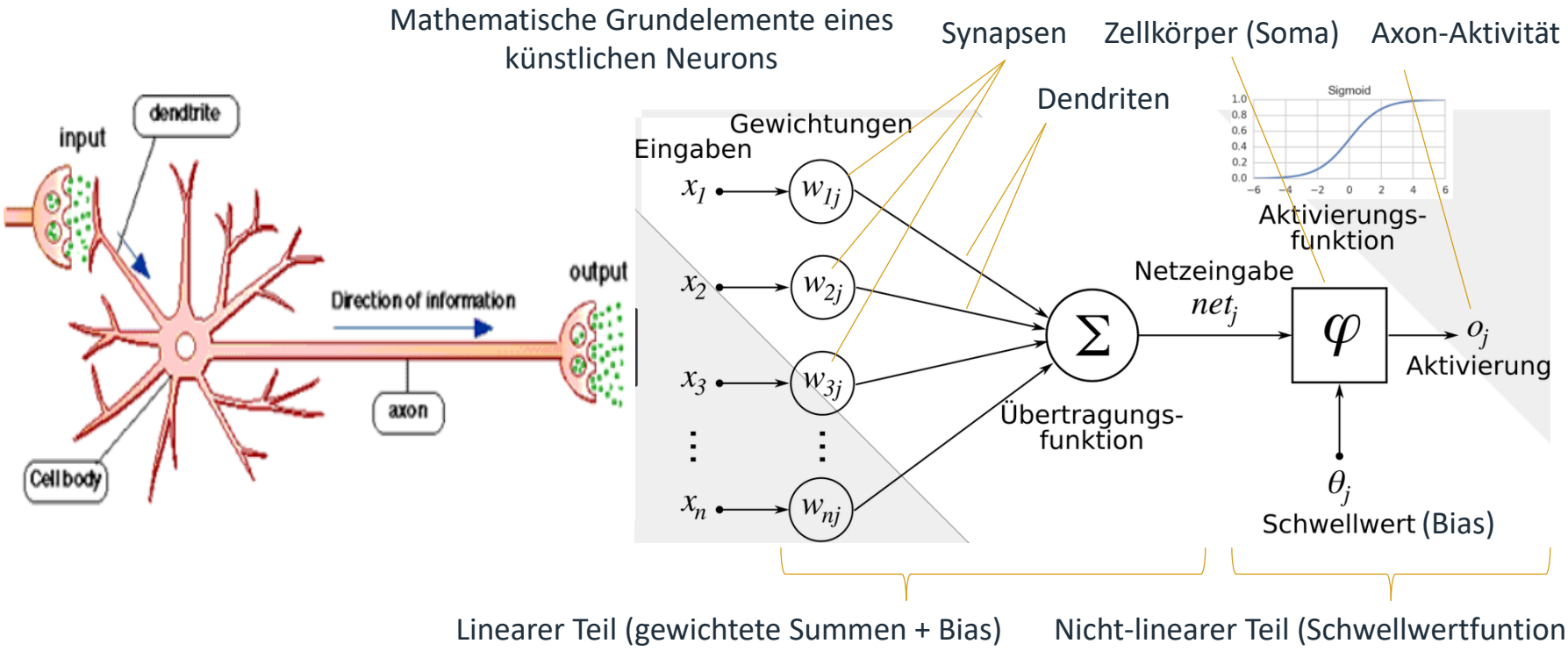
Agenda

Die drei großen Bereiche des Machine Learning –
dazu (Live) Beispiele, sowie Erläuterung der Funktionsweise

1. Supervised Learning: Image Classification mit Convolutional Neural Nets
2. Unsupervised Learning: Anomalieerkennung / Fraud Detection
3. Reinforcement Learning: Deep Q-Learning

<Live Demo>

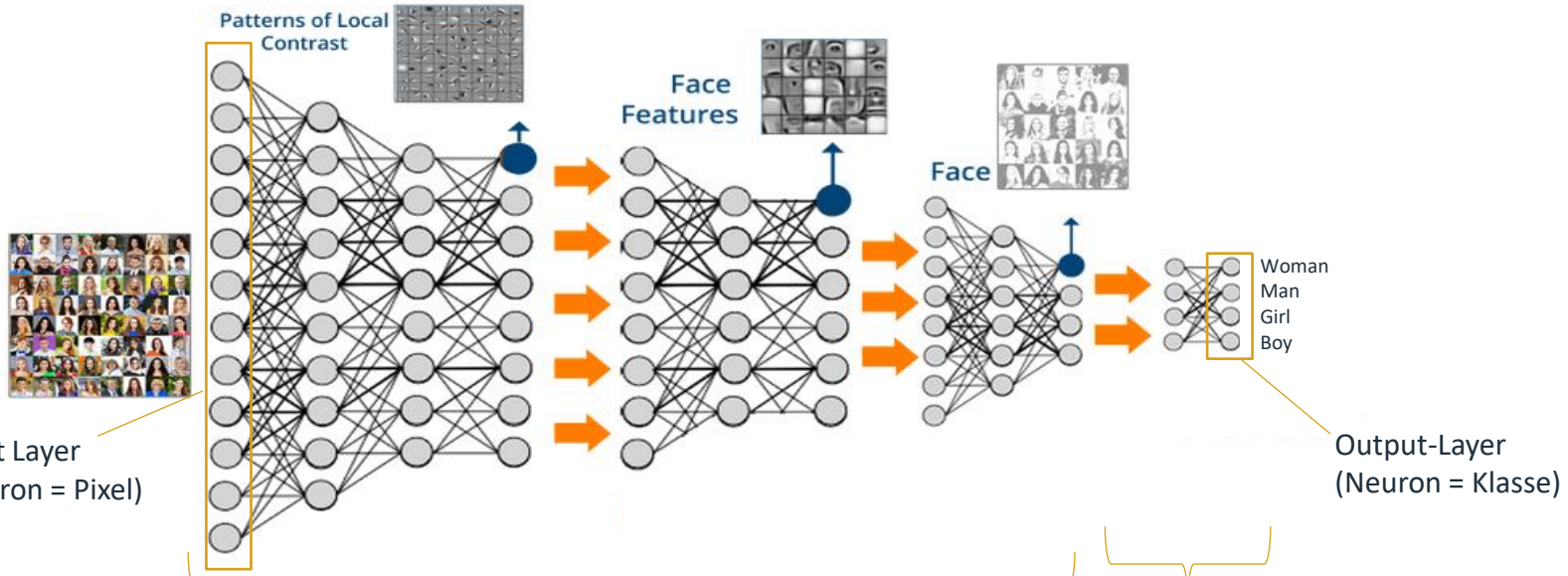
ANN – Einführung: Mathematische Modellierung eines Neurons



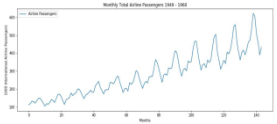
Convolutional Neural Nets – Beispiel: Face Classification

Classification Output

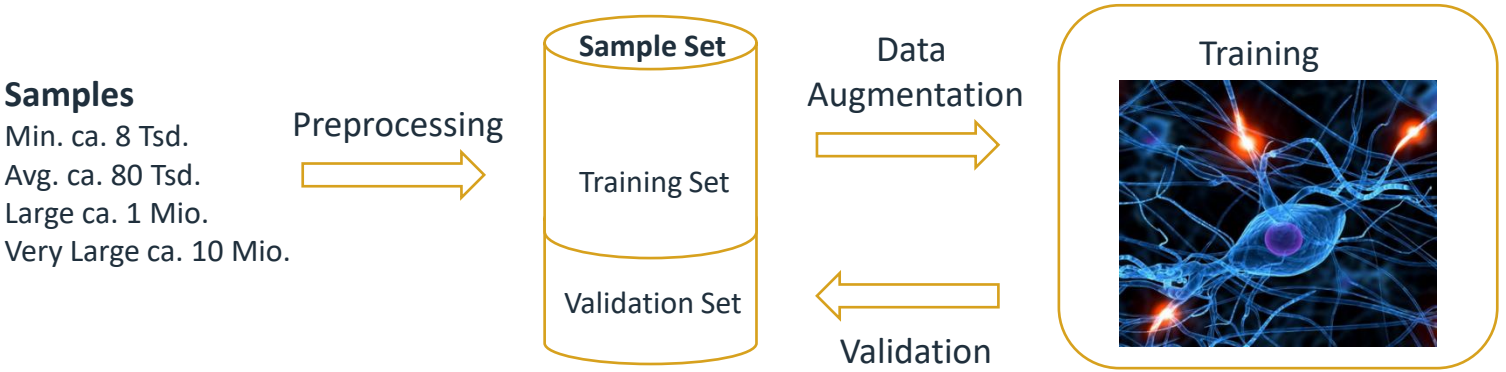
Woman:	85%
Girl:	14,5%
Boy:	0,5%
Man:	0%



Supervised Learning – Training and Validation: Samples

	Input	**Target (Output)**
Image Classification Sample		„Jaguar E-Type Serie-1 Roadster“
Semantic Text Analysis Sample (Topic + Sentiment Analysis)		Beschwerde, Schadenmeldung; „empört“
Data Based Prediction Sample	**Adresse: Schlossallee 1** **Liquidität: 50 TEUR** **Mahnungen: 10** **Mtl. Ausgaben: 3 TEUR** **…**	Kreditausfall nach 3 Jahren
Time Series Prediction Sample	 **Passagierzahlen KW1-6 2018**	Passagierzahlen KW7 2018

Supervised Learning – Training and Validation: Sample Sets



Agenda

Die drei großen Bereiche des Machine Learning –
dazu (Live) Beispiele, sowie Erläuterung der Funktionsweise

1. Supervised Learning: Image Classification mit Convolutional Neural Nets
2. Unsupervised Learning: Anomalieerkennung / Fraud Detection
3. Reinforcement Learning: Deep Q-Learning

Data-Based Unsupervised Learning - Charakteristika

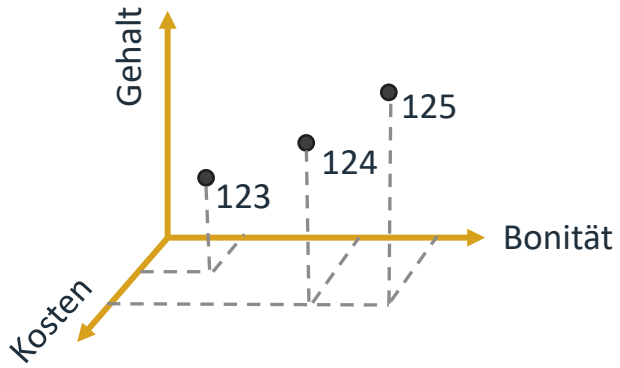
- Samples bestehen nur aus dem Input – es gibt keine Labels (Targets) (d.h. keine Zielvorgabe/Soll-Output).
- Das ML-Modell muss selbstständig Strukturen und Zusammenhänge in den Daten erkennen/lernen.
- Supervised Learning ist in der Regel genauer und führt direkter zum Ziel, dafür kann Unsupervised Learning verwendet werden, wenn das genaue Lernziel a priori noch unbekannt ist (Data-Mining), oder das Labeling der Trainingsdaten zu teuer/aufwändig.
  - Spezialfall Anomalieerkennung: Oftmals sind (fast) nur Daten verfügbar, die den Normalfall repräsentieren (Bsp. Server-Logs oder Bank-Transaktionen).
    Das Modell muss also lernen, diesen Datenraum zu „verstehen“.
    Anomalien sind dann (neue/unbekannte) Daten, die diesem Verständnis nicht entsprechen.

AI/ML Grundlagen – Datenräume und Dimensionen

Beispiel: Vereinfachter Kreditrating Datensatz
Identifikator + 3 Attribute

KdNr	Bonität	Gehalt	Kosten
123	5000	2000	1500
124	25000	2500	2000
125	30000	3000	2000

3dim Datenraum (math.: Vektorraum)



- Spalten/Tabellenüberschriften sind Dimensionen im Datenraum
- Eine Zeile/einzelner Datensatz ist ein Punkt (Vektor) im Datenraum
- Der Wert einer Zelle ist ein Wert auf der entsprechenden Achse im Datenraum (Vektor-Komponente)

In der Praxis entstehen schnell hochdimensionale Datenräume, die daher nicht 1:1 visualisiert werden können

ShowCase: Anomalieerkennung mit Parametric t-SNE und Clustering

Aufgabe:
Erkennung von Anomalien (Betrugsfälle bei Kreditkarten-Transaktionen) mittels Unsupervised Learning - Das Modell wird ausschließlich mit normalen Tx (d.h. Nicht-Betrugsfällen) trainiert und muss lernen, diesen Datenraum zu „verstehen“. Dieser ist **28-dimensional** und enthält 10000 normale Transaktionen

Modell:
Ein Dimensionsreduzierer (28 -> 3), dem ein neuronales Netz „injiziert“ wurde. Somit können auch neue, dem Modell nicht bekannte Daten verarbeitet werden.

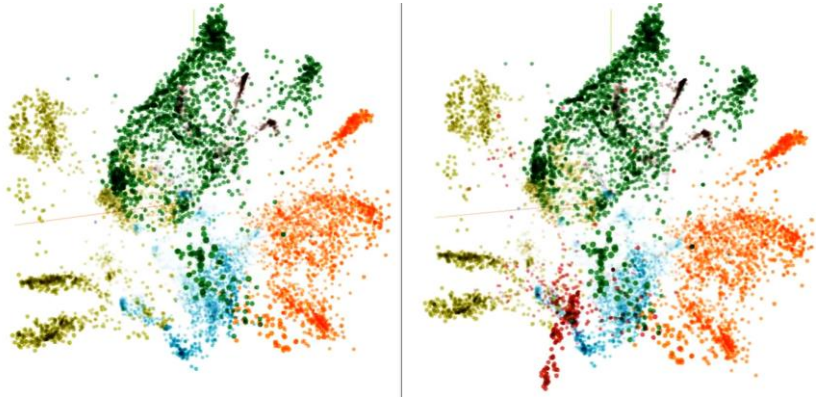
<Animation>

ShowCase: Anomalieerkennung mit Parametric t-SNE und Clustering

Aufgabe:
Erkennung von Anomalien (Betrugsfälle bei Kreditkarten-Transaktionen) mittels Unsupervised Learning - Das Modell wird ausschließlich mit normalen Tx (d.h. Nicht-Betrugsfällen) trainiert und muss lernen, diesen Datenraum zu „verstehen“. Dieser ist 28-dimensional und enthält 10000 normale Transaktionen

Modell:
Ein Dimensionsreduzierer (28 -> 3), dem ein neuronales Netz „injiziert“ wurde. Somit können auch neue, dem Modell nicht bekannte Daten verarbeitet werden.

Ergebnis:
Der trainierte Normalfall enthält 5 Cluster.
Die Betrugsfälle bilden ein neues Cluster, das mathematisch separiert werden kann.
Somit ist eine Prognose für zukünftige Transaktionen möglich.



Metriken:

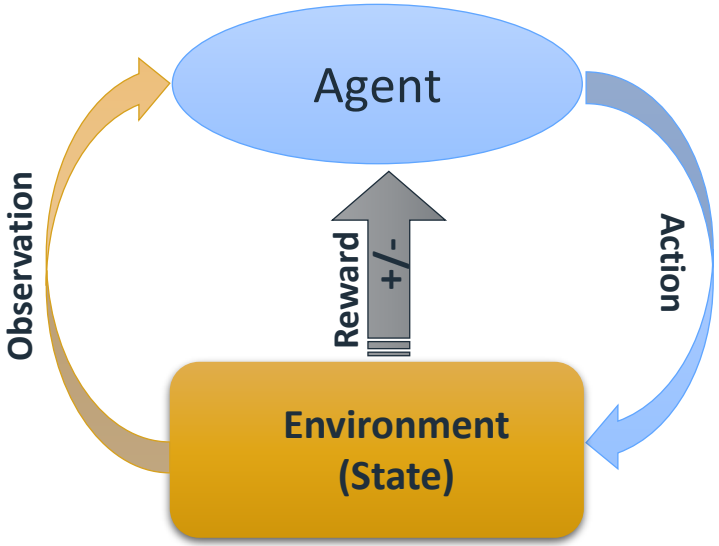
train recall (false negatives critical)	= 97.13%
validation recall (false negatives critical)	= 96.91%
validation accuracy (true positives/negatives critical)	= 93.10%
validation precision (false positives critical)	= 94.00%
validation f1-score (balanced recall and precision)	= 95.43%

Agenda

Die drei großen Bereiche des Machine Learning –
dazu (Live) Beispiele, sowie Erläuterung der Funktionsweise

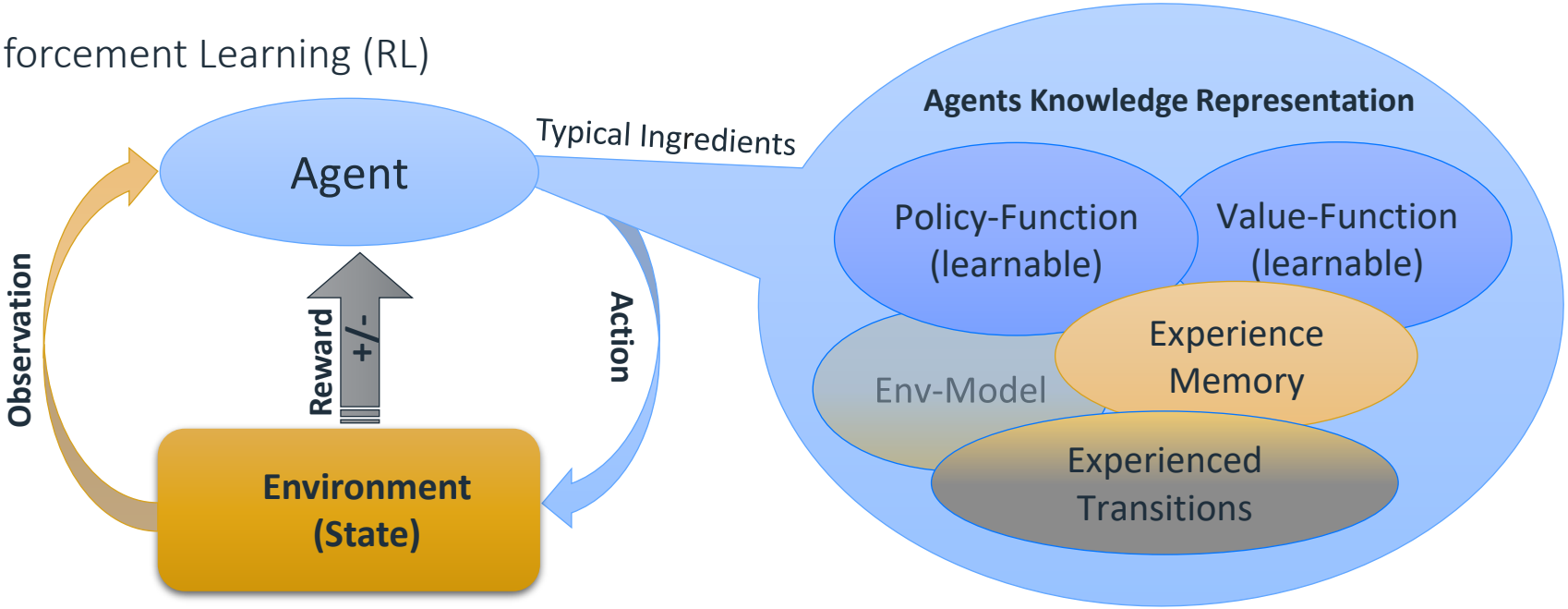
1. Supervised Learning: Image Classification mit Convolutional Neural Nets
2. Unsupervised Learning: Anomalieerkennung / Fraud Detection
3. Reinforcement Learning: Deep Q-Learning

Reinforcement Learning (RL)



Observation: (Observable part of) current state
Reward: Measure of „what is good“ or „bad“ (given by design)

Reinforcement Learning (RL)

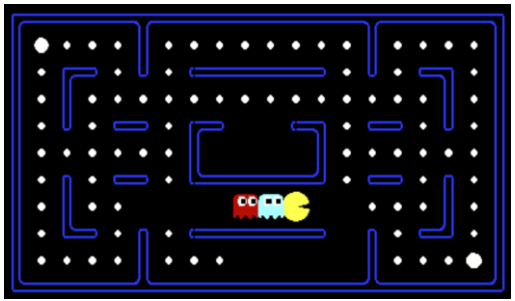


Policy: What to do next (learnable)
Value: Rating of a state based on reward prediction (learnable)
Reward: Measure of „what is good“ or „bad“ (given by design)
Experience: Past reaction of environment in a state to an action: reward + next obs.
Env-Model: Agents representation of Environment (optional, given or learnable)

AI – Research Benchmarks: Games

Board Games (tief aber nicht visuell)

Computer Games (visuell reichhaltig)



?

Real World UseCases

Adobe Stock Datei-Nr.: 162980569 | Urheber: metamorworks

- Autonomes Fahren
- Robotics
- Steuerung von Industrieanlagen
- Intelligente Verkehrsleitung
- Geschäftsprozessoptimierung
- Proteinfaltung
- …

Reinforcement Learning Beispiel – Deep Q-Learning (DQN) mit Atari-Games

- Keine Information über Spielmechanik/Ziel
- Keine taktischen oder strategischen Informationen
- Einzige Aufgabe: Reward-Maximierung (langfristig)

Agent

Pixelstrom (Frames)

Observation

Pos. Reward = Score

Neg. Reward = Live loss

Action

Environment (Emulator)

RL / DQN – Atari Games

Herausforderungen / Setting

- Der Agent bekommt ausschließlich den Pixelstrom (Bildfolge), den Score (**positiver Reward**) sowie das „Life lost“-Ereignis (**negativer Reward**) als Input.
- Der Agent hat 6 Aktionen zur Verfügung:
  Stehen, Stehen+Feuer, Rechts, Links, Rechts+Feuer, Links+Feuer.
- Der Agent hat also initial keinerlei „Vorstellung“ worum es in dem Spiel überhaupt geht und welche Wirkung die Aktionen haben oder welche Regeln gelten. Erst recht besitzt er keinerlei taktische oder gar strategische Information.
- Seine Aufgabe besteht schlicht in der Maximierung des (langfristigen) Rewards.
- Dieses Spiel kann nicht gewonnen werden (aber verloren).

RL / DQN – Atari Games

<Live Demo>

RL / DQN – Beispiel Atari Games

Der Agent lernt im Lauf der Zeit, etwa in dieser Reihenfolge, selbstständig taktische Elemente des Spiels - nämlich dass

- die „Feuer“-Aktionen irgendwann später oftmals einen positiven Reward ergeben
  => Dauerfeuer-Strategie
- weiße Pixelstriche, die von oben kommen „tödlich“ sind (Invader Schüsse)
- Invader-Schüssen mit Aktionen wie „Links“ oder „Rechts“ ausgewichen werden kann.
  Somit hat der Agent ein Verständnis erlangt, dass das Raumschiff den Spieler repräsentiert.
- weiße Pixelstriche, die von unten nach oben laufen, durch „Feuer“ Aktionen ausgelöst werden und ursächlich für Treffer sind (positiver Reward)
- UfOs (Mutterschiffe) mehr Reward erbringen
- spaltenweises Dauerfeuer Reward-Ketten erzeugen kann
- Barrieren (eigenes und Invader-)Feuer abhalten, aber sukzessive zerstört werden
- mit einem Vorlauf in Abhängigkeit zur Entfernung auf Invader gezielt werden kann
- die Aktion „Feuer“ blockiert ist, solange noch ein eigener Schuss zu sehen ist

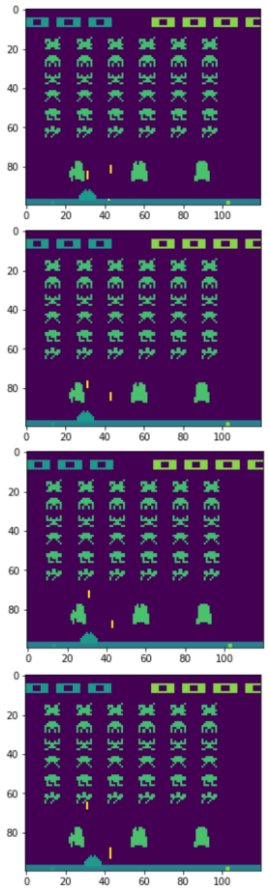
RL / DQN – Beispiel Atari Games

Das Spiel enthält strategische Komponenten, die erst dann zu Tage treten, wenn der Agent ein gewisses Niveau erreicht hat:

- Erreicht ein Invader den „Boden“, ist das komplette Spiel verloren. Dies ist eine zunächst extrem seltene aber sehr wichtige Erfahrung
  (*-> Prioritized Experience Replay Memory)*
- Dies hat zur Folge, dass der Agent eine langfristige Strategie entwickeln muss und nicht einfach „spaltenweise“ vorgehen oder „trödeln“ kann.
- Eine Dauerfeuer-Strategie ist nicht optimal, gerade in der Endphase einer Episode (wegen der Feuer-Blockierung)
- Die Angriffswellen werden pro Episode immer schneller und immer tiefer – das erfordert visuelle Generalisierung

RL/DQN – Policy Function Construction

Input:
Observation from Emulator
Pixel stream 120x100
(1 Observation
= 4 consecutive frames)



Detection of movement direction possible

Policy-Function
=
Convolutional Neural Net (CNN)

Supervised Learning from Experience Memory + Bellman Equation

Output:
Next Best Action
Q-Value „distribution“

Action	Q-Value
None	0.042
Fire	0.132
Left	0.057
Right	0.007
Left+Fire	0.164
Right+Fire	0.018

Agent choses action with highest Q-value

RL Fundamentals

Markov Property for Markov Decision Processes (MDP)
"The future is independent of the past given the present":

$$P[s_{t+1}|s_t] = P[s_{t+1}|s_1, \ldots\ldots, s_t]$$

Bellman Equation:

$$Q^*(s,a) = \mathbb{E}\left[r_{t+1} + \gamma \max_{a'} Q^*(s_{t+1}, a') \middle| s_t = s, a_t = a\right]$$

Used for calculation of new target

where

s: State, a: Action, r: Reward, γ: Discount factor, Q^*: Optimal state action Q-value function

RL / DQN – Summary

Ziel dieses Atari DQN-Experiments war es nicht einen neuen High Score aufzustellen, sondern:

- Lernen und tiefgreifendes Verständnis erlangen (Mathematik *und* softwaretechnische Umsetzung)
  ->
  Der Code subsummiert Ansätze diverser wissenschaftlicher Paper der letzten Jahre.
- Schaffung einer Code-Basis für eigene konzeptionelle und technische Verbesserungen des Deep Q-Learning. Es wurden eigene Innovationen eingebracht.
  ->
  insb. Maßnahmen zur Verkürzung der Lernphase und Reduktion der benötigten Datenmenge
- Schaffung einer Basis (Code + Know-how) für praxisrelevante UseCases und etwaige Forschungsprojekte.
  ->
  Der entstandene Code ist vollständig generisch - d.h. auf ganz andere Domänen als (Atari-)Spiele übertragbar.

Wesentliche Herausforderungen bei der Übertragung von RL-Algorithmen auf praxisrelevante UseCases:

- Auswahl des „richtigen“ Algorithmus bzw. seiner Derivate
- Generierung ausreichend großer Datenmengen für das Training
- Design der Reward- (bzw. Utility-)Function

b+m Informatik AG
Rotenhofer Weg 20
24109 Melsdorf

www.bmiag.de

Ihr Ansprechpartner

Thomas Stahl
CTO
Head of AI-Lab

thomas.stahl@bmiag.de