

# TECHNICAL APPLICATIONS AND DATA MANAGEMENT. SS 2023.

## VORLESUNG 12

13.06.2023

MÜNCHEN

STUDIENGANG  
SUSTAINABILITY  
MANAGEMENT.



## AGENDA

1. Status Projektarbeit
2. Ausblick
  1. Machine Learning in der Cloud
  2. Reinforcement Learning

## GEPLANTE ROADMAP DER VORLESUNG.

ROADMAP	WAS HABEN WIR VOR?
Vorlesung 1	Übersicht und Einführung
Vorlesung 2	Einführung Data Science und Data Science Workflow, Detaillierung Data Engineering
Vorlesung 3 und Vorlesung 4	Deskriptive und explorative Datenanalyse und Vertiefung anhand Case Study
	Vertiefung Datenanalyse anhand Case Study
Vorlesung 5	Aufgabenstellung Data Science, Übersicht und Einführung Machine Learning, unüberwachtes Lernen
Vorlesung 6 und Vorlesung 7	Überwachtes Lernen
	Vertiefung überwachtes Lernen anhand Case Study
Vorlesung 8 und Vorlesung 9	Neuronale Netze und Convolutional Neural Networks (CNN)
	Vertiefung CNN anhand Case Study, Aufgabenstellung AI
Vorlesung 10	Rekurrente Neuronale Netze
Vorlesung 11	Generative AI
Vorlesung 12	Ausblick
Vorlesung 13	„Fragestunde“

Folien der bisherigen Vorlesung verfügbar unter [Link](#)



# 1. STATUS PROJEKTARBEIT

# ÜBERSICHT ANGEBOTENE THEMEN FÜR DIE HAUSARBEIT.

## Teil 1: Data Science

- Boston Housing pricing ([Link](#))
- Telecom Customer Churn ([Link](#))
- Customer Personality Analysis ([Link](#))
- Stroke Prediction ([Link](#))
- Heart Attack prediction ([Link](#))
- Schwer: Life Expectancy ([Link](#))



## Teil 2: Machine Learning

- **Bilderanalyse:** Einsatz und Vergleich von Basic CNN und Transfer Learning inkl. Parametertuning
  - Erkennen von Lungenentzündung ([Link](#))
  - Verkehrszeichen erkennen ([Link](#))
  - Erkennen von Müdigkeit ([Link](#))
  - Tierartenerkennung: Erkennung von Katzen... ([Link](#))
- **Text Analyse:** Einsatz und Vergleich von Basic RNN und LSTM, Attention oder Bert inkl. Parametertuning
  - Fake News Detection/ Classifier ([Link](#))
  - Sentiment Analysis of Movies ([Link](#))

Für jede der Aufgaben finden Sie unter dem Link detaillierte Infos. Zusätzlich werde ich eine Aufgabenstellung bereitstellen. Ihre Prüfungsleistung ist Notebook und Ausarbeitung(~5 Seiten) für Teil 1 und Teil 2. Templates für Ausarbeitung finden Sie auf Ilias.

## 2. AUSBLICK



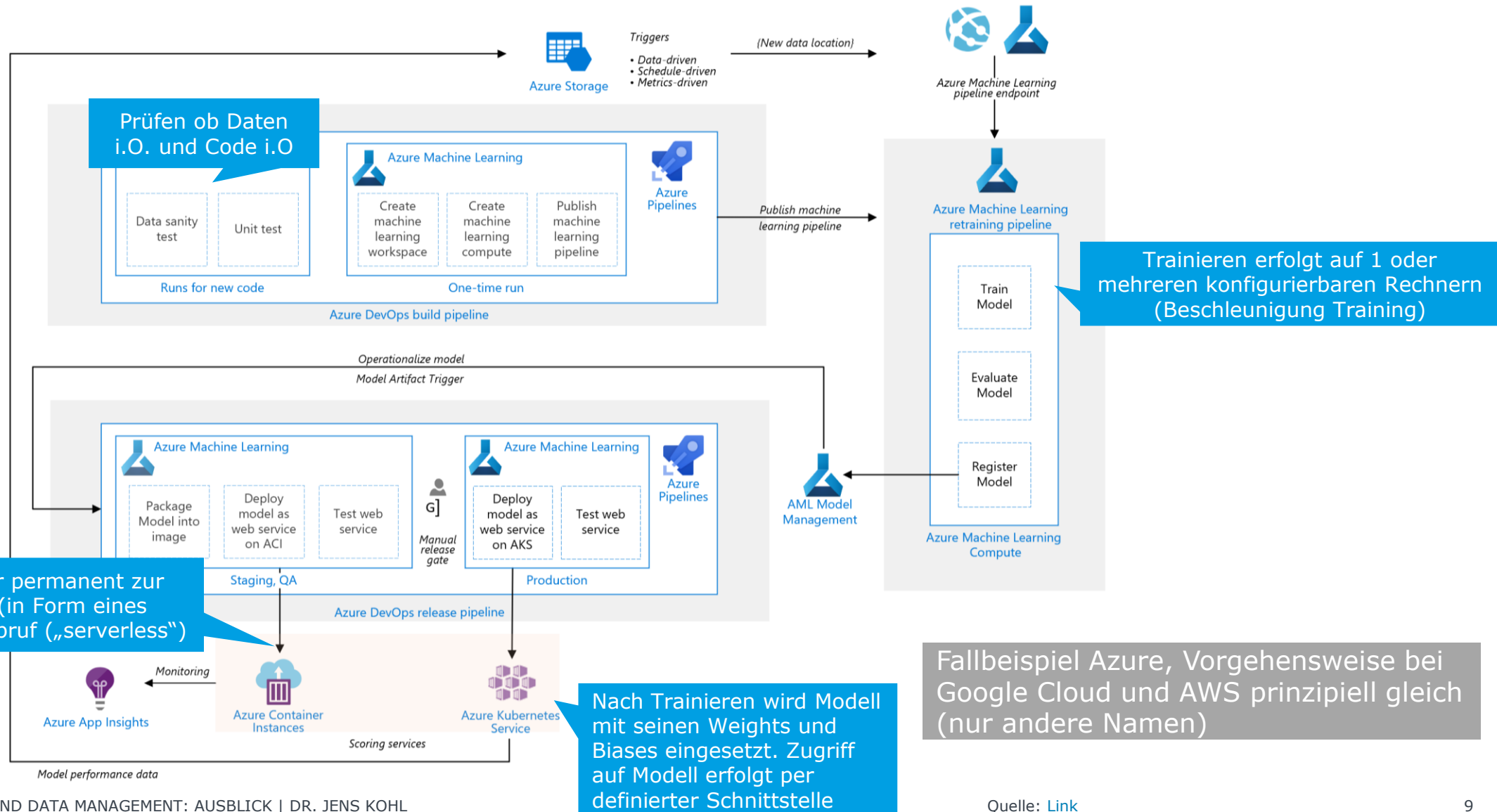
## 2.1 AI IN DER CLOUD

# EINLEITUNG.

- Bis jetzt haben wir vorwiegend auf **einem** Rechner (lokal/ Cloud) unsere Use Cases umgesetzt.
  - Wir haben gesehen, daß Machine Learning sehr ressourcen- und zeitintensiv ist:
    - Speicherbedarf: je mehr Daten, desto bessere Ergebnisse<sup>1</sup> (Datenmenge x 2 oder 3 besser als jedes Parameter-Tunen!!).
    - Rechenbedarf: je länger trainiert wird, desto (meist) besser sind die Ergebnisse. Achtung: Overfitting!
  - Größter Rechenaufwand beim Trainieren, anschließend deutlich geringerer Aufwand für Nutzung Modell (und nur auf Bedarf)
- ➔ Wirtschaftliche/ ökologische Vorteile bei Nutzung Cloud-Anbieter (on-demand) statt Aufbau eigener Strukturen (on-premise).
- ➔ Auch Startups können ohne große Investitionen „gleiche“ Ressourcen wie große Firmen nutzen.



# ÜBERSICHT ARCHITEKTUR MACHINE LEARNING IN DER CLOUD.



## AUSBLICK.

- Starker Wettbewerbsdruck zwischen Amazon Web Services (erster Anbieter 2004/2006!<sup>1</sup>), Microsoft Azure und Google Cloud.
- Etablierte IT-Anbieter werden in das Cloud-Geschäft einsteigen: SAP, Oracle, ....
- Dieser steigende Wettbewerb führt zu:
  - Sinkenden Preisen.
  - Firmen: Unterstützungsleistungen bei Einführung der Cloud.
  - Privatanwender: freie Kontingente für Ausprobieren/ erste Schritte (AWS 300\$, 200\$ Azure, 300\$ Google).
  - Allgemein: Entwicklung von automatisierten ML-Ansätzen (AutoML), Bereitstellen von Standardlösungen, Tutorials, ...

In dieser Vorlesung haben Sie die Grundlagen gelernt; schauen Sie sich doch die Tutorials an!

## 2.2 REINFORCEMENT LEARNING

# ÜBERSICHT/ CLUSTERING MACHINE LEARNING ALGORITHMEN.

## Unsupervised Learning

Lernen **ohne** vorher definierte **Zielwerte** oder Belohnung

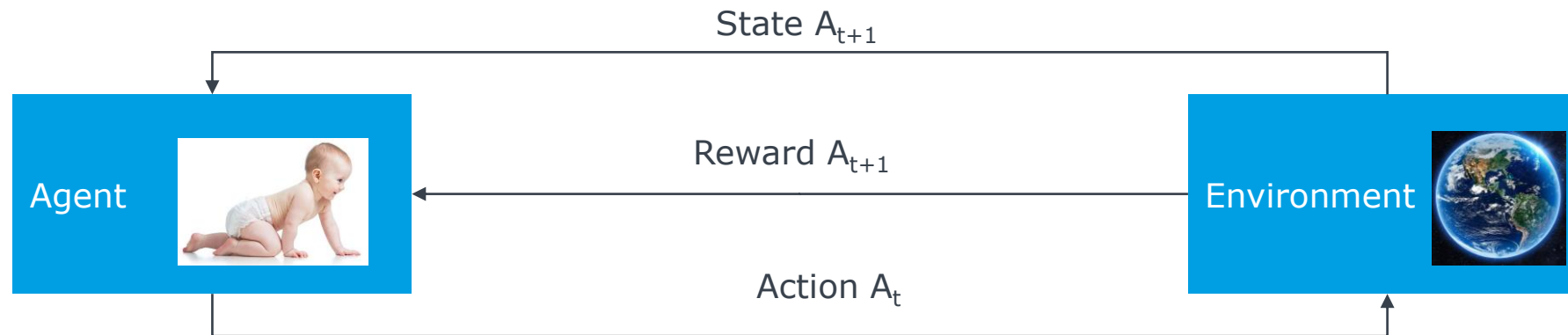
## Supervised Learning

Algorithmus lernt eine Funktion, die Eingabegrößen auf **vorher** definierte Outputs mappt.

## Reinforcement Learning

Agent/ Algorithmus lernt **selbständig** mit Ziel, eine Belohnung zu **maximieren**.

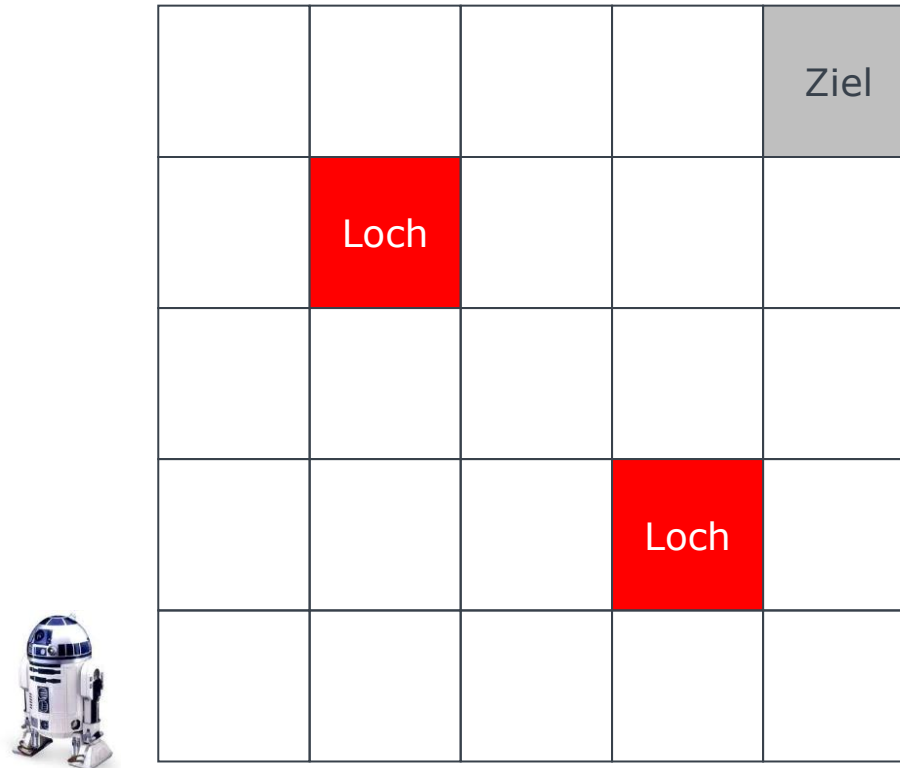
# VEREINFACHTE DARSTELLUNG REINFORCEMENT LEARNING.



## Unterschiede zu bisherigen Verfahren:

- Algorithmus agiert, um Belohnung zu maximieren und nimmt somit Einfluß auf seine Umgebung (und damit auf die nachfolgenden Daten).
- Agent weiß nicht (immer), wie die Umwelt auf seine Aktionen reagiert.
- Diese Belohnung erfolgt nicht zwingend per direktem Feedback, sondern auch ggf. später.
- Deshalb immer zeitabhängige, sequentielle Daten.
- Kann zur Laufzeit auch vorher nicht gelernte Sachen lernen (supervised Models sind statisch!).

# REINFORCEMENT LEARNING: FALLBEISPIEL WEGFINDEN ROBOTER.



Ziel: Gegeben eine zufällige Startposition des Roboters, finde das Ziel mit der geringsten Anzahl von Schritten.

# FORMALE GRUNDLAGE VON REINFORCEMENT LEARNING SIND MARKOV DECISION PROCESSES (MDP).

Eigenschaften:

- $S$  = Menge aller möglichen Zustände/ States  $s$
- $A$  = Menge aller Aktionen,
- $T(s, a, s')$  = Übergangsfunktion von einem Zustand  $s$  in nächsten  $s'$  durch Aktion  $a$ . Diese wird in Prozent angegeben, da Unsicherheit (keine vollständige Sicht auf die Umwelt!!)
- $R(s, a, s')$  = Belohnung für diesen o.a. Übergang.

Da wir mehrere Schritte haben können, gilt:

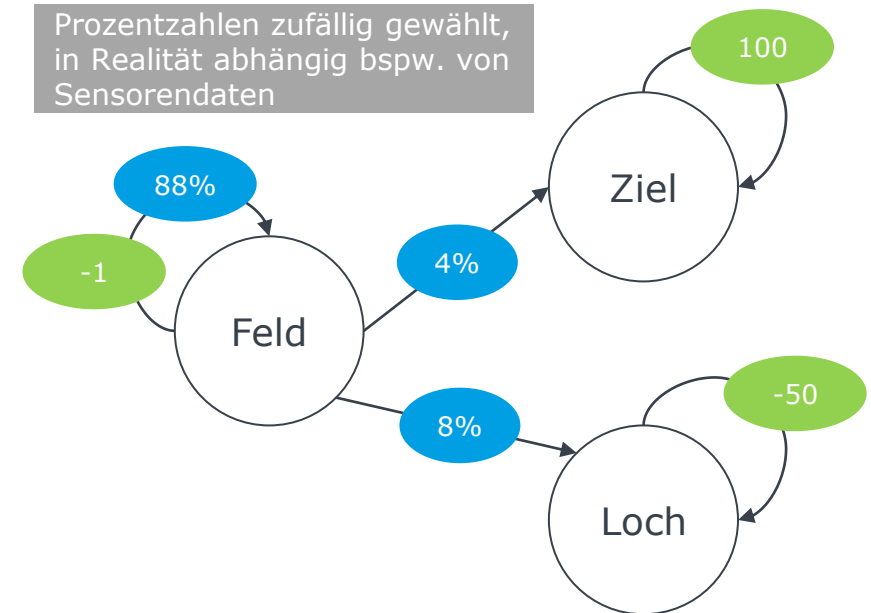
Aktuelle Belohnung

$$G_t = R_{t+1} + \gamma R_{t+2} + \dots = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}$$

- $V(s)$  = Wert eines Status.  $V(s) = \mathbb{E}[G | S_t = s]$

Gemittelte/ erwartete  
Belohnung im Zustand  $s$

Anschließende Belohnung inkl.  
Gewichtungsfaktor Wichtigkeit  
zukünftiger Schritte  $\gamma$   
(1:= sehr, 0:= gar nicht)



## Fürs Fallbeispiel:

Anzahl Felder =  $5 \times 5 = 25$

Zustände  $S := \{\text{Feld}, \text{Ziel}, \text{Loch}\}$

Aktionen  $A := \{\leftarrow, \uparrow, \rightarrow, \downarrow\}$

Übergangsfunktion  $T(\text{Feld}, *, \text{Ziel}) = 1/25 = 4\%$

Reward  $R(\text{Feld}, *, \text{Ziel}) = 100$ ,

Reward  $R(\text{Feld}, *, \text{Loch}) = -50$

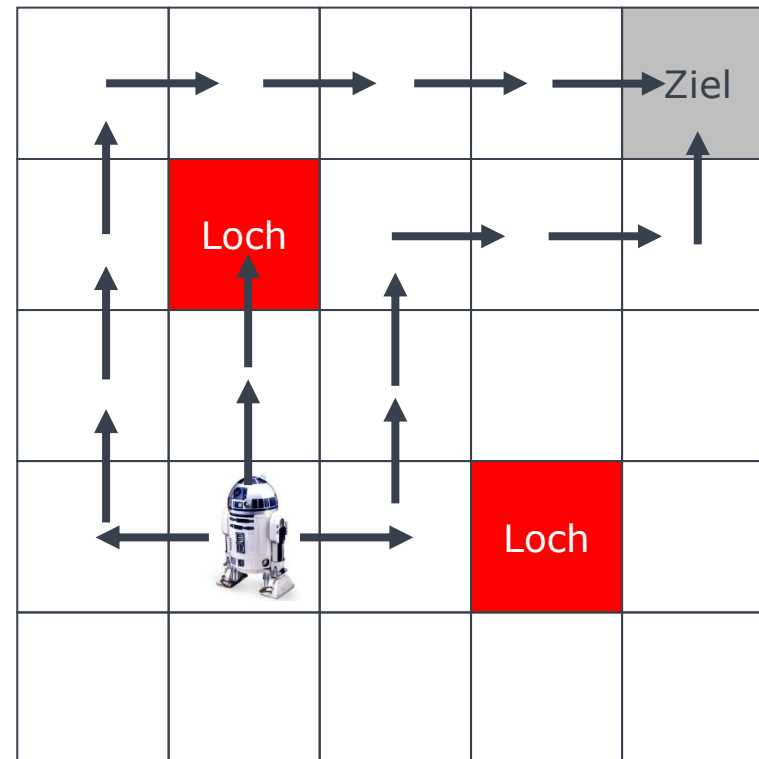
Reward  $R(\text{Feld}, *, \text{Feld}) = -1$  (Ziel: schneller Weg!)

$V(\text{Feld}) = -1 * 0,88$ ,  $V(\text{Ziel}) = 100 * 0,04 + -1 * 0,88$

Vereinfachte Annahme bei Markov-Prozessen: nur der aktuelle Zustand ist relevant für die Zukunft, nicht die vorherigen!

# REINFORCEMENT LEARNING: FALLBEISPIEL WEGFINDEN ROBOTER.

Welcher der 3 Wege  
ist der Beste?



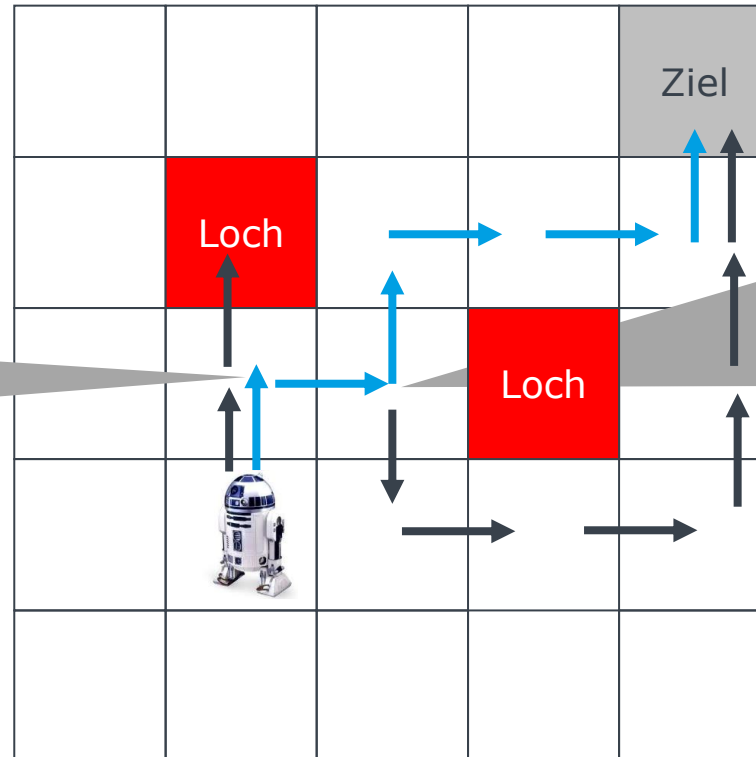
Ziel: Finde das Ziel mit der geringsten Anzahl von Schritten



# REINFORCEMENT LEARNING: FALLBEISPIEL WEGFINDEN ROBOTER.

Wie würde ein Mensch das lösen?  
Stets prüfen, was die beste Aktion ist-  
Aber wie trifft und lernt eine Maschine  
solche Eigenschaften?

Entscheidung RL Agent:  
Weg 1: führt ins Loch,  $R = -50$   
Weg 2: kein Loch



Entscheidung RL Agent:  
Blaue Weg: in weniger Schritten zum Ziel  
Schwarze Weg: mehr Schritte zum Ziel.

Wir hatten definiert, daß Schritte die nicht  
direkt zum Ziel führen, eine Belohnung von  
-1 erhalten.

Dadurch erhält der blaue, kürzere Weg eine  
höhere Belohnung als der schwarze und  
wird deshalb gewählt.

# FUNDAMENTALE PROBLEM DES REINFORCEMENT LEARNING: EXPLORATION VS. EXPLOITATION

## Exploration:

- Initiale Umgebung & Auswirkungen auf diese unbekannt
- Umwelt kann sich über Zeit ändern
- Agent muß explorieren, um Umwelt kennenzulernen. Das heißt, er führt eine Aktion aus, erfaßt die geänderte Umwelt und die Belohnung und speichert das ab.

## Exploitation

- Ziel des Agenten ist Maximieren Belohnung.
- Agent führt also die Aktion bzw. Sequenz von Aktionen aus, die ihm am meisten Belohnung zurückgibt.

Dilemma: Wenn der Agent nur Exploitation wählt, lernt er nie, ob es nicht bessere Aktionen gibt. Wenn der Agent nur exploriert, wird der Nutzen nicht maximiert und viele schlechte Aktionen ausgeführt.  
→ (langfristiger) Ausgleich notwendig.

Falls Sie immer in die gleichen Bars, Clubs, Restaurants gehen, lernen Sie nie besseres kennen. Andererseits vermeiden Sie so Reinfälle. Aber irgendwie müssen Sie die für Sie besten Läden ja auch mal kennengelernt haben....

Lösung Ausgleich Exploration vs. Exploitation durch Epsilon-Greedy<sup>1</sup>-Verfahren:

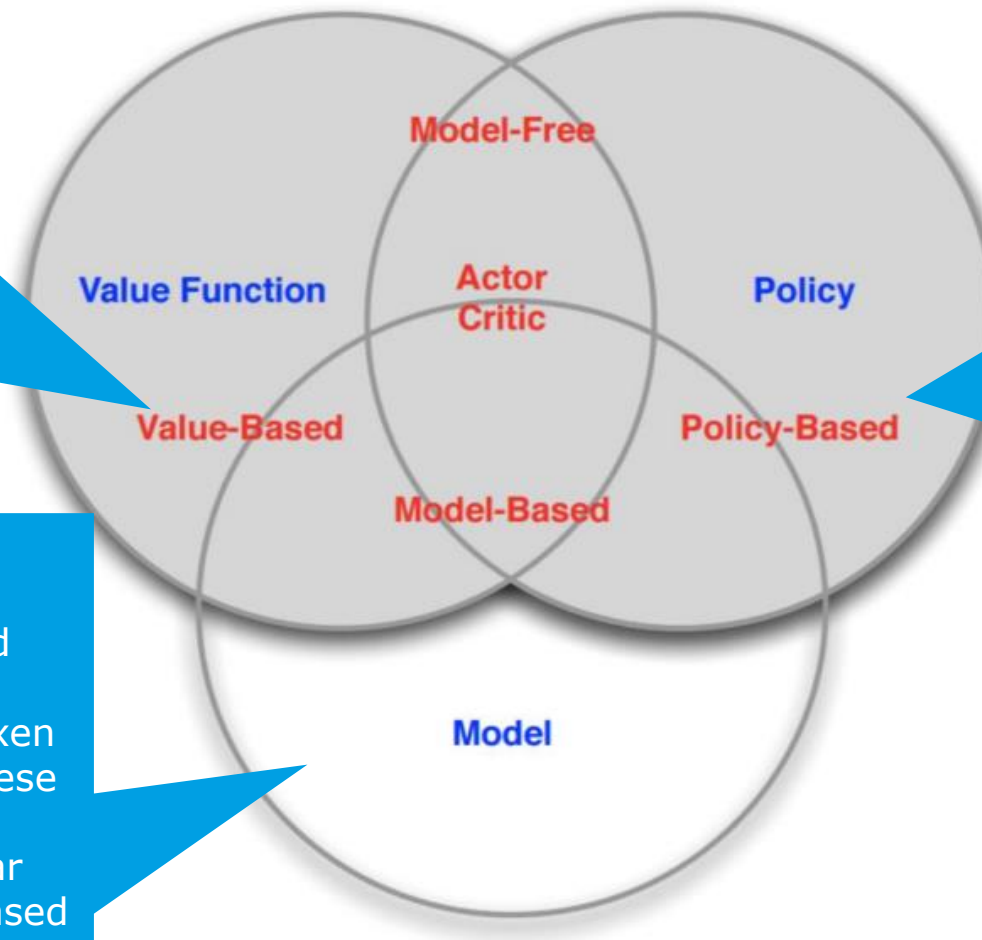
- Initiale Definition eines Wertes  $\epsilon$ , bspw. 0,1.
- Es wird vor jeder Aktion eine Zufallszahl  $p$  berechnet.
- Falls  $p < \epsilon$  dann beliebige Aktion ausführen, sonst beste Aktion (mit Wahrscheinlichkeit  $1 - \epsilon$ ).
- In der Praxis starten wir mit einem höheren  $\epsilon$ -Wert, der dann kontinuierlich verringert wird.

# ÜBERSICHT ENTSCHEIDUNGSVERFAHREN FÜR RL-AGENTEN.

- Iterative Berechnung der Value-Funktion mit höchstem Wert, dann indirektes Ableiten der Entscheidung für Aktion (Policy) aus dieser Value-Fkt. durch Wahl des Status mit höchstem Wert
- Vorteil: geringer Speicherbedarf, keine Tabelle notwendig.
- Nachteil: Aktionen nicht direkt ableitbar.

- Agent lernt Modell, wie Umwelt funktioniert basierend auf seinen Aktionen und plant darauf basierend seine Aktionen.
- Vorteil: kann auch mit sehr komplexen Umgebungen umgehen und lernt diese schneller.
- Nachteil: Modell kann schwer lernbar sein (und dann ist simples Policy-Based einfacher).

Detaillierung im folgenden



- Algorithmus lernt Tabelle mit Zusammenhang Status  $s$  und Aktion  $a$ .
- In jedem Zustand  $s$  wird aus der Tabelle die Aktion  $a$  gewählt, die die höchste Belohnung verspricht
- Vorteil: Aktionen direkt ableitbar
- Nachteil: sehr hoher Speicherbedarf für Tabelle bei komplexen Problemen

# FALLBEISPIELE MODEL-BASED REINFORCEMENT LEARNING.



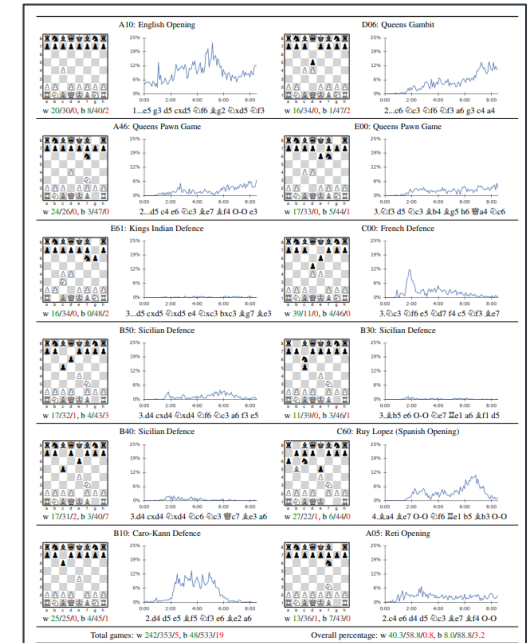
Mnih et al.: "Playing Atari with Deep Reinforcement Learning", 2013.

→ Erster Einsatz von Deep Learning Verfahren für Lernen des Modells (aus Sensorendaten).



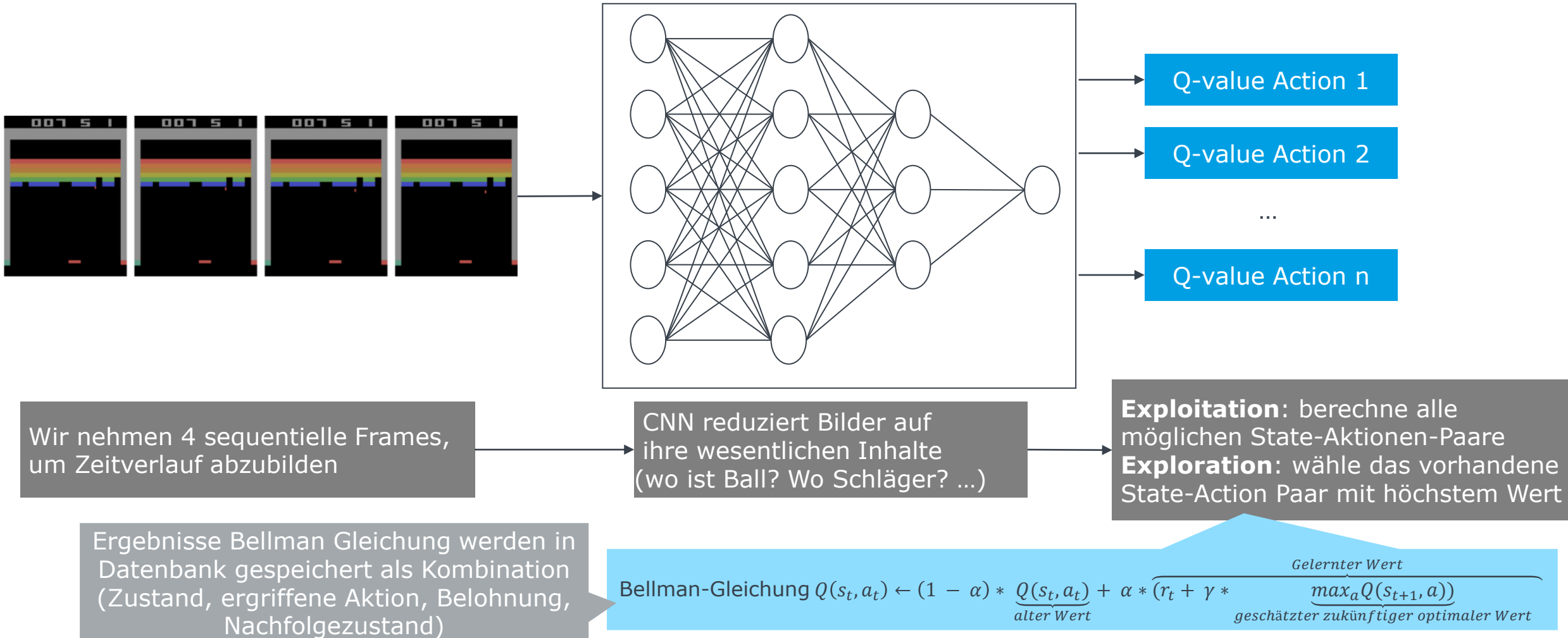
Silver et al.: "Mastering the game of Go with Deep Neural Networks & Tree Search", 2016.

→ Go galt aufgrund seines gigantischen Zustandsraums als nicht lernbar für Computer



Silver et al: "Mastering Chess and Shogi by Self-Play with a General Reinforcement Learning Algorithm", 2017.  
→ Weiterentwicklung Go-Ansatz, Modell lernt komplett eigenständig.

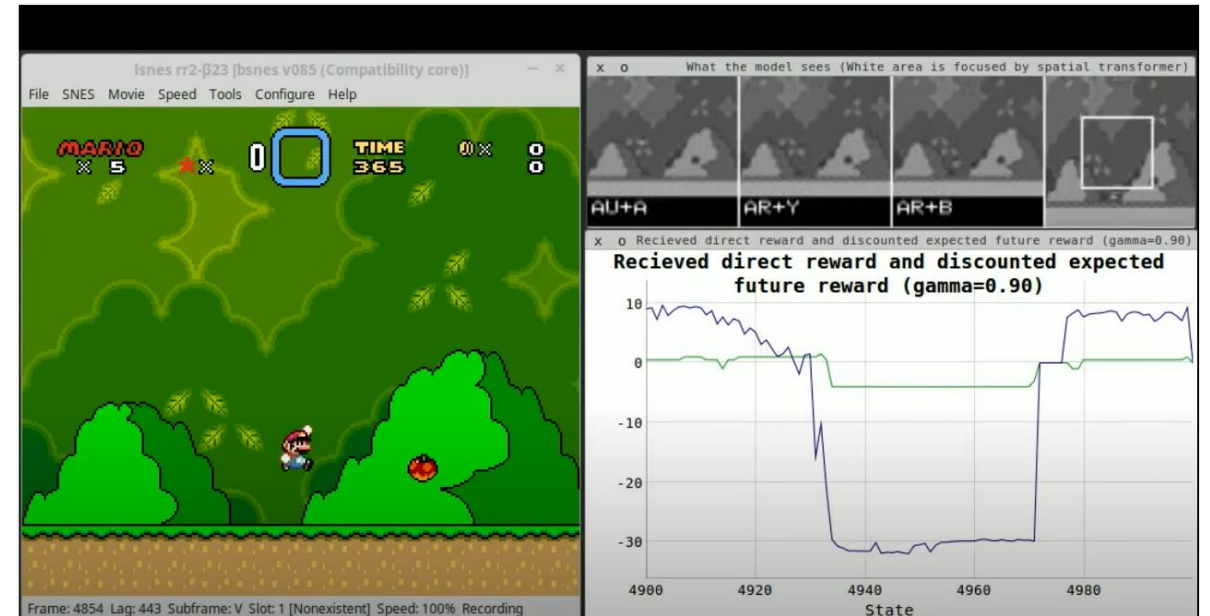
## WIE FUNKTIONIERT MODEL-BASED RL? (VEREINFACHT)



## LIVE DEMOS.



<https://www.youtube.com/watch?v=V1eYniJ0Rnk>



[https://www.youtube.com/watch?v=L4KBBawF\\_bE](https://www.youtube.com/watch?v=L4KBBawF_bE)

## LITERATUR UND WEITERE QUELLEN (AUSZUG).

### Künstliche Intelligenz:

- Russel, Norvig: Artificial Intelligence – a modern approach
- Lapan: Deep Reinforcement Learning Hands-on.
- Silver: Introduction to Reinforcement Learning ([Link](#))
- Barto, Sutton: Reinforcement Learning ([Link](#))

### Online-Kurse (bei Interesse):

- Kostenfrei: Udacity: Reinforcement Learning, [Link](#)
- Coursera: Reinforcement Learning, [Link](#)
- Udacity: Deep Reinforcement Learning, [Link](#)