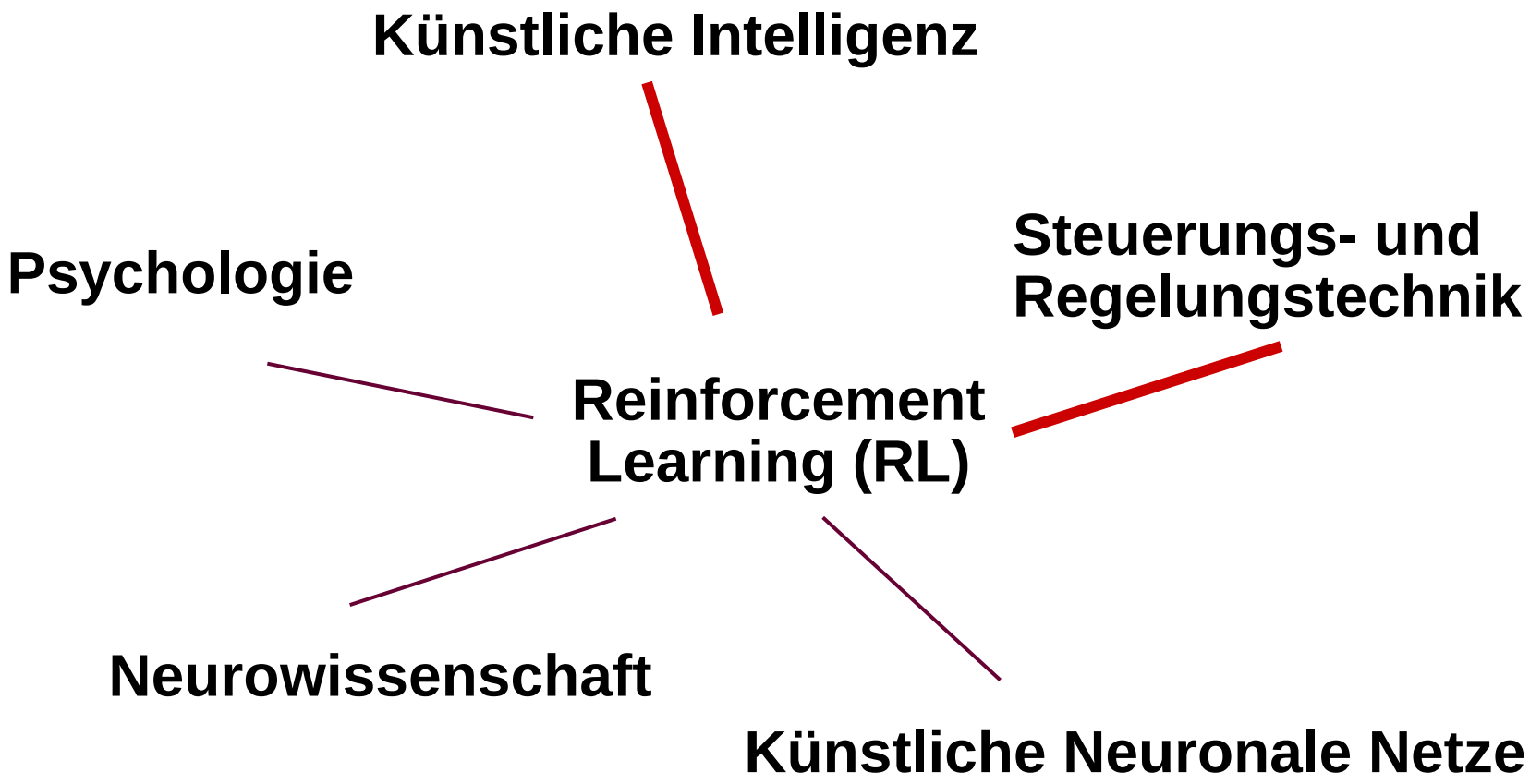
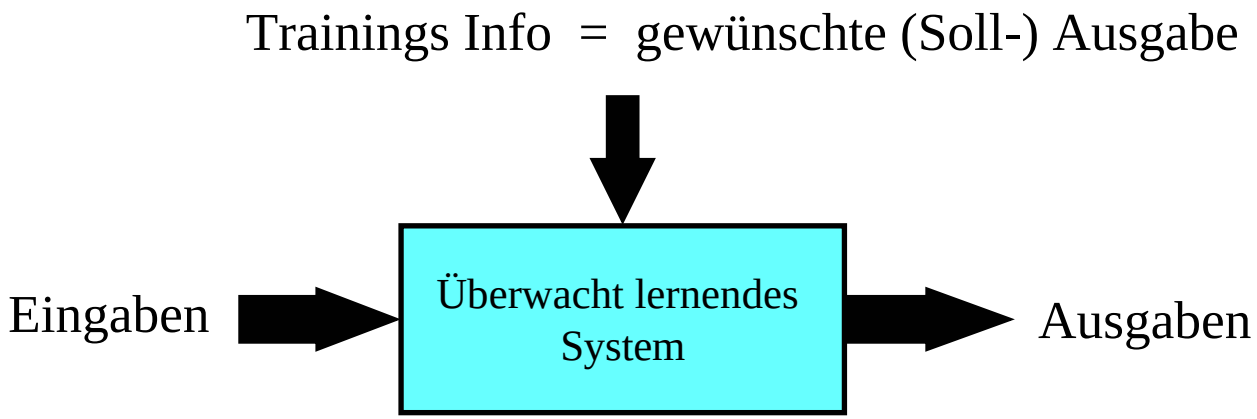
Einführung


Künstliche Intelligenz
Psychologie
Steuerungs- und Regelungstechnik
Reinforcement Learning (RL)
Neurowissenschaft
Künstliche Neuronale Netze

Was ist Reinforcement Learning?

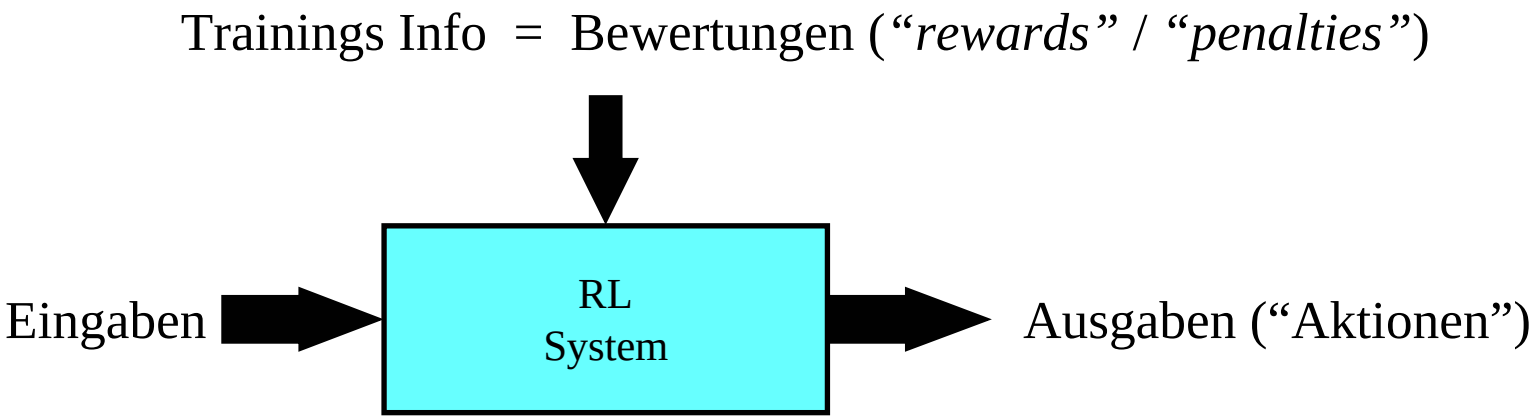
- Lernen aus Interaktion
- Ziel-orientiertes Lernen
- Lernen durch, von, und während der Interaktion mit einer externen Umgebung
- Lernen “was zu tun ist” — wie man Situationen auf Aktionen abbildet — um ein numerisches *Reward*-Signal zu maximieren

Überwachtes Lernen



Fehler = (Soll-Ausgabe – Systemausgabe)

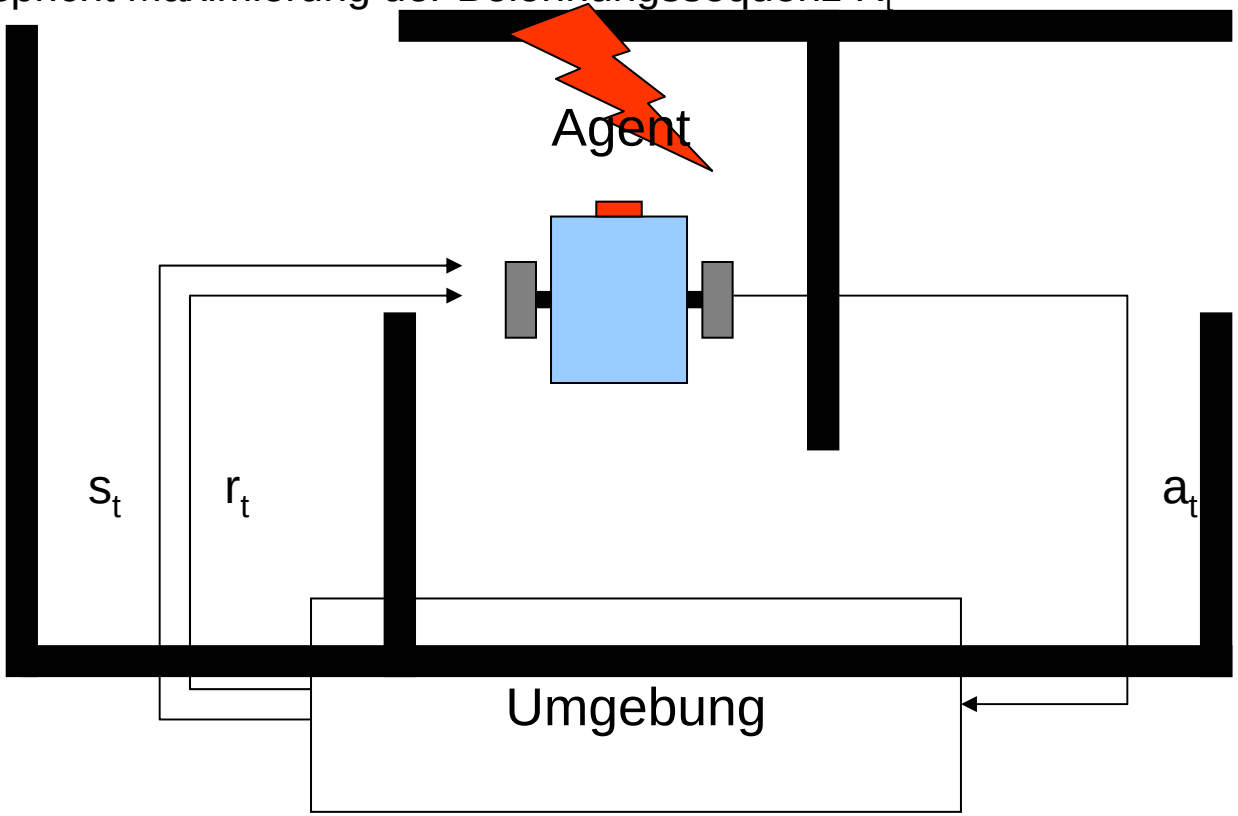
Reinforcement Learning



Ziel: erreiche soviel *Reward* wie möglich

Reinforcement Learning

- Ziel: Möglichst „erfolgreich“ in der Umgebung agieren
- Entspricht Maximierung der Belohnungssequenz R_t

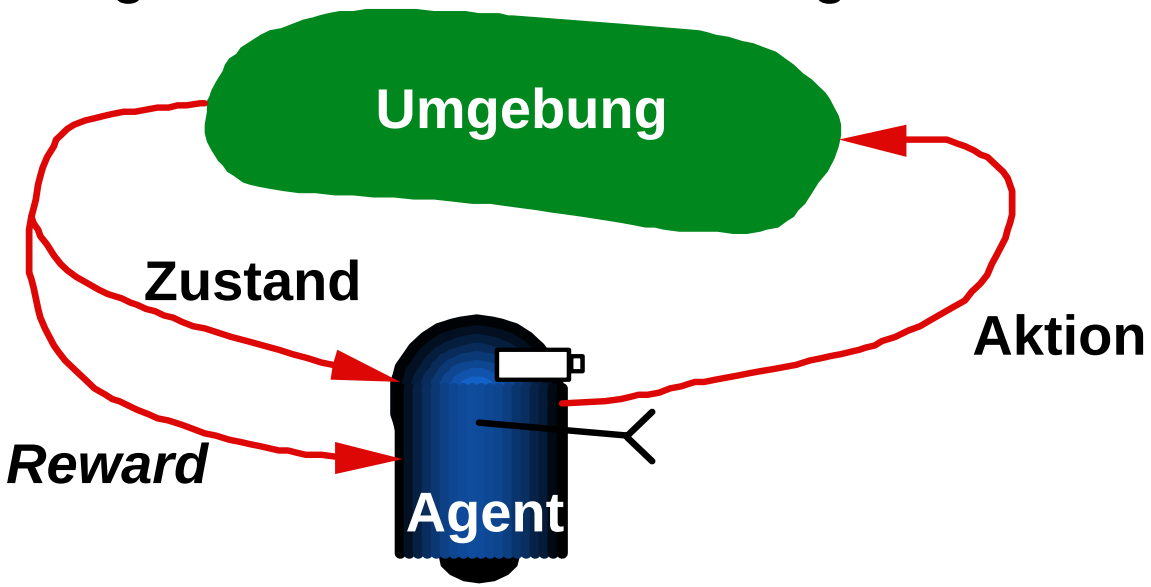


Key Features von RL

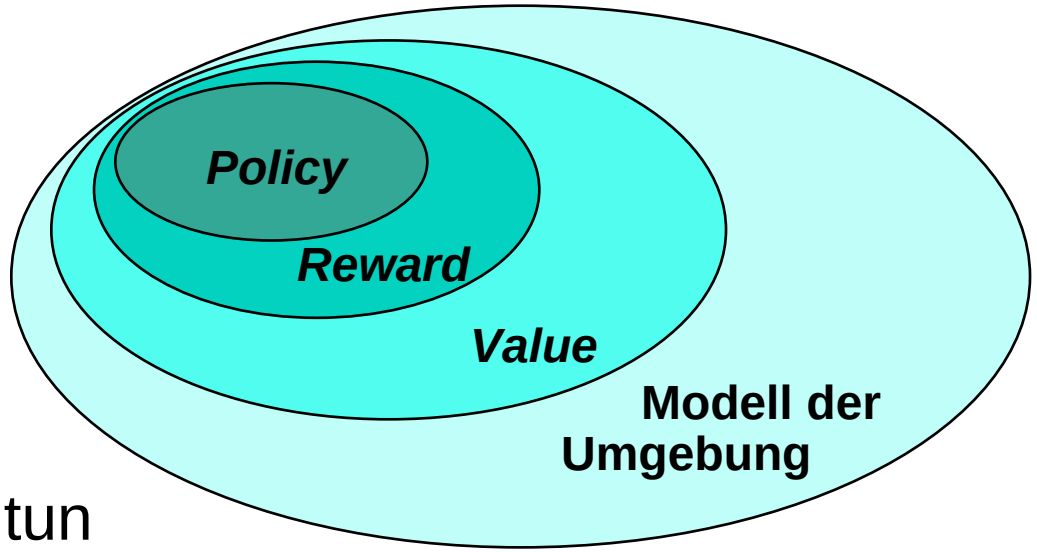
- Lerner bekommt nicht gesagt welche Aktionen zu wählen sind
- Trial-and-Error Suche
- Möglichkeit eines verspäteten ("delayed") *Reward*
  - Aufgeben von kurzfristigem Ertrag um höheren langfristigen Ertrag zu erhalten
- Das Dilemma "***exploration***" vs. "***exploitation***"
- Betrachte das komplette Problem eines ziel-orientierten Agenten in Interaktion mit einer unsicheren Umgebung

Der vollständige Agent

- Zeitlich situiert
- Beständiges Lernen und Planen
- Beeinflusst die Umgebung
- Umgebung ist stochastisch und ungewiss

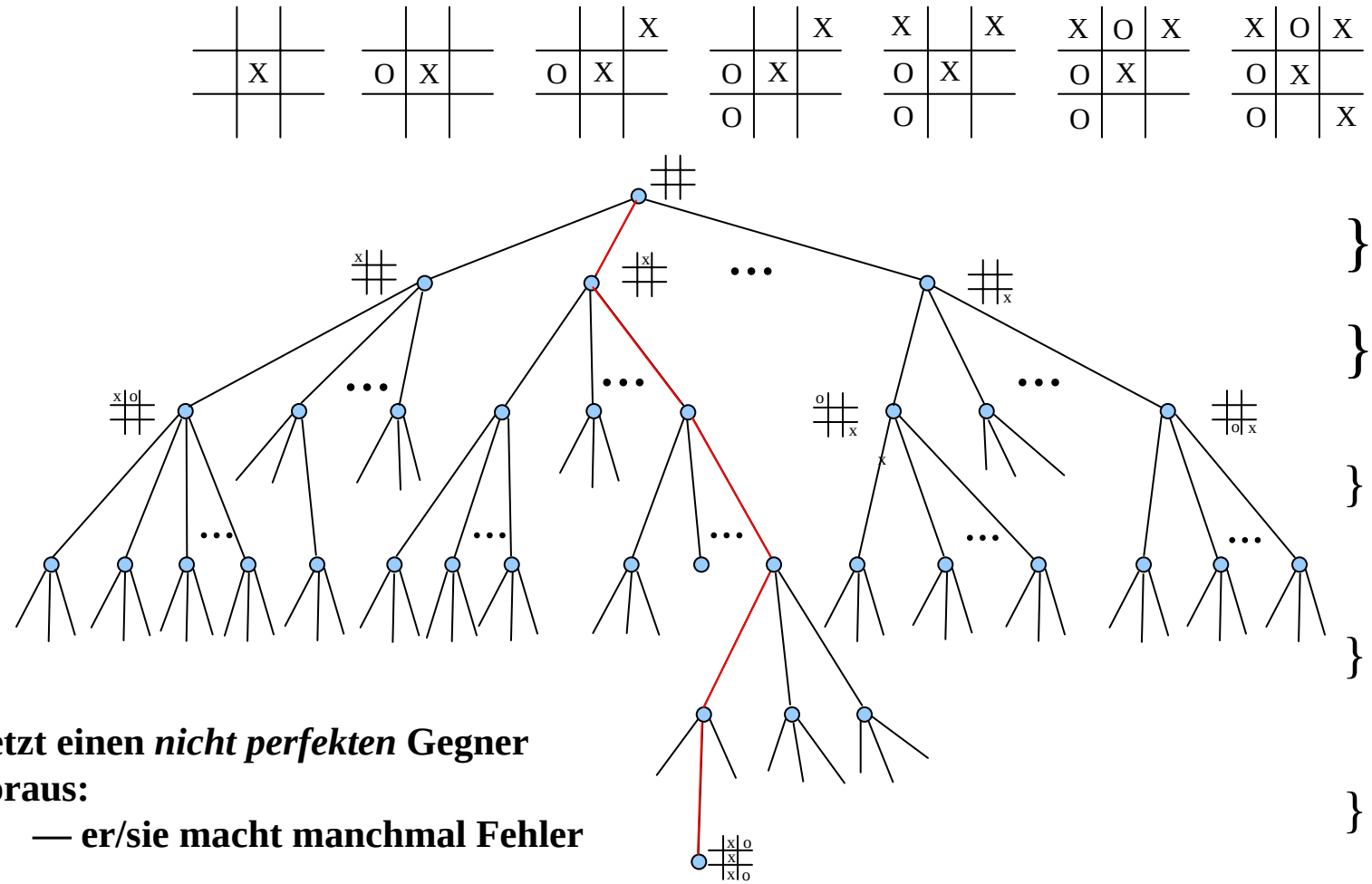


Elemente des RL



- ***Policy***: was ist zu tun
- ***Reward***: was ist gut
- ***Value***: was ist gut, da es Reward vorhersagt
- **Modell**: was folgt auf was

Ein erweitertes Beispiel: Tic-Tac-Toe



Setzt einen *nicht perfekten* Gegner voraus:

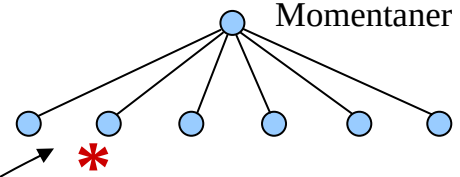
— er/sie macht manchmal Fehler

Ein RL Ansatz für Tic-Tac-Toe

1. Erstelle eine Tabelle mit einem Eintrag pro Zustand:

Zustand	$V(s)$ – geschätzte Wahrscheinlichkeit für den Gewinn	
	.5	
	.5	
⋮	⋮	
	1	gewonnen
⋮	⋮	
	0	verloren
⋮	⋮	
	0	unentschieden

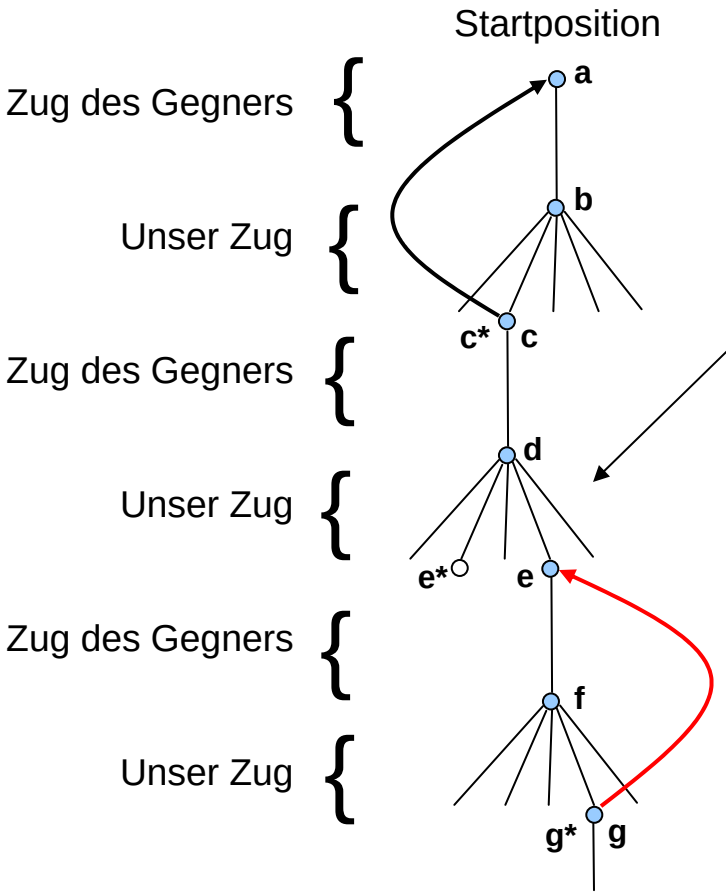
2. Jetzt spiele viele Spiele.
   Um einen Zug zu wählen,
   schaue einen Schritt nach vorne:



Nehme den nächsten Zustand mit der höchsten geschätzten Gewinnwahrscheinlichkeit — das höchste $V(s)$; ein ***greedy*** Zug.

Aber in 10% aller Fälle wähle einen zufälligen Zug; ein ***explorierender*** Zug.

RL-Lernregel für Tic-Tac-Toe



Explorierender Zug

s – Zustand vor dem *greedy* Zug

s' – Zustand nach dem *greedy* Zug

Wir inkrementieren jedes $V(s)$ zu $V(s')$ – ein ***„backup“***

$$V(s) \leftarrow V(s) + \alpha \left[V(s') - V(s) \right]$$

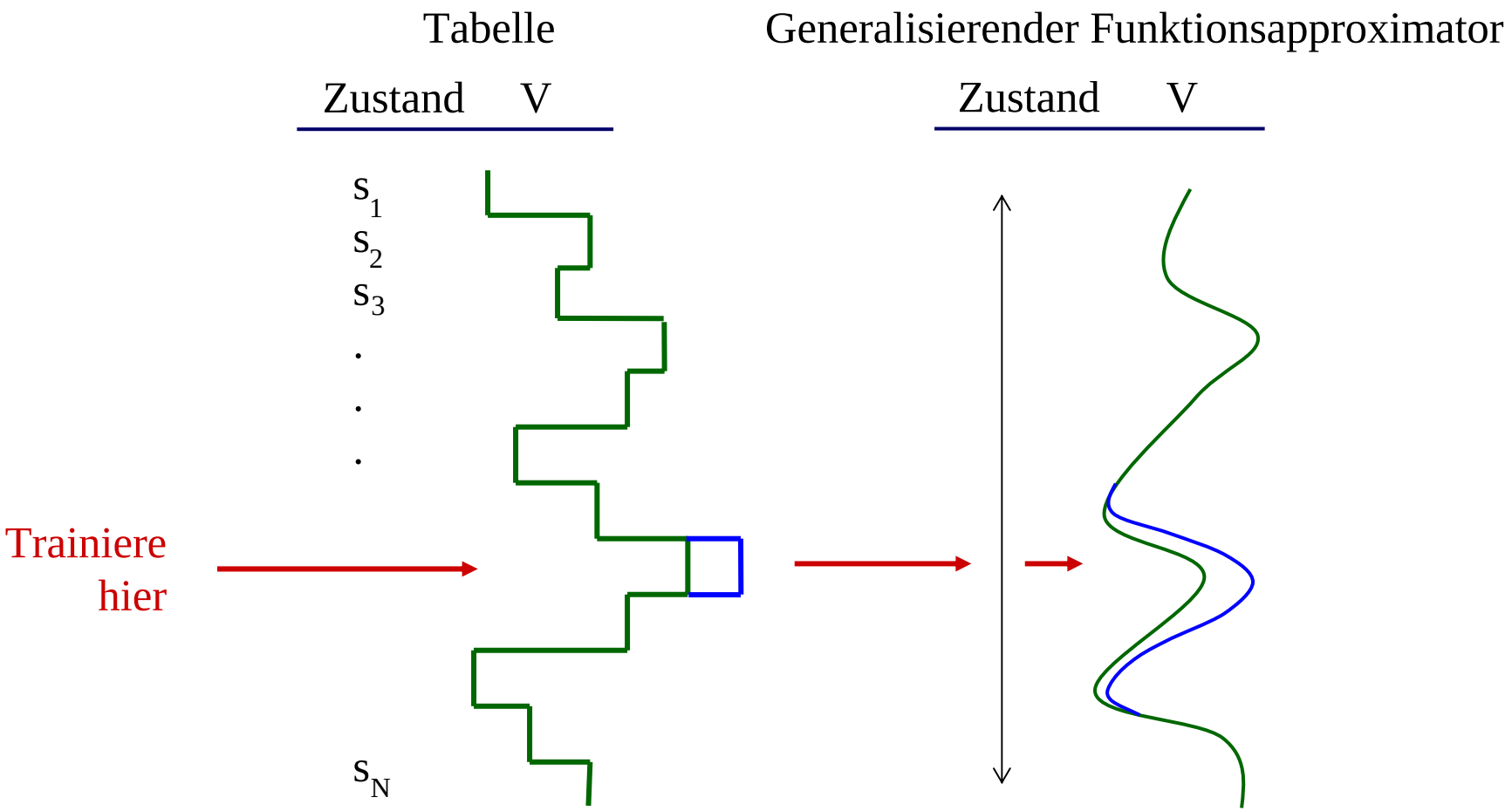
kleiner positiver Wert, z.B. $\alpha = 0.1$

der **„Schrittweitenparameter“**

Verbesserung des T.T.T Spielers

- Beachten von Symmetrien
  - Darstellung/Generalisierung
  - Wie kann dies fehlschlagen?
- Braucht man “Zufallszüge”? Warum?
  - Braucht man immer die 10%?
- Kann man von “Zufallszügen” lernen?
- Kann man *offline* lernen?
  - Vor-Lernen durch Spielen gegen sich selbst?
  - Verwendung von gelernten Modellen des Gegners?
- . . .

z.B. Generalisierung



Warum ist Tic-Tac-Toe einfach?

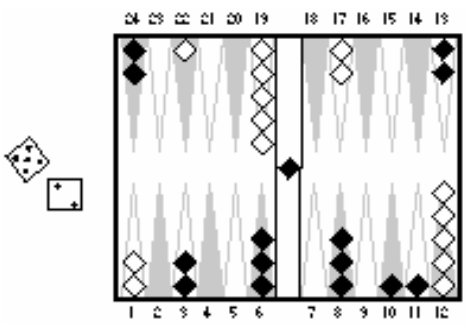
- Endliche, kleine Anzahl an Zuständen
- Es ist immer möglich einen Schritt nach vorne zu gucken (*one-step look ahead*)
- Zustände komplett wahrnehmbar
- . . .

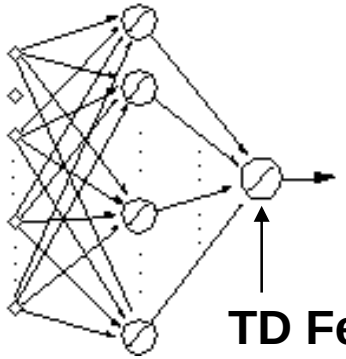
Einige namhafte RL Anwendungen

- **TD-Gammon**: Tesauro
  - weltbestes Backgammon Programm
- **Aufzugssteuerung**: Crites & Barto
  - High Performance “down-peak” Aufzugscontroller
- **Lagerverwaltung**: Van Roy, Bertsekas, Lee & Tsitsiklis
  - 10–15% Verbesserung gegenüber standard Industriemethoden
- **Dynamische Kanalzuordnung**: Singh & Bertsekas, Nie & Haykin
  - High Performance Zuordnung von Funkkanälen zu Mobiltelefonaten

TD-Gammon

Tesauro, 1992–1995





Aktionsauswahl durch 2–3 Lagensuche

Starte mit zufälligem Netzwerk

Spiele sehr viele Spiele gegen dich selbst

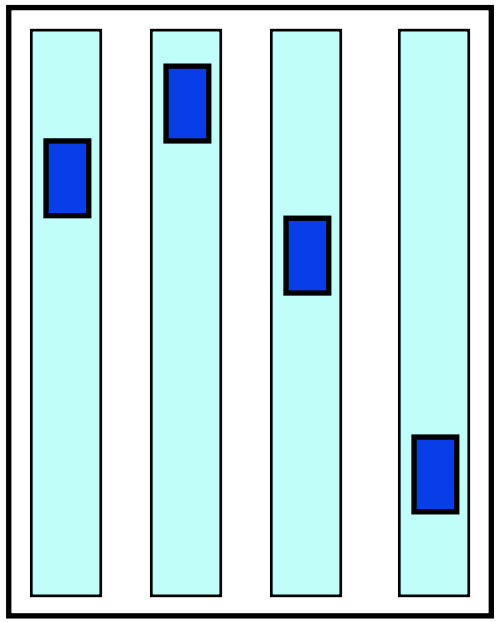
Lerne eine Wertefunktion anhand dieser simulierten Erfahrung

Dies produziert wohl den besten Spieler der Welt

Aufzugseinteilung

Crites and Barto, 1996

10 Stockwerke, 4 Kabinen



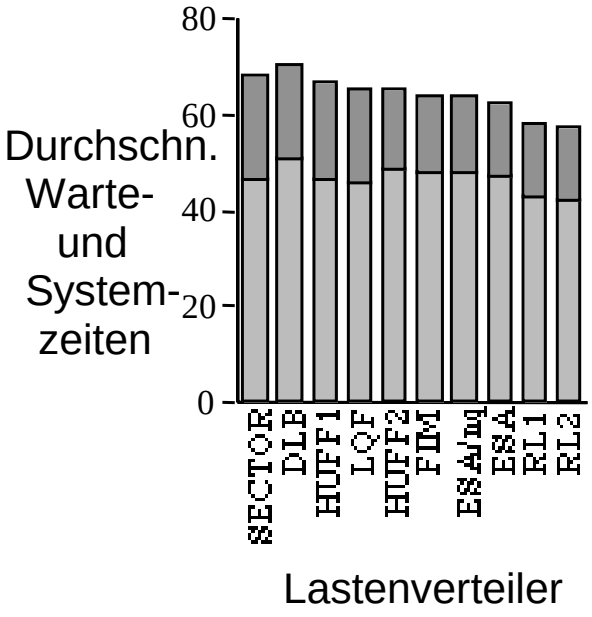
Zustände: Knopfzustände; Positionen, Richtungen, und Bewegungszustände der Kabinen; Personen in Kabinen & in Etagen

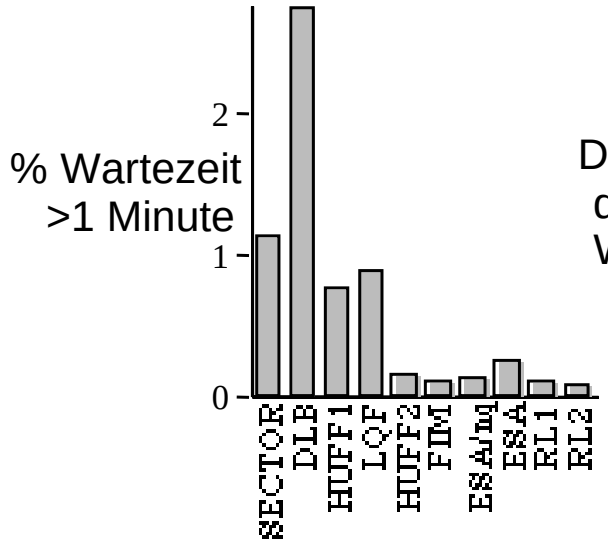
Aktionen: halte an X, oder fahre nach Y, nächste Etage

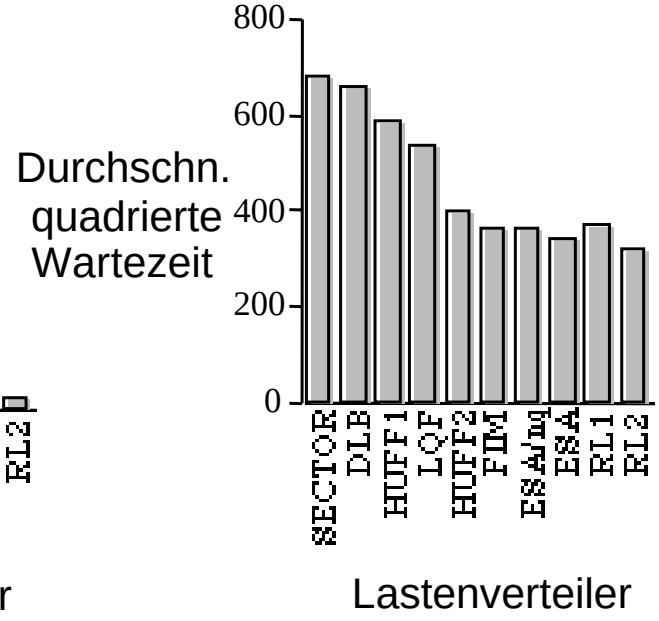
Rewards: geschätzt, –1 pro Zeitschritt für jede wartende Person

Vorsichtige Schätzung: ca. 10^{22} Zustände

Performance Vergleich







RL Geschichte

Trial-and-Error learning	Temporal-difference learning	Optimal control, value functions
Thorndike (Ψ) 1911	Secondary reinforcement (Ψ)	Hamilton (Physics) 1800s
		Shannon
Minsky	Samuel	Bellman/Howard (OR)
	Holland	
Klopf		
	Witten	Werbos
Barto et al.	Sutton	
		Watkins

MENACE (Michie 1961)

"Matchbox Educable Noughts and Crosses Engine"