

**PROSEMINAR ROBOTIK**

**SEMINAR REINFORCEMENT LEARNING**

**OBERSEMINAR ADAPTIVE  
ROBOTERSTEUERUNG**

Organisation, Überblick, Themen

# Überblick heutige Veranstaltung

- Organisatorisches
- Einführung in Reinforcement Learning
- Vorstellung der Themen

# Organisation

- „ Robotik “
  - Proseminar mit 2 SWS (3LP)
- „ Reinforcement Learning“
  - Seminar mit 2 SWS (3 LP)
- „ Adaptive Robotersteuerung“
  - Oberseminar mit 2 SWS (3 LP)
- Ansprechpartner:
  - Uwe Dick, Raum 03.04.0.19, [uwedick@cs.uni-potsdam.de](mailto:uwedick@cs.uni-potsdam.de)
  - Prof. Tobias Scheffer, Raum 03.04.0.17, [scheffer@cs.uni-potsdam.de](mailto:scheffer@cs.uni-potsdam.de)
- Webseite:
  - <http://www.cs.uni-potsdam.de/ml/teaching/ws13/reinforcement.html>

# Ablauf des Seminars

- Heute: Vorstellung der Themen
- Themenwahl per Email an mich
- Nächste Woche: Video-Vorlesung zu Methodik für Ausarbeitung und Vortrag
- Jedem Thema wird ein Betreuer zugewiesen, mit dem der Teilnehmer Termine zur Besprechung vereinbaren kann
- Schriftliche Ausarbeitung und Seminarvortrag (20 min)
- Die Vorträge der Teilnehmer in Blöcken später im Semester (6., 13. und 20. 01. 14, bei vielen Teilnehmern zusätzlicher Termin).

# Literatur / Vorlesung

- Es wird empfohlen, als einführende Lektüre *Richard Sutton, Andrew Barto: Reinforcement Learning* zu lesen. (Findet sich im Web)
- Desweiteren seien die Vorlesungen RL1 und RL2 im Rahmen der Veranstaltung *Machine Learning 2* im nächsten Semester empfohlen.

# Reinforcement Learning

- = Lernen von sequenziellen Entscheidungen. Die Güte einer Entscheidung wird durch die Güte der Entscheidungssequenz bestimmt.
  - ▣ Temporal Credit Assignment Problem.
  - ▣ Modell für Auswirkungen einer Entscheidung oft nicht bekannt.
  
- $\neq$  überwachtes Lernen: Lernen einer Entscheidungsfunktion aus Beispielen der richtigen (Einzel-)Entscheidung.

# Reinforcement Learning

## □ Beispiele

- Schach / Go: Welcher Zug hat welchen Einfluss auf das Spielergebnis?
- Robofußball: Eine positive Bewertung bekommt der Roboter für ein geschossenes Tor. Aber welche Bewegungen haben das ermöglicht?
- Helikopterflug: Welche Bewegungen müssen ausgeführt werden, um bei unvorhersehbaren äußeren Bedingungen nicht abzustürzen?

# Was ist Reinforcement Learning?

- RL-Methoden sind „Sampling based methods to solve optimal control problems “ (Richard Sutton)
- Suche nach einer optimalen Policy: Funktion von Zustand zu Aktion.
- Optimalität des Lernens: Policy mit höchstem erwarteten Reward.
- Aber auch: schnelles Lernen ohne zuviele Fehler zu machen.



# Markov Decision Processes

- Markov-Entscheidungsprozess  $(S, A, R, P)$
- $S$  : endliche Zustandsmenge
- $A$  : endliche Aktionsmenge
- $P$  : Übergangswahrscheinlichkeiten

$$P(s'|s, a) \quad s, s' \in S, a \in A$$

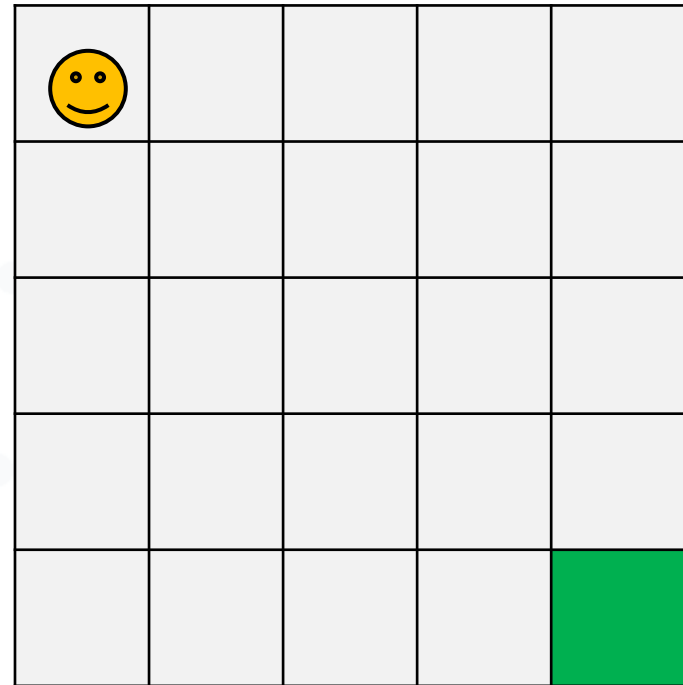
- $R$  : Erwarteter Reward. Beschreibt den sofort erzielten Gewinn.

$$R : (S \times A) \rightarrow \mathbb{R}$$

- Discount factor  $0 \leq \gamma < 1$  .

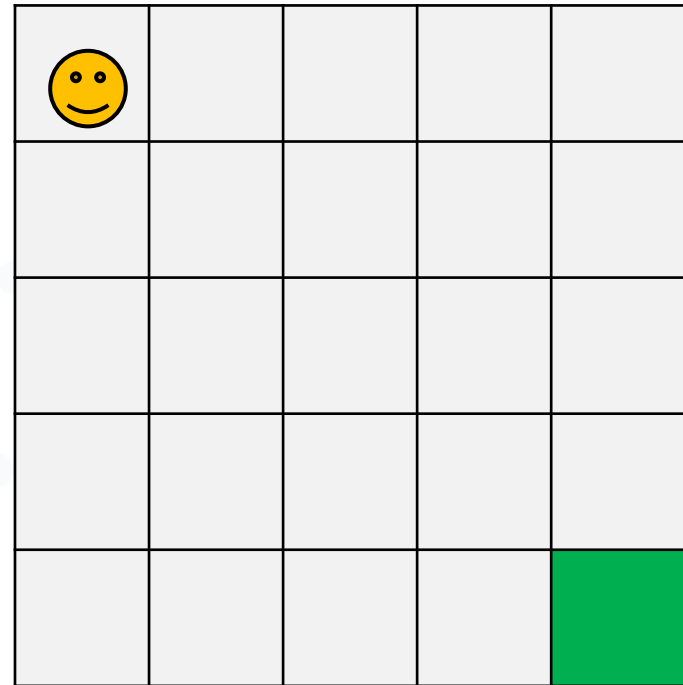
# Beispiel: Gridworld

- Zustandsraum  $\mathcal{S}$ 
  - Startzustand  $s_s \in \mathcal{S}$
  - Zielzustand  $s_z \in \mathcal{S}$



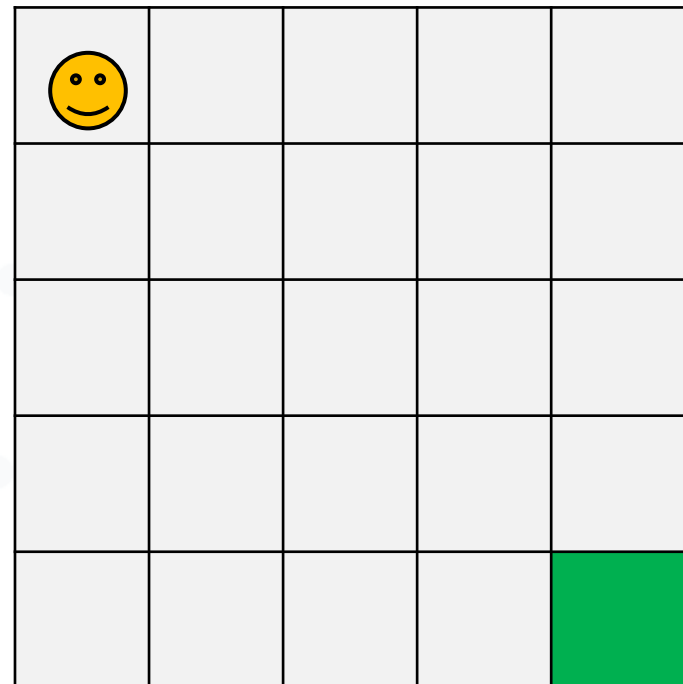
# Beispiel: Gridworld

- Zustandsraum  $S$
- Aktionsmenge  $A$ 
  - $A = (\text{links, rechts, oben, unten})$



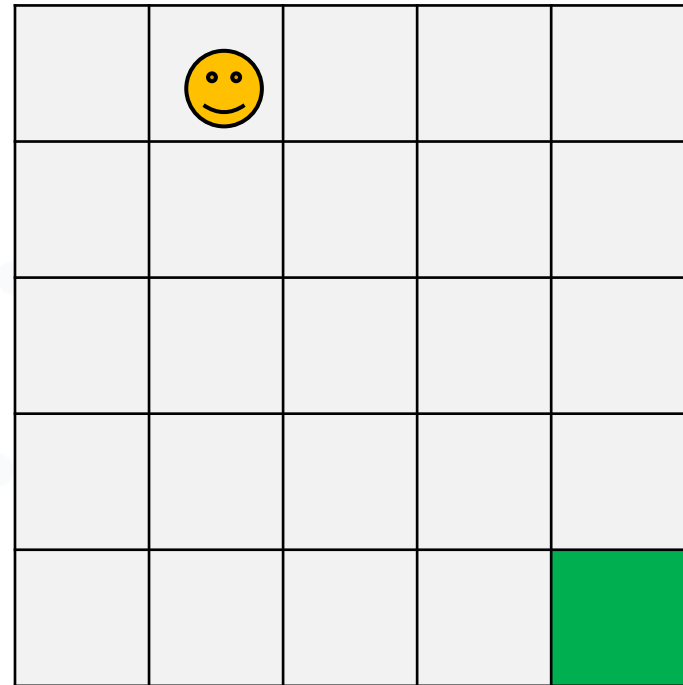
# Beispiel: Gridworld

- Zustandsraum  $S$
- Aktionsmenge  $A$
- Übergangswahrscheinlichkeit  $P$ 
  - $P((1,2) | (1,1), \text{rechts}) = 1$



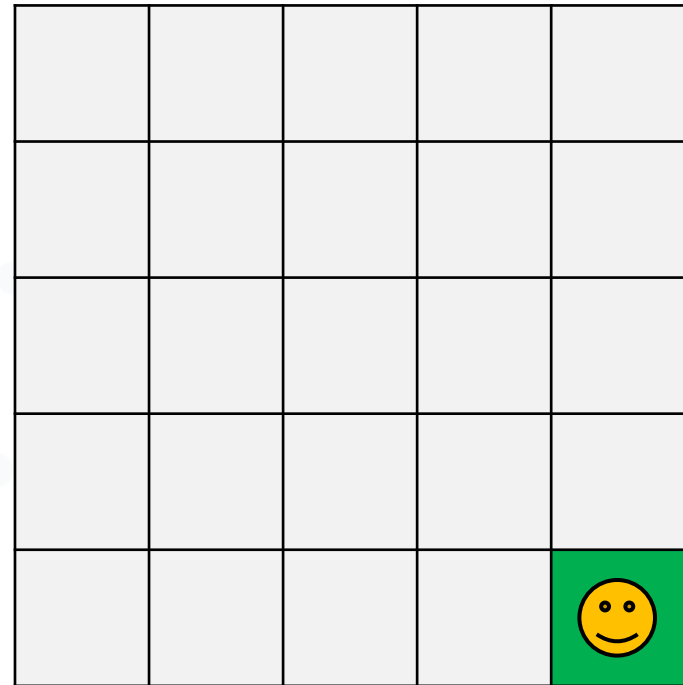
# Beispiel: Gridworld

- Zustandsraum  $S$
- Aktionsmenge  $A$
- Übergangswahrscheinlichkeit  $P$
- Erwarteter Reward  $R$ 
  - $R((1,1),\text{rechts}) = 0$



# Beispiel: Gridworld

- Zustandsraum  $S$
- Aktionsmenge  $A$
- Übergangswahrscheinlichkeit  $P$
- Erwarteter Reward  $R$ 
  - $R((4,5),\text{unten}) = 1$



# Markov Decision Processes

- Markov-Entscheidungsprozess  $(S, A, R, P)$
- $S$  : endliche Zustandsmenge
- $A$  : endliche Aktionsmenge
- $P$  : Übergangswahrscheinlichkeiten

$$P(s'|s, a) \quad s, s' \in S, a \in A$$

- $R$  : Erwarteter Reward. Beschreibt den sofort erzielten Gewinn.

$$R : (S \times A) \rightarrow \mathbb{R}$$

- Discount factor  $0 \leq \gamma < 1$  .

# MDP

- Eine deterministische stationäre Policy bildet Zustände auf Aktionen ab.  $\pi : S \rightarrow A$

- Stochastische Policy: Funktion von Zuständen auf eine Verteilung von Aktionen.

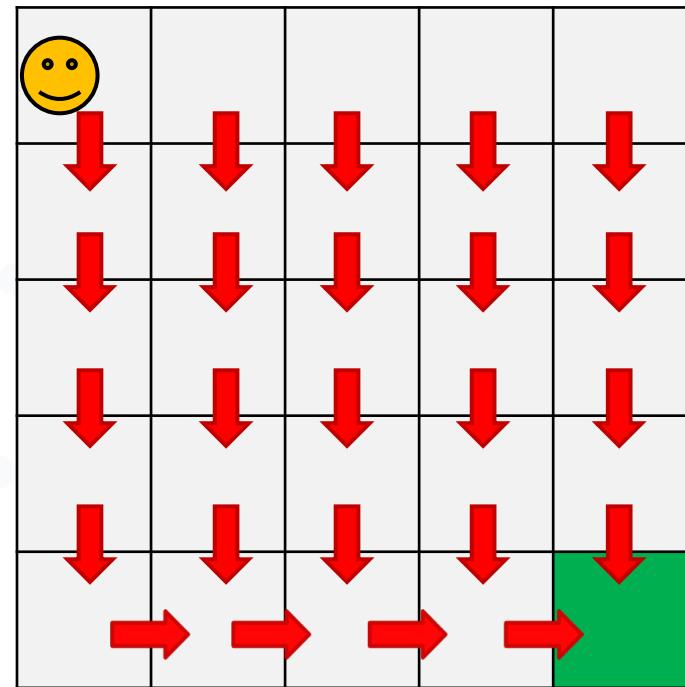
- Ziel: Finde Policy  $\pi$  die den erwarteten kumulativen (discounted) Gewinn maximieren.

$$E_{\pi, P} \left[ \sum_{t=0}^{\infty} \gamma^t R(s_t, \pi(s_t)) \right]$$



# Beispiel: Gridworld

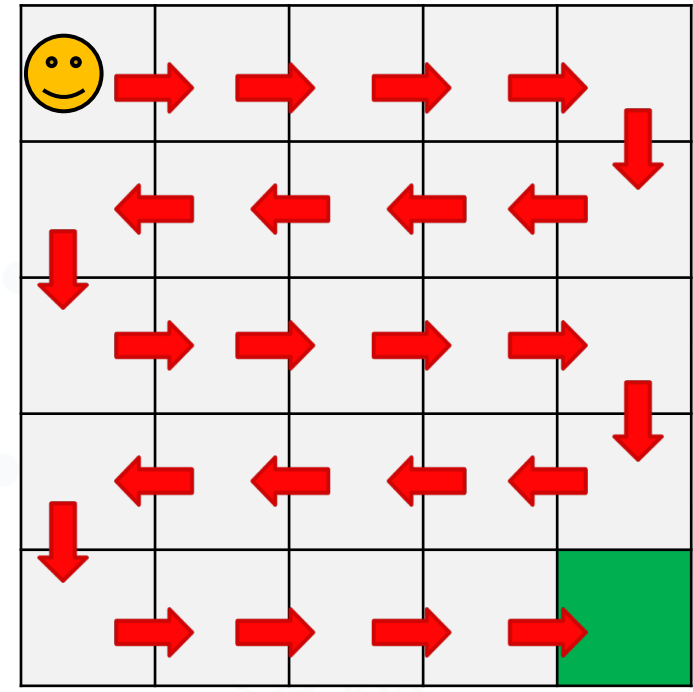
- Zustandsraum  $S$
- Aktionsmenge  $A$
- Übergangswahrscheinlichkeit  $P$
- Erwarteter Reward  $R$
- Discountfaktor  $\gamma = 0,9$
- Policy  $\pi$ 
  - Gute Policy  $\pi_1 \rightarrow$
- Erwarteter discounted Reward



$$E_{\pi, P} \left[ \sum_{t=0}^{\infty} \gamma^t R(s_t, \pi(s_t)) \mid s_0 = s_s \right] = 0,9^7$$

# Beispiel: Gridworld

- Zustandsraum  $S$
- Aktionsmenge  $A$
- Übergangswahrscheinlichkeit  $P$
- Erwarteter Reward  $R$
- Discountfaktor  $\gamma = 0,9$
- Policy  $\pi$ 
  - ▣ Schlechte Policy  $\pi_2$  →
- Erwarteter diskounted Reward



$$E_{\pi,P} \left[ \sum_{t=0}^{\infty} \gamma^t R(s_t, \pi(s_t)) \mid s_0 = s_s \right] = 0,9^{23}$$

# Value Functions – Bewertungsfunktionen

- Value function  $V^\pi(s)$  für einen Zustand  $s$  und Policy  $\pi$  beschreibt den erwarteten kumulativen Gewinn der von diesem Zustand aus erreicht wird.


$$V^\pi(s_t) = E_{\pi, P} \left[ \sum_{k=0}^{\infty} \gamma^k R(s_{t+k}, \pi(s_{t+k})) \right]$$

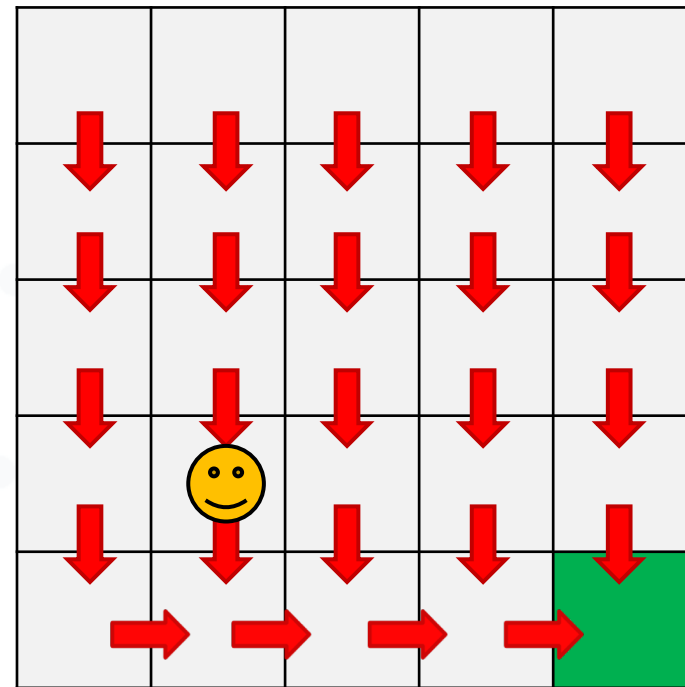
- Es existiert immer eine optimale deterministische stationäre Policy  $\pi^*$ , die die Value Function maximiert.

$$V^*(s) = \max_{\pi} V^\pi(s)$$

$$V^{\pi^*}(s) = V^*(s)$$

# Beispiel: Gridworld

- Zustandsraum  $S$
- Aktionsmenge  $A$
- Übergangswahrscheinlichkeit  $P$
- Erwarteter Reward  $R$
- Discountfaktor  $\gamma = 0,9$
- Policy  $\pi$ 
  - Gute Policy  $\pi_1$  
- Value Function  $V$



$$V^{\pi_1}(s_t) = E_{\pi, P} \left[ \sum_{k=0}^{\infty} \gamma^k R(s_{t+k}, \pi(s_{t+k})) \right] = 0,9^3$$

# Modellwissen

- Es ergeben sich unterschiedliche Problemstellungen je nach Wissen über den MDP.
- MDP vollständig bekannt.  
→ Planen.
- MDP nicht oder teilweise bekannt. Es kann Erfahrung gesammelt werden durch Interaktion mit der Umgebung.  
→ Reinforcement Learning.

# Arten von Reinforcement Learning

- Reinforcement Learning-Methoden können eingeteilt werden bezüglich der Verwendung der Interaktionsbeispiele.
  
- Indirekte Methoden:
  - Model learning
  
- Direkte Methoden:
  - Direct Policy Search
  - Value function estimation

# Themenübersicht Proseminar

- DARPA Grand Challenge
- Autonome Autos (Mercedes / Google)
- RoboCup
- BigDog / LittleDog
- Helicopter
- Quadrocopter
- STAIR
- Rescue Robot
- Eigenes Thema?

# Robotik: DARPA Grand Challenge

- Wettbewerb für unbemannte Landfahrzeuge.
  - 2005: Autonome Fahrt über Wüstenstraßen
    - Gewinner: Stanley, entwickelt in Stanford unter Sebastian Thrun (Jetzt Google, z.B. Google driverless car)
  - 2007: Urban Challenge. Verlassene Kaserne, also stadtartiges Gelände
    - Gewinner: Tartan Racing von der CMU.





# Robotik: Autonome Autos (1-2 Themen)

## Mercedes-Benz und Google

- 2 Projekte für selbststeuernde Autos:
  - Google driverless car
  - Mercedes-Benz



# Robotik: RoboCup (1-2 Themen)

- Mehrere Ligen und (Welt-)Meisterschaften in unterschiedlichen Roboterklassen, z.B. Humanoide in unterschiedlichen Größen, Robo-Hunde
- Ausarbeitungen und Vorträge können auch gerne näher auf einen bestimmten technischen Aspekt eingehen, z.B. das Lernen eines guten Schusses oder der Positionsbestimmung.



# Robotik: BigDog / LittleDog

- Roboter, der von der Stanford-Universität und Boston Dynamics entwickelt wurde.
- Vierbeiner, die bereits anspruchsvolle Aufgaben erledigen können, z.B. Laufen über schwieriges Terrain.



# Robotik: Autonomous Helicopter

- An der Stanford-Universität entwickelter selbständig fliegender Modellhubschrauber, der schwierigste Manöver mindestens so gut bewerkstelligen kann wie menschliche Weltmeister.



# Robotik: Quadrocopter

- Autonome Quadrocopter, die z.T. auch auf Einfachheit der Bauteile wert legen, wie z.B. das Verwenden von Handies als Kameraelemente.
- Uni Tübingen.





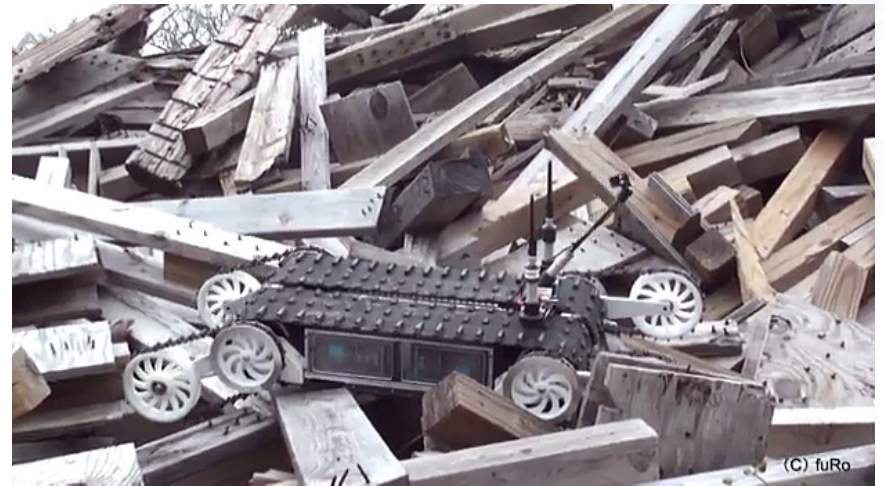
# Robotik: STAIR (1-2 Themen)

- Stanford Artificial Intelligence Robot
- 2 mögliche vertiefende Themen:
  - Greifen von Gegenständen
  - Sprachkommunikation



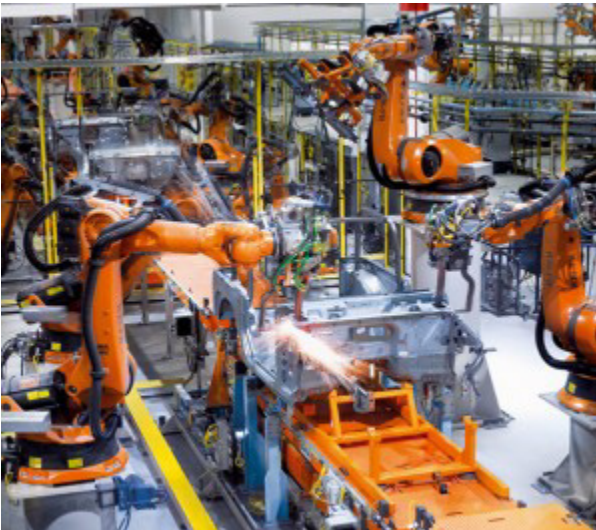
# Robotik: Rescue Robot

- Roboter, die zur Rettung von Menschen eingesetzt werden, aber auch zum Begehen ansonsten unzugänglicher Orte.
  - Z.B. Roboter Quince in Fukushima



# Robotik: Eigenes Thema?

- Es gibt natürlich noch sehr viele andere mögliche Seminararbeiten zum Thema Robotik. Wenn Sie eine gute Idee haben, können Sie uns gerne darauf ansprechen!





# Themenübersicht Proseminar

- DARPA Grand Challenge
- Autonome Autos (Mercedes / Google)
- RoboCup
- BigDog / LittleDog
- Helicopter
- Quadrocopter
- STAIR
- Rescue Robot
- Eigenes Thema?

# Themenüberblick Seminar und Oberseminar

- Hierarchisches Reinforcement Learning
- Planen in großen Zustandsräumen
- RL in Spielen
  - Go
  - Backgammon
- Modellbasiertes RL
- Roboterfußball
- Apprenticeship Learning:
  - Autonomer Helikopterflug
  - Autonomer vierfüßiger Roboter
- Actor-Critic: Traffic Routing



# Hierarchisches RL

- Dekomposition von Policy und Value Function in Hierarchische Subtasks.
- Verbesserte Exploration durch größere Schritte auf höherer Abstraktionsebene
- Es müssen weniger Parameter gelernt werden
- Besserer Transfer des Gelernten auf neue Probleme, da gelernte Subtasks wiederverwendet werden können.
- *T.G. Dietterich: The MAXQ Method for Hierarchical Reinforcement Learning.*

# Planen in großen Zustandsräumen

- Angenommen ein Model ist bekannt, aber der Zustandsraum ist sehr groß -> Dynamische Programmierung nicht mehr realisierbar.
- Monte-Carlo-Sampling Methoden besser geeignet.
- Baum-basierte Methoden
- Deutliche Verbesserung der Performanz durch State-Of-The-Art Methode UCT
  
- *Kearns, Mansour, Ng: A Sparse Sampling Algorithm for Near-Optimal Planning in Large Markov Decision Processes.*
- *Kocsis, Szepesvári: Bandit based Monte-Carlo Planning.*

# RL in Spielen 1: Go

- Go ist ein Brettspiel, in dem die Spielstärke menschlicher Großmeister noch immer weit über der der besten Computerprogramme liegt.
- Neue Methoden des RL haben jedoch kürzlich zu einer deutlichen Leistungssteigerung von Programmen geführt.
- Upper Confidence Bound for Trees (UCT).
  
- *Sylvain Gelly: Ph.D. Thesis, Kapitel 4*
- *Kocsis, Szepesvári: Bandit based Monte-Carlo Planning.*
- *Gelly, Silver: Combining Online and Offline Knowledge in UCT.*

# RL in Spielen 2: Backgammon

- Eine der großen Erfolgsgeschichten von RL.
- Bereits 1996 wurde TD-Gammon entwickelt, ein Programm, das erstaunlich gute Ergebnisse lieferte.
- Eine weitere Möglichkeit ist das Lernen mit UCT.
  
- *G. Tesauro: Temporal Difference Learning and TD-Gammon.*
- *Lishout, Chaslot, Uiterwijk: Monte-Carlo Tree Search in Backgammon.*
- *Kocsis, Szepesvári: Bandit based Monte-Carlo Planning.*

# Modellbasiertes RL: Anwendung für Produktauslieferung

- Anwendung impliziert sehr große Zustands- und Aktionsmengen.
- Beispiel eines modellbasierten RL-Algorithmus.
- Zeigt exemplarisch, welche Probleme in RL-Anwendungen auftreten können und wie sie gelöst werden können.
  
- *Proper, Tadepalli: scaling Model-Based Average-Reward Reinforcement Learning for Product Delivery.*
- *Sutton, Barto: Reinforcement Learning, Kap. 8*

# Anwendung: Roboterfußball

- Roboterfußball eine bevorzugte Test- und Entwicklungsumgebung zur Erforschung autonomer Robotersteuerung.
- Ausschnitt aus einer Vielzahl von behandelten Problemen und Lösungen durch RL.
- *Kohl, Stone: Machine Learning for Fast Quadrupedal Locomotion.*
- *Hester, Quinlan, Stone: Generalized Model Learning for Reinforcement Learning on a Humanoid Robot*
- *Hausknecht, Stone: Learning Powerful Kicks on the Aibo ERS-7: The Quest for a Striker.*



# Apprenticeship Learning: Helikopter

- Apprenticeship Learning: Lernen von Beispieltrajektorien, also Beispielen einer optimalen Steuerung durch einen Experten.
  - Inverses RL: Reward-Funktion ist unbekannt. Lerne aus Beispielen, welcher Reward am wahrscheinlichsten optimiert werden sollte. Lerne daraufhin die Policy.
  - Anwendung: Autonomer Helikopterflug.
- 
- Coates, Abbeel,Ng: *Learning for Control From Multiple Demonstrations.*
  - Coates, Abbeel,Ng: *Apprenticeship Learning for Helicopter Control.*
  - Abbeel,Ng: *Apprenticeship Learning via Inverse Reinforcement Learning.*

# Apprenticeship Learning: LittleDog

- Apprenticeship Learning.
- Anwendung: Lernen von autonomer Steuerung eines vierfüßigen Roboters.
  
- *Kolter, Abbeel, Ng: Hierarchical Apprenticeship Learning, with Application to Quadrupes Locomotion.*
- *Abbeel, Ng: Apprenticeship Learning via Inverse Reinforcement Learning.*

# Actor-Critic Learning—Traffic Routing

- Policy Gradient Learning, im Speziellen Natural Actor-Critic (NAC) Methoden, lernen die Policy direkt, indem ein Gradient der erwarteten Rewards bzgl. der Parameter der Policy geschätzt wird.
- Anwendung auf optimale Verkehrsführung in Sydney.

# Themenüberblick Seminar und Oberseminar

- Hierarchisches Reinforcement Learning
- Planen in großen Zustandsräumen
- RL in Spielen
  - Go
  - Backgammon
- Modellbasiertes RL
- Roboterfußball
- Apprenticeship Learning:
  - Autonomer Helikopterflug
  - Autonomer vierfüßiger Roboter
- Actor-Critic: Traffic Routing



# Themenwahl: per Mail

- Mail an mich mit 3 Themenwünschen, welches Seminar, Mat. Nr.
  - [uwedick@cs.uni-potsdam.de](mailto:uwedick@cs.uni-potsdam.de)



# Termine

- Vorlesungsvideo zur Methodik: nächsten Montag
- Deadline erste Version der Ausarbeitung: 18.11.
- Deadline Endversion Ausarbeitung, erste Version der Folien: 2.12.
- Seminarvorträge: 6., 13. und 20. 01. 14
- Abgabe der Vortragsfolien: für den ersten Termin 16.12., ansonsten 2 Wochen vorher.