Agenten und Robotfußball

Selbstorganisation und Lernen

Maschinelles Lernen im Robotfußball

Dieter Kerkfeld

Agenda

Einleitung

Simulationsliga ermöglicht Forschung an intelligenten Software-Systemen, den so genannten intelligenten Agenten

Intelligente Agenten in der Simulationsliga:

- Leiten ihr Verhalten autonom aus der Beobachtung ihrer Umwelt ab
- Verfolgen ein Ziel
- Kommunizieren mit anderen Agenten
- Lernen im Laufe ihres Lebenszyklus dazu

Einleitung

- Ziel des Vortrags: Einblick in maschinelles Lernen im Robotfußball (speziell: Simulationsliga)
- Inhalt orientiert am RoboCup-Team *Karlsruhe Brainstormers*
- Brainstormers waren bei vielen Meisterschaften erfolgreich

Einleitung

Konzept der Karlsruhe Brainstormers:

- Verhalten der einzelnen Spieler nicht fest programmiert
- Fähigkeiten werden von den Spielern selbständig erlernt und verbessert
- Analog zum menschlichen Lernen durch wiederholtes Spielen gegen die eigene und gegen fremde Mannschaften
- Maschinelles Lernen durch Reinforcement-Learning

Agenda

Einführung in Reinforcement-Learning

- Agent soll lernen, sich auf Basis eigener, gewonnener Erfahrungen in einer unbekannten Umwelt möglichst optimal zu verhalten.

- Trainingssignal nach Sequenz von Entscheidungen: Erfolg oder Misserfolg
- Drei Anforderungen an ein Reinforcement Learning-Problem:
  - Zustand der Umwelt erfassbar
  - Umweltzustand manipulierbar
  - Ziel definiert

Einführung in Reinforcement-Learning

- Umwelt besteht aus Zustandsraum
- Positive Endzustände repräsentieren das Ziel
- Agent verfügt über Set von Aktionen
- Agent wählt Aktion, Umwelt ändert sich daraufhin
- Jede Aktion verursacht Kosten
- Ziel soll mit minimalen Kosten erreicht werden

Einführung in Reinforcement-Learning

- Ausprobieren, wie Agent sein Ziel mit minimalen Kosten erreichen kann
- Falls zufällig Ziel erreicht, so war letzte Aktion erfolgreich
- Nicht unbedingt gesamt optimal, Fehler könnten wieder ausgeglichen worden sein
- Falls erneut in vorletztem Zustand vor Ziel, so ist Ziel führende Aktion bekannt (Wissen)

Einführung in Reinforcement-Learning

- Außerdem auch vorletzte Aktion erfolgreich
- Rückwärtsgerichtete Bewertung eines Ziel führenden Verhaltens (in hinreichend vielen Durchläufen)
- Falls in bekanntem Zustand, entweder bewährte Aktion anwenden (*exploit*) oder neue Aktion ausprobieren (*explore*)
- Exploration ermöglicht, optimales Verhalten zu entwickeln

Einführung in Reinforcement-Learning

- In positivem Endzustand können Kosten für hinführende Aktionen zurückgerechnet werden
- Für jeden Zustand entsteht dadurch eine sich verfeinernde Kostenschätzung für die zukünftigen, über alle Folgezustände bis zum Ziel kumulierten, Kosten.
- Für Entscheidung über Aktion werden Folgezustände aller verfügbaren Aktionen berechnet und bewertet
- Bei Exploitation Auswahl der Aktion mit „günstigstem“ Folgezustand

Agenda

Architektur der Brainstormers-Agenten

Modularer Ansatz zur Komplexitätsreduktion

- Sensorverarbeitung

  Aufgabe: Einschätzung des Umweltzustands

- Entscheidungsfindung

  Aufgabe: Auswahl der Aktionen eines Agenten auf Basis des erkannten Umweltzustands

Architektur der Brainstormers-Agenten

- Forschungsschwerpunkt der meisten Teams ist die Entscheidungsfindung

- Brainstormers haben die Implementierung der Sensor-verarbeitung von einen anderen Team (CMU) übernommen

Hypothese: Sensorverarbeitung in Simulationsliga ausgereizt (Komplexität gering)

kein wichtiger Konkurrenzvorteil (mehr)

Im Folgenden wird die Entscheidungsfindung betrachtet.

Architektur der Brainstormers-Agenten

Komplexität der Entscheidungsfindung hat drei wesentliche Quellen:

Zustandsraum der Umwelt

22 Spieler und der Ball: 5400^{23} mögliche Stellungen (dazu Geschwindigkeit und Ausdauer)

Aktionsraum der Agenten

300^{11} Aktionen pro Team in jedem Zyklus (ohne Ballbesitz)

Jeder Zyklus (100 ms) des SoccerServers erfordert neue Entscheidungen.

Architektur der Brainstormers-Agenten

Zur Komplexitätsbeherrschung weitere Zerlegung der Entscheidungsfindung in zwei Subprobleme:

- Moves

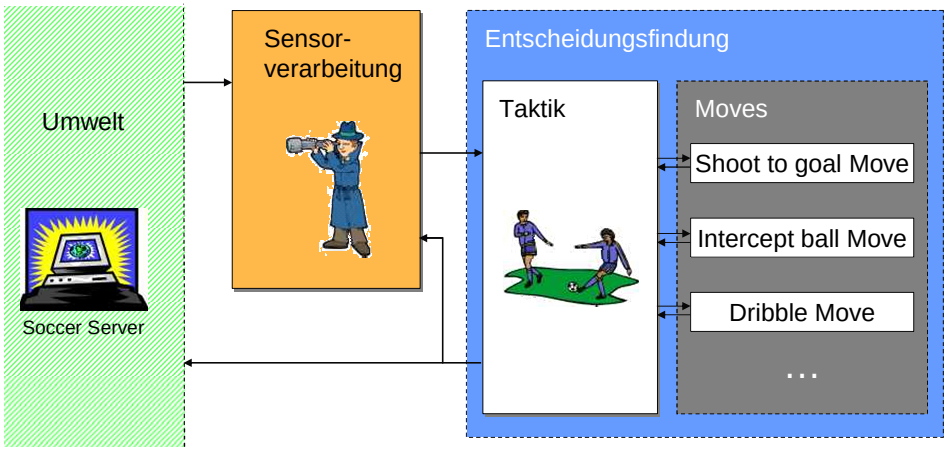
  individuelle Fähigkeiten der einzelnen Agenten (passen, dribbeln, schießen, …)

  Sequenz von Aktionen an den Soccer Server

- Taktik

  Mannschaftsspiel des Teams auf Basis der Moves

Architektur der Brainstormers-Agenten



Quelle: vgl. RMH+03, S. 9

Agenda

A. Einleitung

B. Methodik der Karlsruhe Brainstormers

C. Anwendung des Reinforcement-Learnings
- 1. Lernen von Moves
- 2. Lernen von taktischen Entscheidungen

D. Fazit und Ausblick

Lernen von Moves

Move: Sequenz von Aktionen (z. B. *kick*, *dash*, *turn*) an den Soccer Server

- Überführt einen Anfangszustand in mehreren Zyklen in einen Endzustand
- Vorgegeben: - mögliche Anfangszustände
  - zu erreichendes Ziel
- Ein Move endet, wenn ein möglicher Endzustand erreicht wurde.
- Die Endzustände können positiv/erwünscht oder negativ/nicht erwünscht sein.

Lernen von Moves

- Move-bezogene Konzepte werden von anderen Teams zumeist manuell fest implementiert.
- Die Feinabstimmung der Aktionen und ihrer Parameter ist manuell sehr aufwändig und nicht flexibel.
- Brainstormers: dynamisches Optimierungsproblem
- Aufgabe: automatisiertes Lernen einer Strategie, die mit minimalen Gesamtkosten in möglichst vielen Startzuständen zum Ziel führt.

Lernen von Moves

- Jede Aktion verursacht Kosten, für jede Aktion gleich hoch
- Die Gesamtkosten hängen folglich linear von der Zahl der benötigten Aktionen ab.
- „Kürzere“ Strategien haben niedrigere Gesamtkosten und werden bevorzugt.
- Negativer Endzustand mit hohen Strafkosten belegt

„Kürzester Pfad“-Problem zwischen Anfangs- und gewünschtem Endzustand. Kürze = niedrige Kosten

Lernen von Moves

Problem: Mangelnde Berechenbarkeit der Kostenfunktion – große, prinzipiell unendliche, Zustands- und Aktionsräume

Lösung: Näherung der Kostenfunktion durch ein künstliches neurales Netz (KNN).

Input: aktueller Zustand

Output: erwartete, kumulierte, zukünftige Kosten im Zustand

Vorteile: - Anwendbar auf stetige Räume
- Generalisierbar für neue Inputs

Lernen von Moves

Lernvorgang:

1. Start in einem zufälligen, für den Move gültigen, Startzustand
2. Auswahl einer Aktion durch den Agenten
3. Beobachtung des Nachfolgezustands der Umwelt durch den Agenten
4. Aktualisierung der Kostenschätzung für den Vorgängerzustand

Lernen von Moves

2. Auswahl einer Aktion durch den Agenten

Zwei Optionen:

- Exploitation: Auswahl der - nach bisherigem Wissen - kostenoptimalen Aktion
  Für jede mögliche Aktion wird der resultierende Nachfolgezustand berechnet und bewertet.
- Exploration: Zufälliges Ausprobieren einer neuen Aktion, um mögliche, bessere Strategien zu finden

Ausgewogenes Verhältnis zwischen beiden Optionen notwendig

Lernen von Moves

4. Aktualisierung der Kostenschätzung für den Vorgängerzustand

Die erwarteten, kumulierten, zukünftigen Kosten im Vorgängerzustands werden neu geschätzt.

Summe aus:
- erwarteten, kumulierten, zukünftigen Kosten des Nachfolgezustands
- Kosten der durchgeführten Aktion

Anpassung der Gewichte im KNN, um Abweichung zwischen altem und neuem Output zu minimieren

Lernen von Moves

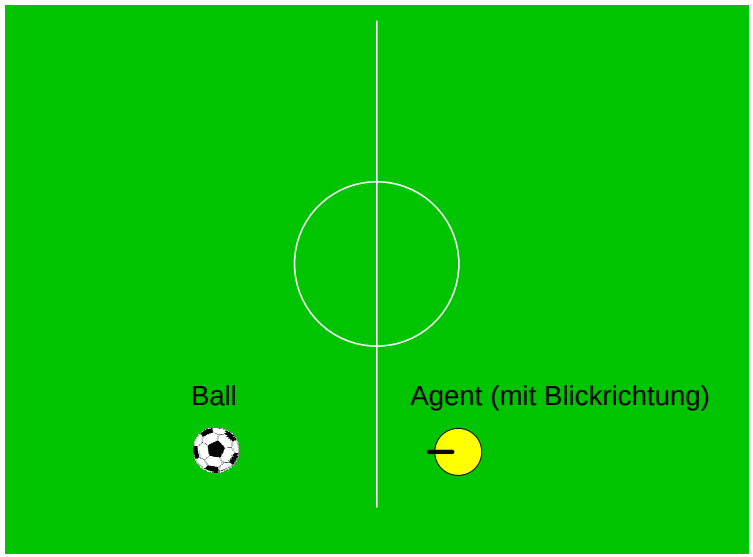
Demonstration des Lernvorgangs an zwei Beispielen:

- Lernen des Moves *intercept ball*
- Lernen des Moves *shoot to goal (kick)*

Beispiel: Lernen des Intercept ball-Moves

- Ziel: Ein Agent versucht, einen rollenden Ball abzufangen
- Mögliche Aktionen:
  - *turn*
  - *dash* (Stürmen)

Beispiel: Lernen des Intercept ball-Moves

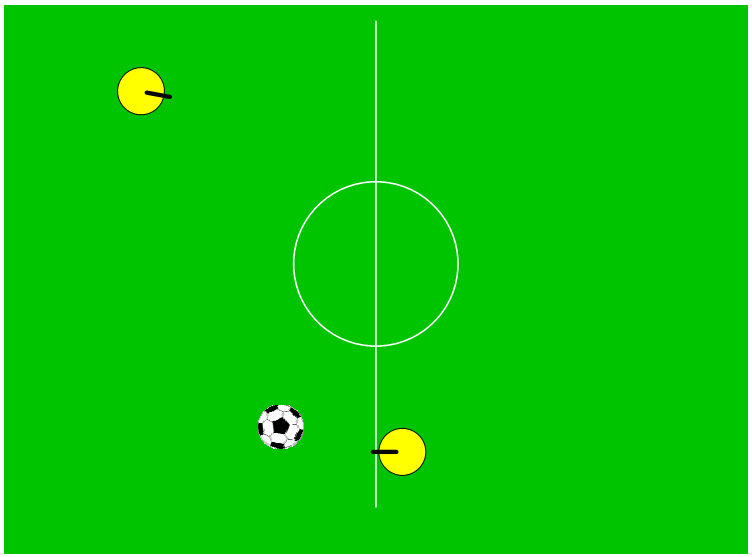


Beispiel: Lernen des Intercept ball-Moves

Interpretation:

- Agent weiß, wo der Ball ist, erreicht ihn aber nicht
- Er wählt zufällig Aktionen aus dem möglichen Aktionsraum aus.
- Kein gelerntes Wissen, da noch kein positiver Zielzustand erreicht wurde (den Ball abfangen)

Beispiel: Lernen des Intercept ball-Moves

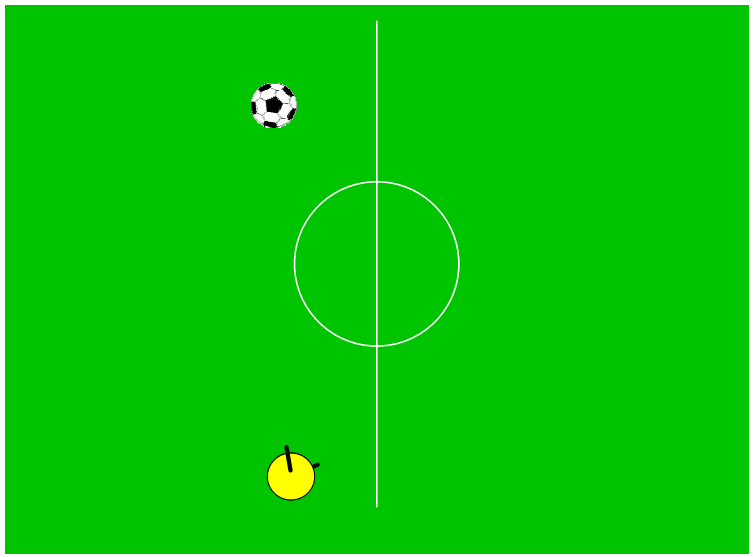


Beispiel: Lernen des Intercept ball-Moves

Interpretation:

- Agent schafft es häufiger, Ball abzufangen
- Für jeden besuchten Zustand können die erwarteten, kumulierten, zukünftigen Kosten zurückgerechnet werden.
- Der Agent kann den Ball zwar abfangen, aber die Gesamtkosten sind noch sehr hoch.
- Exploration neuer Aktionen optimiert die Gesamtkosten

Beispiel: Lernen des Intercept ball-Moves



Beispiel: Lernen des Intercept ball-Moves

Interpretation:

- Der Agent geht direkt auf einen Abfangkurs zum Ball.
- Die Kostenfunktion für den Move ist bestimmt, eine optimale Strategie wurde gelernt
- Die Lernphase ist nun vorbei. In jedem Zustand wird die optimale Aktion auf Basis der Kosten in den möglichen Nachfolgezuständen ausgewählt.

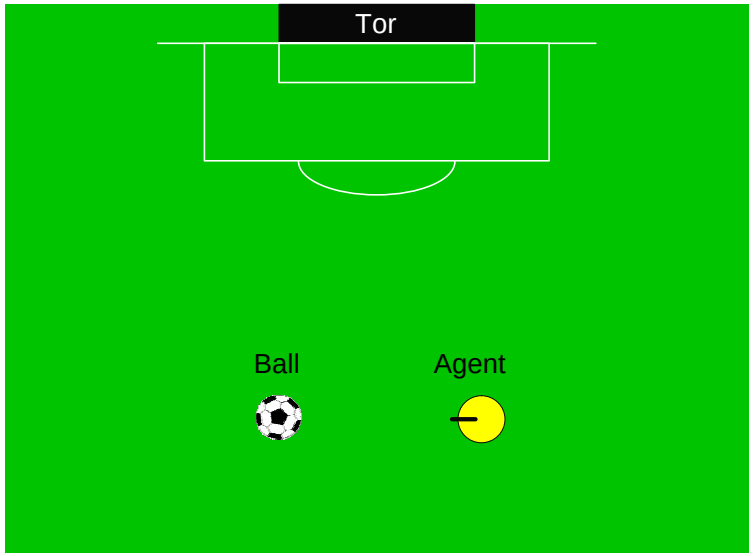
Beispiel: Lernen des Kick-Moves

- Ziel: Ein Agent versucht, einen auf ihn zurollenden Ball ins Tor zu schießen.
- Mögliche Aktionen: - *turn (Richtung: 36 Möglichkeiten)*

  - *kick (Richtung: 100 Möglichkeiten, Kraft: 5 Möglichkeiten)*

  insgesamt 536 mögliche Aktionen

Beispiel: Lernen des Kick-Moves

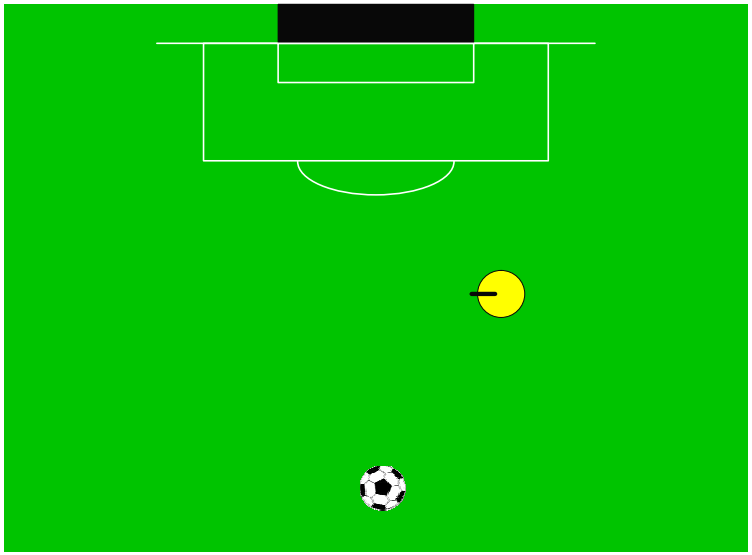


Beispiel: Lernen des Kick-Moves

Interpretation:

- Der Agent schießt den Ball zufällig irgendwohin.
- Kein Fortschritt beim Lernvorgang, bis ein positiver Zielzustand erreicht wird

Beispiel: Lernen des Kick-Moves

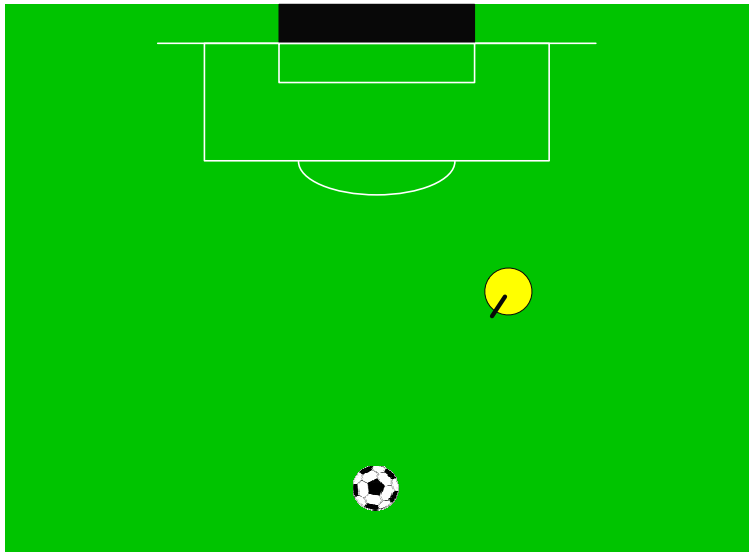


Beispiel: Lernen des Kick-Moves

Interpretation:

- Der Agent trifft häufiger das Tor.
- Für jeden besuchten Zustand können die erwarteten, kumulierten, zukünftigen Kosten zurückgerechnet werden.
- Der Agent trifft zwar das Tor, aber nicht auf dem kürzesten Weg.

Beispiel: Lernen des Kick-Moves



Beispiel: Lernen des Kick-Moves

Interpretation:

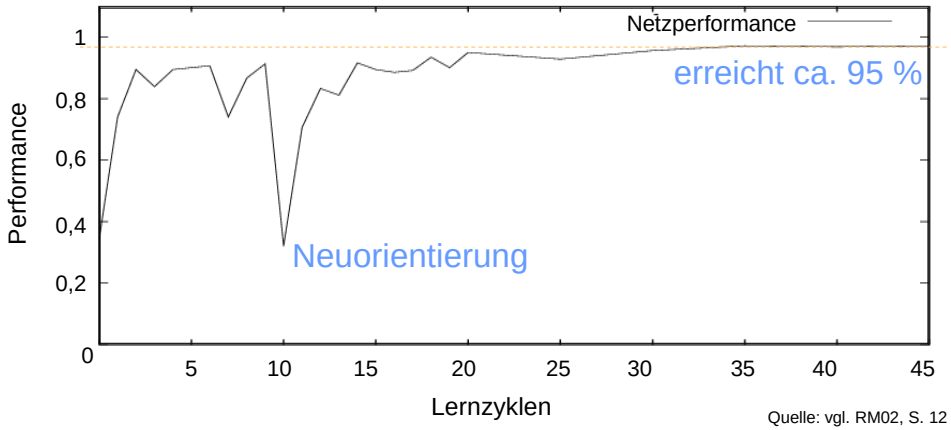
- Der Agent schießt den Ball auf dem kürzesten Weg ins Tor.
- Dabei dreht er sich, um eine höhere Präzision des Schusses zu erreichen.
- Der Lernvorgang ist abgeschlossen.

Beispiel: Lernen des Kick-Moves

Eingesetztes KNN:

- Backpropagation Network
- 4 Input-Neuronen für Ort und Geschwindigkeit des Balls
- 1 Output-Neuron für den Kostenfunktionswert
- 20 verborgene Neuronen für das nicht lineare Mapping zwischen Input und Output
- Drei verschiedene KNN für drei gewünschte Schuss-Geschwindigkeiten eingesetzt

Performance beim Lernen des Kick-Moves



Quelle: vgl. RM02, S. 12

Jeder Lernzyklus besteht aus 2.000 unterschiedlichen Startzuständen.

Lernen von Moves

Gelernte Moves:

- *intercept ball*
- *go in direction* (acht Richtungen)
- *dribble in direction* (acht Richtungen)
- *shoot to goal (kick)* (drei Geschwindigkeiten)
- *pass ball to teammate* (bis zu zehn Mitspieler)
- *wait at position*

= 31 Moves insgesamt

Agenda

A. Einleitung

B. Methodik der Karlsruhe Brainstormers

C. Anwendung des Reinforcement-Learnings
1. Lernen von Moves
2. Lernen von taktischen Entscheidungen

D. Fazit und Ausblick

Lernen von taktischen Entscheidungen

- Ein einzelner Move führt (normalerweise) nicht direkt zu einem Tor.

- Eine Mannschaft ist nur erfolgreich, wenn alle Spieler miteinander kooperieren. Erfordert Koordination von elf Agenten

- Zur Komplexitätsreduktion nur Moves als Aktionen der Agenten möglich

- Gemeinsames Oberziel: Gewinn des Spiels
  Unterziele: eigene Tore schießen, fremde Tore vermeiden

Lernen von taktischen Entscheidungen

- Beschreibungsmodell wird auf Multi-Agenten-Fall ausgeweitet

- MMDP (Multi-agent Markov Decision Process) als Tupel
  $M_n := [S, A, r, p]$

- Für jeden Agenten gibt es ein Set an Aktionen
  (*A* wird zum kartesisches Produkt dieser Sets)

- Für jeden Agenten gibt es ein eigenes Reinforcement-Signal
  r wird zu Tupel von Abbildungen $r_i : S \times A \to \mathbb{R}$, für $i \in \{1,...,n\}$

Lernen von taktischen Entscheidungen

- Reinforcement-Signal kann folglich von Aktionen anderer Agenten abhängen

- Im allgemeinen Fall unkorrelierter Reinforcement-Signale kein Optimalitätsmaß definierbar

Betrachtung zweier Spezialfälle:

- Kooperation durch gemeinsames Reinforcement-Signal

- Opponierendes Verhalten durch gegensätzliches Reinforcement-Signal („Nullsumme“)
  Erfolg des einen Agenten wird zum Misserfolg des anderen

Lernen von taktischen Entscheidungen

- Für die Spezialfälle können die Existenz optimaler Strategien nachgewiesen und Kostenfunktionen aufgestellt werden

- Fußball wird modelliert als Nullsummen-Spiel zweier untereinander kooperierender Teams

- Problem der Berechenbarkeit optimaler Strategien

- Lokal optimales Verhalten muss keine optimale Gesamtstrategie bilden

- Daher Verhalten zur Zeit noch regelbasiert mit Prioritäten für einzelne Moves realisiert

Lernen von taktischen Entscheidungen

Im Folgenden werden zwei Ansätze vorgestellt, mit denen die Brainstormers versuchen, taktische Entscheidungen zu lernen.

Für beide Ansätze gilt:

- Positiver Endzustand: erzieltes Tor
- Negativer Endzustand: Gegner erlangt Ball (maximale Kosten)

- gleiche, konstante Kosten für alle Agenten
- Führt zu kooperativem Verhalten, denn Solospiel kaum Erfolg versprechend und höhere Gesamtkosten

Lernen von taktischen Entscheidungen

1. Ansatz: Joint Action Learners

- Jederzeit Kenntnis über Aktionen aller Agenten
- Agenten haben nur Einfluss auf ihren Teil aller Aktionen
- Aus Beobachtung hinreichend vieler Zustandsübergänge Bestimmung erwarteter, kumulierter, zukünftiger Kosten für Zustand wie bei Moves
- Ableitung individueller Strategien durch Projektionen der optimalen Gesamtstrategie

Lernen von taktischen Entscheidungen

Probleme mit Joint Action Learners:

- Möglicher Aktionsraum wächst exponentiell mit Anzahl beteiligter Agenten

  Lernvorgänge der KNN konvergieren nicht mehr

- Verfahren ist an eine fixe Anzahl Agenten gebunden

  Kein „Aushelfen“ bei Bedarf

Beispiel: Joint Action Learners

- Ziel: Zwei Agenten versuchen gegen einen Verteidiger ein Tor zu schießen.

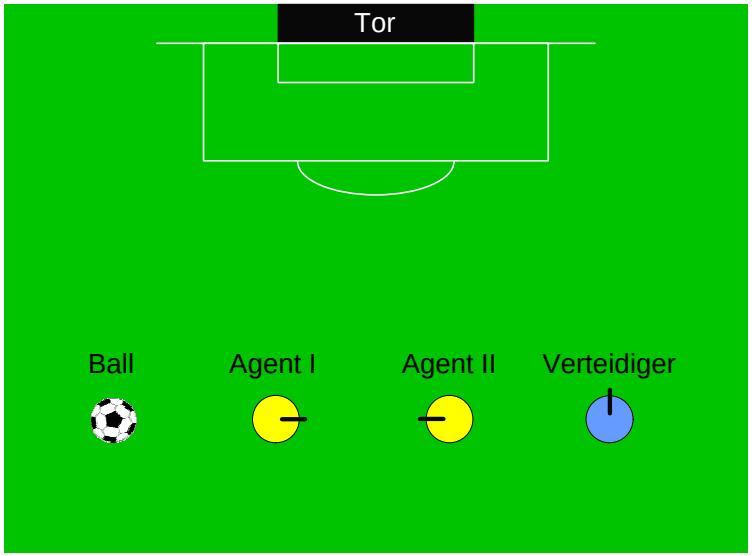
- Für jeden Move wurde ein eigenes KNN eingesetzt.

  Input: - Positionen der Spieler und des Balls

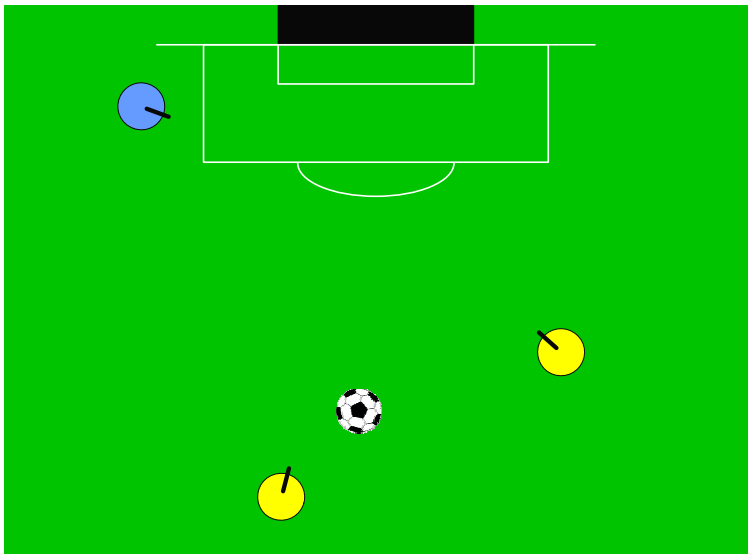
  - Geschwindigkeit des Balls

- Der Verteidiger geht immer direkt zum Ball, aber seine Startposition variiert.

Beispiel: Joint Action Learners



Beispiel: Joint Action Learners

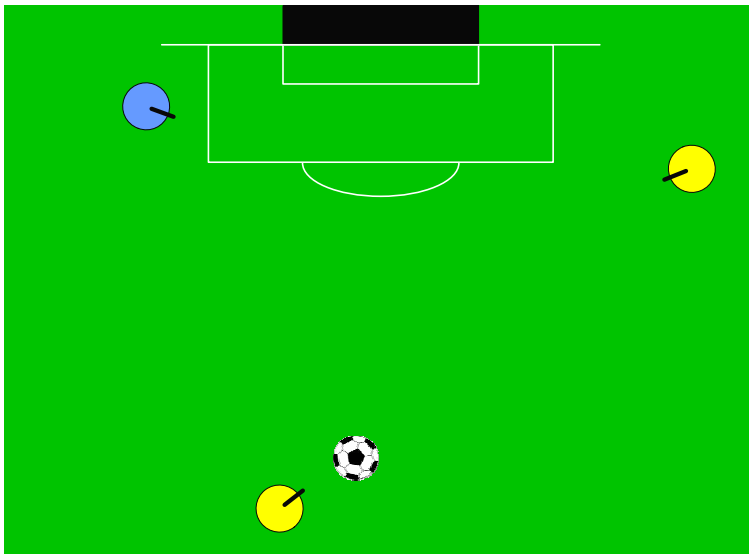


Beispiel: Joint Action Learners

Interpretation:

- Die Agenten versuchen, direkt ein Tor zu schießen.
- Scheinbar der einfachste Weg, ein Tor zu schießen, aber der Verteidiger fängt den Ball meistens ab.

Beispiel: Joint Action Learners

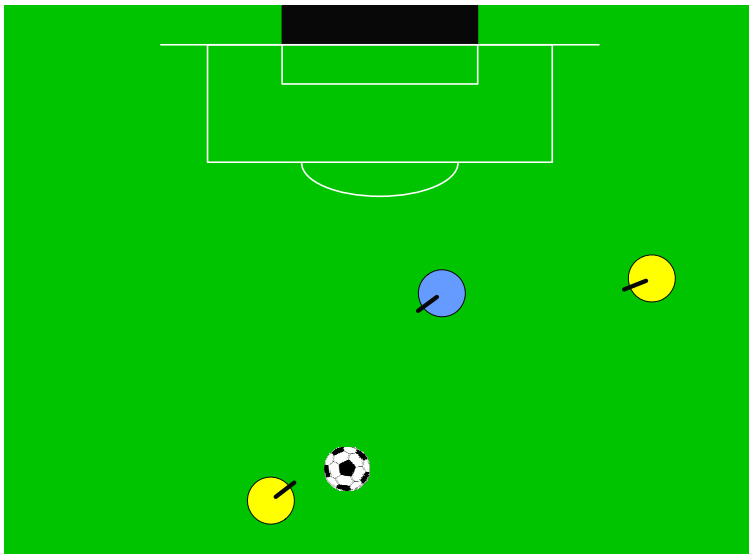


Beispiel: Joint Action Learners

Interpretation:

- Die Agenten haben gelernt, nur dann auf das Tor zu schießen, wenn der Verteidiger den Ball nicht abfangen kann.
- Andernfalls wird der Ball zum Mitspieler gepasst.
- Aber: Taktik nicht erfolgreich, falls der Verteidiger in der Mitte zwischen beiden Agenten steht.

Beispiel: Joint Action Learners



Beispiel: Joint Action Learners

Interpretation:

- Die Agenten haben gelernt, einen Doppelpass zu spielen.
- Der Verteidiger erreicht den Ball nur noch selten.
- Der Lernprozess ist abgeschlossen.

Beispiel: Joint Action Learners

Bewertung des Experiments:

- Erfolgsquote stieg von 35 % auf 85 % gegen einen Verteidiger, bzw. von 10 % auf 55 % bei zwei Verteidigern
- Kooperatives Spielverhalten erzeugt (Doppelpässe)
- Es ist aber nicht gelungen, die Ergebnisse auf weitere Agenten zu verallgemeinern
- Aufgrund der exponentiell wachsende Aktionsräume konvergierten die Lernvorgänge nicht mehr

Lernen von taktischen Entscheidungen

2. Ansatz: Independent Learners

- Schwächerer Agententyp als Joint Action Learners
- Nur Kenntnis über eigene Aktionen
- Andere Agenten werden nicht wahrgenommen
- Einfluss der anderen Agenten kann nicht von stochastischen Einflüssen der Umwelt unterschieden werden
- Keine theoretische Garantie mehr, dass eine optimale Taktik gefunden werden kann

Lernen von taktischen Entscheidungen

Lernvorgang:

1. Leere Sammlung von Beispielen (Zustand + Kostenschätzung)
2. Zufällige Anfangstaktik. Speicherung der Moves, bis zehn erfolgreiche Sequenzen gefunden wurden.
3. Zustände entlang der erfolgreichen Sequenzen werden, ohne Duplikate, in die Beispielsammlung eingefügt
4. Training eines KNN mit der Beispielmenge
   Input: Positionen Spieler + Ball, Geschwindigkeit des Balls

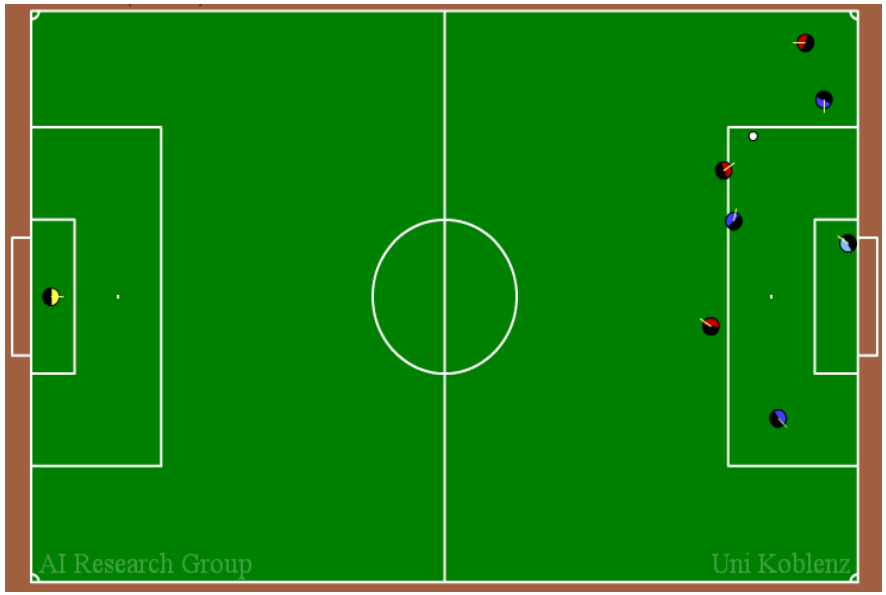
5.000 Iterationen des Lernvorgangs (Schritte 2 - 4)

Beispiel: Independent Learners

- Ziel: Drei Agenten versuchen gegen drei Verteidiger und einen Torwart ein Tor zu schießen

- Gelernt wird die Positionierung der Spieler ohne Ballbesitz bzw. -nähe (frei stehen für Pässe, Räume öffnen, …)
- Verhalten für Spieler im Ballbesitz oder in Ballnähe regelbasiert mit Prioritäten für anzuwendende Moves (hybrider Ansatz)
- Verteidiger sind vollständig regelbasiert (bewährte Brainstormers-Agenten)

Beispiel: Independent Learners



Beispiel: Independent Learners

Bewertung des Experiments:

- Der Lernalgorithmus führte zu einer signifikanten Verbesserung des Sturms.
  Erfolgsquote gegen die bewährte Verteidigung stieg von 19 % auf 28 %
- Gelerntes Verhalten erwies sich als generalisierbar gegenüber nicht trainierten Startzuständen und anderen Gegner-Teams.
- Hybrider Ansatz aus gelernten und festen Teilen erfolgreich

Agenda

Fazit und Ausblick

- Ergebnisse sind manueller, fester Implementierung überlegen.
- Gelerntes Verhalten in KNN lässt sich generalisieren,
  z. B. auf neue Gegner
- Maschinelles Lernen für einzelne Agenten funktioniert gut
- Für den verteilten Fall existieren bislang nur Lern-Ansätze
- Erfolge mit hybridem Ansatz gelernter und statischer Komponenten.

Fazit und Ausblick

- Nächstes Ziel ist die Einbeziehung von Kommunikation in die Kooperation der Agenten
- Diese Mischung aus Joint Action und Independent Learners erlaubt Kommunikation der Aktionen und weiteren Absichten

Abschließende Bewertung:

- Anwendung von Reinforcement Learning-Methoden ist sinnvoll und Erfolg versprechend.
- Das zugrunde liegende Konzept ist vielfältig einsetzbar und nicht auf das Fußballspiel beschränkt.

Literaturquellen im Vortrag

[RM02] Riedmiller, Martin; Merke, Artur: *Using Machine Learning Techniques in Complex Multi-Agent Domains*. In: Perspectives on Adaptivity and Learning. Hrsg.: Stamatescu, I.; Menzel, W.; Richter, M.; Ratsch, U. LNCS, Springer, 2002.

[RMH+03] Riedmiller, Martin; Merke, Artur; Hoffmann, Andreas; Withopf, Daniel; Nickschas, Manuel; Zacharias, Franziska: *Brainstormers 2002 - Team Description*. In: RoboCup 2002: Robot Soccer World Cup VI, LNCS, Springer, (noch nicht erschienen). http://lrb.cs.uni-dortmund.de/~riedmill/publications/riedml.merke.robocup02.ps.gz [Abrufdatum: 30.04.2003]