Airhockey auf übermenschlichen Level mit Deep Reinforcement Learning

Rico Steinke, Pascal Graf und Nicolaj C. Stache
Automotive Systems Engineering, Heilbronn University of Applied Sciences

Projektübersicht

Motivation:

- Demonstrator zeigt KI im Bereich Robotik, die ohne menschliches Expertenwissen in der Simulation trainiert und dann in der Realität nutzbar gemacht wird.
- Handlungsheuristik für komplexe Szenarien fast unmöglich manuell zu programmieren. Deshalb trainiert ein Software-Agent beim Reinforcement Learning im Spiel gegen sich selbst, Spielstrategien zu optimieren.

Ansatz / Ziele:

- Künstliche Intelligenz lernt gegen sich selbst, das Reaktionsspiel Airhockey zu meistern. Luftgelagerter Puck soll dabei mit eigenem Pusher ins gegnerische Tor geschossen werden.
- Agent erhält zu jedem Zeitschritt eine numerische Belohnung, welche über hunderttausende Spieleepisoden maximiert wird.

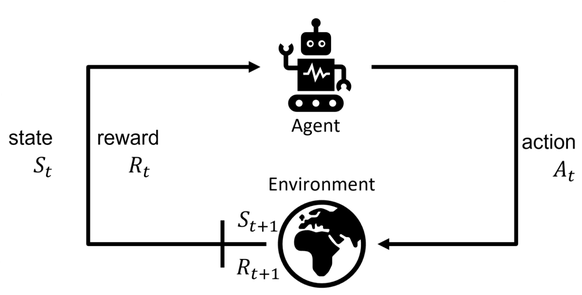

Abbildung 1: Reinforcement Learning Prinzip: Der Agent für eine Aktion in seiner Umgebung aus und erhält als Rückmeldung einen neuen State zusammen mit einem numerischen Reward.

Übertragbarkeit auf reale Aufgaben:

- Selbes Prinzip lässt sich im Bereich industrielle Produktion und autonomes Fahren anwenden. Beispiele hierfür sind:
  - ➢ Anlagenoptimierung zur effizienteren Nutzung von Zeit, Ressourcen, Energie abhängig von Umwelteinflüssen
  - ➢ Handling komplexer Objekte in der Robotik (Griff in die Kiste).
  - ➢ Automatisches Bremsen und Beschleunigen eines Fahrzeugs in Reaktion auf unvorhergesehene Ereignisse

Airhockey-Simulation

- Simulation basiert auf der Spieleengine *Unity3D* [1] in Kombination mit hochrealistischer Physik-Library *MuJoCo* (Multi-Joint dynamics with Contact) [2].
- *MuJoCo* ermöglicht genaue Physik-Parametrisierung (z.B. Reibung, Abprallen) ➔ bessere Übertragbarkeit auf die Realität.
- Mehrere Spielinstanzen werden zeitlich beschleunigt gleichzeitig simuliert, pro Spielepisode vergehen nur zwei Sekunden Realzeit.

Abbildung 2: 3D-Airhockey-Simulation aus Spielerperspektive (links). Das User Interface zeigt Informationen zum Reinforcement Learning, der Steuerung sowie der Zusammensetzung der Belohnung, welche der Agent erhalten hat. 3D Modell des realen Airhockey-Tischs ohne Hintergrund gerendert (rechts).

Reales Modell

- Realer Demonstrator ist mit Kinematik ausgestattet, die über Schrittmotoren hochpräzise sowie dynamische Ansteuerung des Pushers erlaubt.
- High-Speed-Kamera über dem Feld angebracht, die das Spielfeld 150 mal pro Sekunde aufnimmt.
- Positionen & Geschwindigkeiten relevanter Spielelemente (Pusher und Puck) werden aus Bildern extrahiert.
- KI benötigt Informationen, um sinnvolle Entscheidungen treffen zu können.

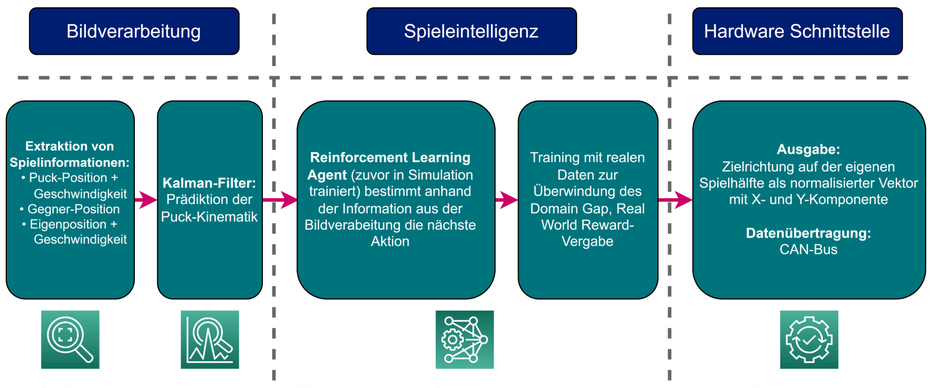

Abbildung 3: Programmablauf von der Extraktion der Spielinformation mittels klassischer Bildverarbeitung, über das Treffen einer Bewegungsentscheidung mit Reinforcement Learning (Spieleintelligenz) bis zur Bewegung am realen Airhockey-Tisch (Hardware-Schnittstelle).

Reinforcement Learning

Soft-Actor-Critic-Algorithmus [3]:

- Reinforcement Learning Algorithmus basiert auf Einsatz von zwei tiefen neuronalen Netzwerken.
- *Actor* bestimmt die nächste Aktion des Agents, also Bewegungsrichtung- und Geschwindigkeit des Pushers.
- *Critic* schätzt Qualität der gewählten Aktionen anhand des bisher erhaltenen Rewards ab und verbessert so implizit den *Actor.*
- KI muss abwägen, ob Spielstrategie verfolgt wird, die sie favorisiert, oder ob Neues ausprobiert wird, um eventuell bessere Strategien zu erforschen (Exploration-Exploitation-Dilemma).

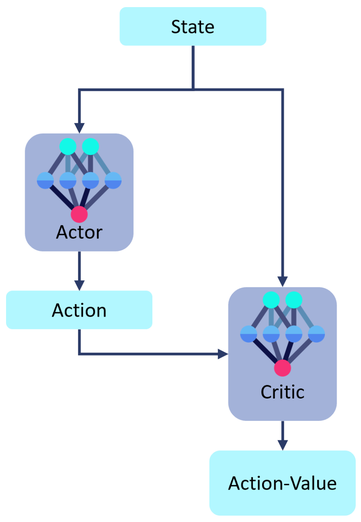

Abbildung 4: Schematische Darstellung der Soft-Actor-Critic Netzwerkarchitektur .

Reward-Zusammensetzung:

- Neben Belohnung bzw. Bestrafung für (Gegen-)Tore, sollen andere Verhaltensweisen mit in Reward-Struktur einfließen.
- Beispielsweise Fahren in Banden, schnelle Änderungen der Bewegungsrichtung oder Dulden des Pucks in der eigenen Spielhälfte.

Trainingsergebnisse

- Belohnung der KI im Spiel hängt von der Stärke des Gegners ab.
- Unabhängige Bewertung durch Rating-System ähnlich ELO-Rating anderer Spiele (z.B. Schach oder Online Multiplayer Games).
- Unterschiedliche Agents (abhängig von Trainingsdauer und -konfiguration) spielen in Turnier gegeneinander.
- Nach 48h Training hat der Agent das durchschnittliche menschliche Spielerniveau erreicht. Bester, bisher ungeschlagener Agent hat Training von 138h (knapp sechs Tagen) absolviert, entspricht etwa 220.000 Spielen.

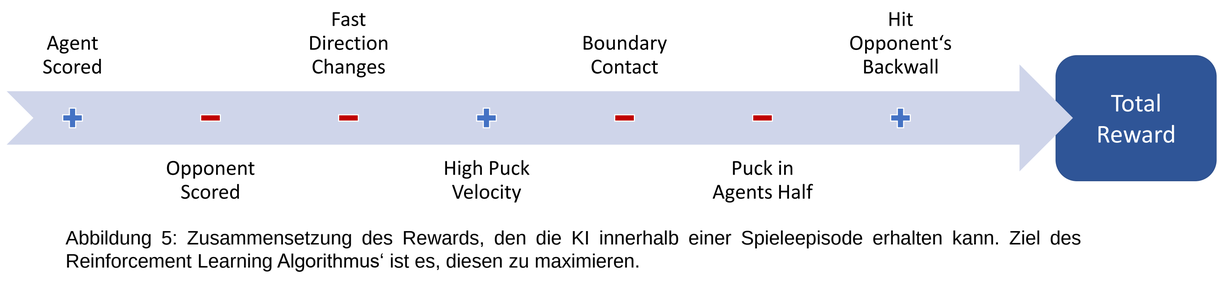

Abbildung 5: Zusammensetzung des Rewards, den die KI innerhalb einer Spieleepisode erhalten kann. Ziel des Reinforcement Learning Algorithmus' ist es, diesen zu maximieren.

Bewegungsanalyse

- Detaillierte Analyse der trainierten KI zeigt, dass Abprallverhalten des Pucks über die Bande prognostiziert und sich so frühzeitig positioniert wird.
- Heatmaps zeigen, an welchen Positionen der Agent sich besonders häufig aufhält.

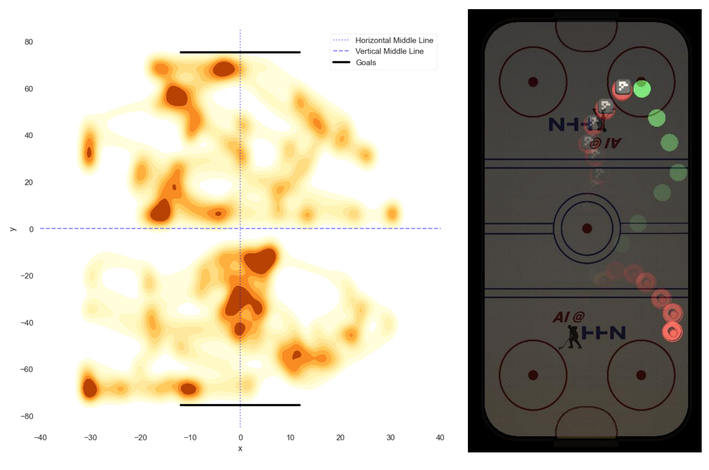

Abbildung 6: Mithilfe der Heatmaps (links) wird visualisiert, an welchen Positionen der Agent sich vorzugsweise aufhält. Die Bewegungstrajektorie (rechts) zeigt, wie der Agent gelernt hat, das Abprallverhalten des Pucks zu prognostizieren und diesen rechtzeitig abzufangen.

Quellen

[1] J. Haas, „A history of the unity game Engine," Worcester Polytechnic Institute, 2014.

[2] E. Todorov, T. Erez, and Y. Tassa, "Mujoco: A physics engine for model-based control," in 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 5026–5033, IEEE, 2012.

[3] T. Haarnoja et al., "Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor," CoRR, 2018.

rsteinke@stud.hs-heilbronn.de, Pascal.Graf@hs-heilbronn.de, Nicolaj.Stache@hs-heilbronn.de