

Reinforcement Learning - Schach

Implementierung im Rahmen des Applied Machine Learning Projekts

A. Bernrieder, J. Brebeck, N. Wichter, and T. Hilbradt

1

Erstellt am 05. Juli 2020

ABSTRACT

Context. Im Rahmen dieses Projekts wird die Implementierung eines Reinforcement Learning Algorithmus zum Erlernen des Schachspiels untersucht.

Ziel der Ausarbeitung ist dabei ein funktionierendes Modell, dass in der Lage ist gegen einen Menschen zu spielen und eine angemessene Herausforderung darzustellen.

Aims. Das Projekt soll eine vereinfachte Version des Reinforcement Learnings zeigen, dass genutzt wird, um das Schachspielen zu erlernen. Dabei wird besonderer Wert auf eine Implementierung des Monte Carlo Tree Search Algorithmus gelegt, der die Basis für das RL bietet.

Methods. Die Basis des Projekts sind die Arbeiten von Deepmind an der AlphaZero AI (Deepmind, 2018), die Vorreiter im Zuge des Reinforcement Learnings waren und sind und mit der aktuellen MuZero AI erneut Standards gesetzt haben. (Schrittwieser, et al, 2020)

Dieses Projekt orientiert sich in seiner Grundstruktur an diesem Vorbild, verwendet also einen Monte Carlo Tree Search zum Suchen der optimalen Züge und greift für die Implementierung des Reinforcement Learnings auf einen Temporal Difference Learning Algorithmus (SARSA) zurück.

Results. Dieses Ziel dieses Projekts konnte in seinen Grundzügen erreicht werden, wobei unklar bleibt inwiefern es gegen menschliche Spieler dauerhaft gewinnen kann. (ELO ZAHL EINFÜGEN @NIKLAS)

Conclusions. Es kann davon ausgegangen werden, dass weitere Verbesserungen am Modellalgorithmus, insbesondere dem Monte Carlo Tree Search zu Leistungssteigerungen führen werden. Ebenso wird eine erhöhte Trainingszeit dem Modell erlauben besser zu spielen.

1. Einführung

Das Reinforcement Learning stellt eine Unterart des Machine Learnings dar. Hierbei erlernt das Programm, Agent genannt, selbständig, also ohne äußeres Einnwirken wie ein bestimmtes Problem zu lösen ist. Hierunter fallen das selbständige Erlernen des aufrechten Gangs eines Roboters, das Betreiben eines autonomen Helikopters und viele andere realweltliche Probleme (Kober, et al, 2013).

Jedoch kann diese Art des Lernens auch verwendet werden, um einer schwachen Künstlichen Intelligenz beizubringen ein bestimmtes Spiel zu spielen. Deepmind hat im Jahr 2016 mit AlphaGo bewiesen, dass ein Programm inzwischen menschliche Spieler auch in komplexen Spielen wie Go, einem asiatischen Brettspiel, schlagen kann, indem der damalige Weltmeister dieses Spiels Lee Sedol von dem Programm 2016 in vier von fünf Partien geschlagen wurde (Spiegel, 2016).

Seitdem haben sich die Algorithmen weiterentwickelt, sodass inzwischen selbst Echtzeitstrategiespiele, wie StarCraft II kein Problem mehr sind (Deepmind, 2019). Durch umfangreiche Veröffentlichungen auf diesem Themengebiet, zahlreichen Tutorials und Online Vorlesungen, wie von David Silver (Silver, 2015) ist es inzwischen auch für Anfänger möglich einfache Reinforcement Learning Algorithmen zu implementieren.

Im Rahmen dieses Projekts soll dies am Beispiel des Schachspiels dargelegt werden. Dieses wurde gewählt, da es intuitiv zugänglich, also schnell erlernbar, ist, sowie eine hohe Spieltiefe bietet. In diesem Report soll beschrieben werden,

welche Algorithmen für die Implementierung verwendet wurden, warum diese gewählt wurden, an welchen Vorlagen sich orientiert wurde und was ein potentieller Business Use-Case für dieses Projekt wäre

2. Implementierung des Algorithmus

Die Basis für dieses Projekt bietet die Arbeit von ArjanGroen in seinen Kaggle Notebooks (ArjanGroen), sowie die Videoreihe zum Thema von David Silver. Prinzipiell werden zwei Algorithmen implementiert, die aufeinander aufbauen, bzw. sich gegenseitig bedingen um ein komplettes Modell zu erstellen. Diese wären zum einen ein Monte Carlo Tree Search Algorithmus, sowie zum anderen ein Temporal Difference Algorithmus, im speziellen der SARSA-Algorithmus.

3. Business Use Case

Das erlernte Modell kann

4. Ausführen

Um das Modell auszuführen, ..

5. Ergebnisse

Die Ergebnisse ...

Acknowledgements. Part of this work was supported by the German *Deutsche Forschungsgemeinschaft*, DFG project number Ts 17/2–1.

References

- Baker, N. 1966, in *Stellar Evolution*, ed. R. F. Stein, & A. G. W. Cameron (Plenum, New York) 333
- Balluch, M. 1988, *A&A*, 200, 58
- Cox, J. P. 1980, *Theory of Stellar Pulsation* (Princeton University Press, Princeton) 165
- Cox, A. N., & Stewart, J. N. 1969, *Academia Nauk, Scientific Information* 15, 1
- Mizuno H. 1980, *Prog. Theor. Phys.*, 64, 544
- Tscharnutter W. M. 1987, *A&A*, 188, 55
- Terlevich, R. 1992, in *ASP Conf. Ser. 31, Relationships between Active Galactic Nuclei and Starburst Galaxies*, ed. A. V. Filippenko, 13
- Yorke, H. W. 1980a, *A&A*, 86, 286
- Zheng, W., Davidsen, A. F., Tytler, D. & Kriss, G. A. 1997, preprint