

Hochschule Darmstadt

– Fachbereich Informatik –

Robustheit und Generalisierbarkeit in algorithmischen und Reinforcement Learning gestützten Lösungsansätzen: Eine Fallstudie mit Vier Gewinnt

Abschlussarbeit zur Erlangung des akademischen Grades
Bachelor of Science (B.Sc.)

vorgelegt von

Leo Herrmann

Matrikelnummer: 1111455

Referentin: Prof. Dr. Elke Hergenröther
Korreferent: Adriatik Gashi

1 Kurzfassung

Inhaltsverzeichnis

1	Kurzfassung	2
2	Einleitung	1
3	Grundlagen	2
3.1	Vier Gewinnt	2
3.2	Symbolische Algorithmen	3
3.2.1	Minimax	3
3.2.2	AlphaBeta	3
3.2.3	MCTS	3
3.3	Reinforcement Learning	4
3.4	Robustheit und Generalisierbarkeit	4
4	Konzept	4
5	Realisierung	5
6	Ergebnisdiskussion	6
7	Zusammenfassung und Ausblick	7
8	Literaturverzeichnis	8

Abbildungsverzeichnis

Eigenständigkeitserklärung

Ich versichere hiermit, dass ich die vorliegende Arbeit selbständig verfasst und keine anderen als die im Literaturverzeichnis angegebenen Quellen benutzt habe. Alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten oder noch nicht veröffentlichten Quellen entnommen sind, sind als solche kenntlich gemacht. Die Zeichnungen oder Abbildungen in dieser Arbeit sind von mir selbst erstellt worden oder mit einem entsprechenden Quellennachweis versehen.

Darmstadt, 21.03.2023

Leo Herrmann

2 Einleitung

Fortschreitende Automatisierung durchdringt zahlreiche Bereiche der Gesellschaft, so zum Beispiel die Fertigungsindustrie, das Gesundheitswesen oder den Straßenverkehr. Zwei fundamentale Ansätze sind dabei regelbasierte Algorithmen und Machine Learning. Die Einsatzbedingungen von Automatisierungssystemen unterscheiden sich häufig von den Bedingungen, unter denen sie entwickelt und getestet werden. Häufig müssen Systeme mit fehlerhaften oder veralteten Informationen arbeiten oder es treten Situationen ein, die bei der Konzipierung der Systeme nicht berücksichtigt werden können. Dabei sinkt die Leistungsfähigkeit dieser Systeme.

Im Rahmen dieser Arbeit werden Robustheit und Generalisierbarkeit eines algorithmischen Ansatzes und eines Reinforcement Learning basierten Ansatzes zur Lösung des Brettspiels „Vier Gewinn“ untersucht. Bei Robustheit und Generalisierbarkeit handelt es sich um Eigenschaften, die beschreiben, wie gut ein Algorithmus oder RL-Modell in der Praxis funktioniert, in der andere Bedingungen herrschen können als während der Entwicklung und Qualitätssicherung. Diese Kriterien sind besonders relevant für den Erfolg von Algorithmen und Modellen in der Praxis.

Spiele eignen sich zur Untersuchung von Algorithmen und Modellen, weil sie reale Probleme auf kontrollierbare Umgebungen abstrahieren und gleichzeitig reproduzierbare und vergleichbare Messungen ermöglichen. Die Untersuchungen dieser Arbeit erfolgen am Beispiel des Brettspiels „Vier Gewinn“, da aus früheren Untersuchungen ersichtlich wird, dass sich sowohl algorithmische als auch Reinforcement Learning basierte Lösungen eignen.

Es wird Grundlagenforschung zu verbreiteten algorithmischen Ansätzen und Reinforcement Learning basierten Ansätzen betrieben. Anschließend werden die Aspekte Robustheit und Generalisierbarkeit von zwei Ansätzen aus den jeweiligen Bereichen am Beispiel von Vier Gewinn empirisch untersucht. Dabei werden neue Erkenntnisse über Lösungsansätze von Vier Gewinn gewonnen, die sich auf vergleichbare Szenarien in der realen Welt übertragen lassen.

Die zentrale Fragestellung lautet: Inwiefern sind bei Vier Gewinn algorithmische oder Reinforcement Learning basierte Ansätze robuster oder besser generalisierbar? Das Ziel dieser Arbeit besteht darin, ein detailliertes Verständnis über verschiedene Aspekte von Robustheit und Generalisierbarkeit der zu untersuchenden Ansätze zu bekommen.

3 Grundlagen

In diesem Kapitel wird durch Literaturrecherche eine fundierte theoretische Basis geschaffen, auf die im weiteren Verlauf dieser Arbeit Bezug genommen wird. Zunächst wird Vier Gewinnt als zu lösendes Problem untersucht und eingeordnet. Anschließend folgt eine Auswahl von jeweils einem algorithmischen und einem Reinforcement Learning basiertem Lösungsansatz. Die Funktionsweise beider Lösungsmethoden wird erklärt. Außerdem werden bestehende Theorien und Definitionen zum Thema Robustheit und Generalisierbarkeit zusammengetragen. Sie bilden die Grundlage für die Szenarien und Bewertungskriterien in den Experimenten des Hauptteils.

3.1 Vier Gewinnt

Bei Vier Gewinnt handelt es sich um ein kombinatorisches Nullsummenspiel für zwei Spieler. Kombinatorische Spiele weisen „perfekte Information“ auf. Das bedeutet, dass alle Spieler zu jeder Zeit den gesamten Zustand des Spiels kennen. Bei kombinatorischen Spielen sind außerdem keine Zufallselemente enthalten. Die einzige Herausforderung beim Spielen kombinatorischer Spiele besteht darin, unter einer Vielzahl von Entscheidungsoptionen diejenige auszuwählen, die den besten weiteren Spielverlauf verspricht([4], S. 96-100)([6], Kapitel 4.1).

Nullsummenspiel bedeutet, dass der Gewinn eines Spielers zwangsläufig einen Verlust des anderen Spielers verursacht. Die beiden Spieler haben also entgegengesetzte Interessen. Zwei-Personen-Nullsummenspiele sind insofern interessant für die Untersuchung von ..., weil - Zwei Personen über Mehr Personen -> Wechselwirkungen - Zwei Personen über Eine Person - Null

Das Spiel besteht aus einem 7 x 6 Spielfeld. Beide Spieler dürfen abwechselnd einen Spielstein in eine Spalte hinwerfen, der in dieser Spalte bis zur untersten freien Position fällt. Es gewinnt der Spieler, der als erstes vier Spielsteine in einer horizontalen, vertikalen oder diagonalen Reihe nebeneinander stehen hat[5].

Vier Gewinnt und verschiedene Lösungsverfahren sind bereits ausgiebig untersucht. Das Spiel wurde 1988 von James Dow Allen und Victor Allis unabhängig voneinander mit wissensbasierten Methoden schwach gelöst. Dabei wurde gezeigt, dass der Spieler, der den ersten Zug macht, immer gewinnen kann[2][3].

Inzwischen wurde das Spiel auch durch einen Brute-Force Ansatz stark gelöst. Bei der Lösung durch John Trump kam Alpha-Beta-Pruning zum Einsatz, wobei alle 4.531.985.219.092 legalen Zustände des Spiels untersucht und auf ihre Gewinnchancen bewertet wurden[14].

Lösungen, die alle Möglichkeiten durchrechnen, sind für den Einsatz in der Praxis aufgrund des hohen Rechenaufwands bei komplexeren Anwendungen selten praktikabel. Aus diesem Grund wird bevorzugt auf gute Heuristiken zurückgegriffen, die den Rechenaufwand minimieren, aber dennoch gute Ergebnisse liefern([7], Kapitel 7.6).

Verschiedene allgemeine algorithmische und RL-basierte Ansätze wurden am Beispiel von Vier Gewinnt auf ihre Leistung untersucht. Es wurde gezeigt, dass sich sowohl algorithmische als auch RL-Ansätze bei Vier Gewinnt eignen[1][13][15][12][11][10].

3.2 Symbolische Algorithmen

3.2.1 Minimax

Minimax (auch Minmax) ist ein Algorithmus, der ausgehend von einem Knoten im Spielbaum die darauf folgenden Knoten bewertet und den Knoten mit der besten Bewertung zurückgibt. Bei der Bewertung wird davon ausgegangen, dass der Gegner ebenfalls den Zug wählt, der für sich am günstigsten ist. Das führt dazu, dass wenn die Bewertung anhand der Gewinnchancen erfolgt, auch tatsächlich die Gewinnchancen maximiert werden.

Um die Gewinnchancen zu ermitteln, müssen jedoch alle Knoten des Spielbaums untersucht werden. Die Laufzeit des Algorithmus steigt linear zur Anzahl der zu untersuchenden Knoten und damit bei konstanter Anzahl von Möglichkeiten pro Zug exponentiell zur Suchtiefe. Den gesamten Spielbaum zu durchsuchen, ist daher nur für wenig komplexe Spiele praktikabel. Für komplexere Spiele muss die Suchtiefe begrenzt und auf Heuristiken zurückgegriffen werden, damit die Bewertung in akzeptabler Zeit erfolgen kann([6], Kapitel 4)([7], Kapitel 7.6).

3.2.2 AlphaBeta

Bei AlphaBeta handelt es sich um eine Erweiterung von Minimax.

3.2.3 MCTS

Bei MCTS handelt es sich um eine Erweiterung von AlphaBeta.

3.3 Reinforcement Learning

3.4 Robustheit und Generalisierbarkeit

Verschiedene Reinforcement Learning Ansätze sind auf ihre Robustheit untersucht und es gibt verschiedene Verfahren, um RL-Modelle auf Robustheit zu optimieren[9].

4 Konzept

In diesem Kapitel wird erklärt, wie eine Messumgebung aufgesetzt wurde, um die Eigenschaften der Lösungsansätze Robustheit und Generalisierbarkeit empirisch zu bewerten. In dieser Messumgebung spielen zwei Agents, die die zu untersuchenden Ansätze implementieren, das Spiel wiederholt gegeneinander. Dabei werden deren Gewinnraten und die Spieldauer gemessen.

Die Messungen werden unter verschiedenen Szenarien durchgeführt, in denen die Spielumgebung verschiedene Eigenschaften besitzt. Diese Szenarien enthalten unter anderem gestörte Daten, stochastische Elemente oder veränderte Spielregeln:

- Neutrale Umgebung als Grundlage für die folgenden Messungen.
- Rauschen: Agents erhalten fehlerhafte Informationen über das Spielfeld.
- Stochastik: Unter einer bestimmten Wahrscheinlichkeit landet ein Spielstein nicht in der vorgesehenen Spalte sondern in eine benachbarte Spalte.
- Stochastik: Unter einer bestimmten Wahrscheinlichkeit führt ein Spieler nicht den Zug aus, den er für am besten hält, sondern einen zufälligen Zug.
- Stochastik: Unter einer bestimmten Wahrscheinlichkeit führt ein Spieler mehrere Züge hintereinander durch.
- Generalisierbarkeit: Zum Gewinnen werden nicht vier Spielsteine in einer Reihe benötigt, sondern fünf.

Als Grundlage für die Messumgebung dient das PettingZoo Toolkit. Es abstrahiert Probleme in Umgebungen und stellt eine Schnittstelle für Agents bereit, die mit verschiedenen Lösungsstrategien mit den Umgebungen interagieren. Eine Umgebung, die das Spiel Vier Gewinnt abstrahiert, ist Teil des PettingZoo Toolkits. Es kommen Reinforcement Learning Modelle zum Einsatz, die aus Machine Learning Bibliotheken wie

CleanRL oder Stable-Baselines bereitgestellt werden. Falls vorhanden, wird auf fertig implementierte Algorithmen zurückgegriffen.

5 Realisierung

6 Ergebnisdiskussion

7 Zusammenfassung und Ausblick

8 Literaturverzeichnis

- [1] E. Alderton, E. Wopat, J. Koffman. *Reinforcement Learning for Connect Four*. Techn. Ber. Stanford University, Stanford, California 94305, USA, 2019.
- [2] James Dow Allen. *The complete book of Connect 4: history, strategy, puzzles*. New York, NY : Puzzle Wright Press, 2010.
- [3] Victor Allis. „A Knowledge-Based Approach of Connect-Four“. In: *J. Int. Comput. Games Assoc.* 11 (1988), S. 165. URL: <https://api.semanticscholar.org/CorpusID:24540039>.
- [4] Jörg Bewersdorff. *Glück, Logik und Bluff: Mathematik im Spiel - Methoden, Ergebnisse und Grenzen*. 7. Aufl. Springer Spektrum Wiesbaden, 8. Mai 2018. ISBN: 978-3-658-21764-8. DOI: 10.1007/978-3-658-21765-5.
- [5] Milton Bradley Company. *Connect Four*. <https://www.unco.edu/hewit/pdf/giant-map/connect-4-instructions.pdf>. [Letzer Zugriff am 17. December-2024]. 1990.
- [6] Kevin Ferguson, Max Pumperla. *Deep Learning and the Game of Go*. Manning Publications, January 2019.
- [7] George T. Heineman, Gary Pollice, Stanley Selkow. *Algorithms in a Nutshell*. O'Reilly Media, Inc., October 2008.
- [8] „IEEE Standard Glossary of Software Engineering Terminology“. In: *IEEE Std 610.12-1990* (1990), S. 1–84. DOI: 10.1109/IEEESTD.1990.101064.
- [9] Janosch Moos u. a. „Robust Reinforcement Learning: A Review of Foundations and Recent Advances“. In: *Machine Learning and Knowledge Extraction* 4.1 (2022), S. 276–315. ISSN: 2504-4990. DOI: 10.3390/make4010013. URL: <https://www.mdpi.com/2504-4990/4/1/13>.

-
- [10] Yiran Qiu, Zihong Wang, Duo Xu. „Comparison of Four AI Algorithms in Connect Four“. In: *MEMAT 2022; 2nd International Conference on Mechanical Engineering, Intelligent Manufacturing and Automation Technology*. 2022, S. 1–5.
- [11] Kavita Sheoran u. a. „Solving Connect 4 Using Optimized Minimax and Monte Carlo Tree Search“. In: *Advances and Applications in Mathematical Sciences* 21.6 (2022), S. 3303–3313.
- [12] Henry Taylor, Leonardo Stella. *An Evolutionary Framework for Connect-4 as Test-Bed for Comparison of Advanced Minimax, Q-Learning and MCTS*. 2024. arXiv: 2405.16595 [cs.AI]. URL: <https://arxiv.org/abs/2405.16595>.
- [13] Markus Thill, Patrick Koch, Wolfgang Konen. *Reinforcement Learning with N-tuples on the Game Connect-4*. Techn. Ber. Department of Computer Science, Cologne University of Applied Sciences, 51643 Gummersbach, Germany, 2012.
- [14] John Tromp. *John’s Connect Four Playground*. <https://en.wikipedia.org/w/index.php?title=Wine&oldid=1262619132>. [Letzter Zugriff am 13. Dezember-2024].
- [15] Stephan Wäldchen, Felix Huber, Sebastian Pokutta. *Training Characteristic Functions with Reinforcement Learning: XAI-methods play Connect Four*. 2022. arXiv: 2202.11797 [cs.LG]. URL: <https://arxiv.org/abs/2202.11797>.