Reinforcement Learning für strategische Brettspiele

Prof. Dr. Wolfgang Konen
wolfgang.konen@fh-koeln.de, Tel. 02261/8196-6275

Prof. Dr. Thomas Bartz-Beielstein
thomas.bartz-beielstein@fh-koeln.de, Tel. 02261/8196-6391

Abstract: Reinforcement Learning (bestärkendes Lernen) ist eine wichtige Lernmethode für Anwendungen, in denen eine Belohnung erst zeitverzögert erfolgt, wie es beispielsweise in Brettspielen der Fall ist. Wir zeigen, dass es selbst für einfache Brettspiele sehr stark von den Merkmalen abhängt, ob und wie schnell ein Lernerfolg eintritt. Schlech gewählte Merkmale können den Lernprozess verhindern, geeignet gewählte Merkmale können ihn dagegen um den Faktor 100 beschleunigen.

Zielsetzung

Computerprogramme, die gute Gegenspieler in strategischen Brettspielen sind, gehören schon seit Jahren zum Forschungsgegenstand der Künstlichen Intelligenz, und es wurden hier auch beachtliche Erfolge erzielt.[1] Allerdings sind die Programme in der Regel von menschlichen Experten mit viel Erfahrung im jeweiligen Spiel entwickelt, haben Bibliotheken erfolgreicher Spiele eingebaut oder sie verfolgen mehr oder weniger aufwändige Suchstrategien. Jedes neue Spiel bedeutet dann wieder einen völlig neuen Analyse- und Codierungsaufwand.

Ein generischerer Ansatz bestünde darin, wenn ein Programm durch „try & error“ aus der Beobachtung und Durchführung zahlreicher Spielverläufe selbst lernen kann, was die besten Strategien sind. Wenn ein solcher Ansatz gelingt, so ist er viel besser auf andere strategische Situationen übertragbar. Wir verfolgen in unserem Projekt diesen anderen Ansatz, der die Bedingungen für das Lernen an sich erforscht: Wie können wir es schaffen, dass ein Computer ohne Strategiewissen, allein durch das Spielen gegen sich selbst, zum Teil gemischt mit dem zufälligen Ausprobieren neuer Spielzüge, sukzessive besser lernt, sich in einem solchen für ihn neuen Spiel zu behaupten? Welche Bedingungen müssen erfüllt sein, damit Lernen hier möglich ist?

Reinforcement Learning

Reinforcement Learning (RL, dt.: Berstärkendes Lernen) ist eine mächtige Optimierungsmethode für komplexe Probleme. Es hat besonders dann seine Vorteile, wenn nicht für jede einzelne Aktion eine Belohnung gegeben werden kann, sondern erst später, nach einer Sequenz von Aktionen. Dies ist typischerweise bei Brettspielen der Fall.

Temporal Difference (TD) Learning ist eine Variante des Reinforcement Learning, die durch Sutton und Barto [1] entwickelt wurde und mit Tesauro’s TD-Gammon [2], einem selbstlernenden Computerprogramm, das das Spiel Backgammon auf Weltklasseniveau spielt, große Popularität erlangt hat. Trotz dieses Anfangserfolges stellte sich in späteren Anwendungen das TD Learning oft als schwierig heraus, da es für andere Spiele oder leicht andere Randbedingungen keine guten Ergebnisse erzielte.

---

[1] Beispielsweise mit dem Fritz-Schachprogramm (www.chessbase.de), das sich bereits in Turnieren gegen menschliche Schachweltmeister behauptet hat.

Wir starteten deshalb ein Forschungsprojekt, um die Bedingungen für den Erfolg oder Mißerfolg von TD-Anwendungen genauer zu studieren. Die grundlegenden Algorithmen für selbstlernende TD-Agenten in Brettspielen sind in [3] detailliert beschrieben.

Die Bedeutung von Merkmalen

Zuerst begannen wir mit der Anwendung von TD-Algorithmen auf sehr einfache Spiele wie Nim-3 oder TicTacToe, um die Bedingungen für gutes Lernen zu testen.[2] Genauere Details zu unseren Untersuchungen sind in [4] nachzulesen.

Überraschenderweise stellte es sich heraus, dass es selbst für ein fast triviales Spiel wie Nim-3 von zentraler Bedeutung ist, welche Merkmale dem lernenden TD-Agenten angeboten werden:

- Sieht der TD-Agent Merkmale, die eine hohe Korrelation mit der Gewinnwahrscheinlichkeit haben, so kann er dies sehr schnell lernen, auch wenn er dieses Merkmal aus einer möglichweise großen Anzahl von weniger relevanten oder irrelevanten Merkmalen herausfiltern muss.
- Bekommt der Agent die logisch äquivalente Information in einer Form dargeboten, die zwar im Prinzip genauso gut auf Gewinn oder Verlust schließen lässt, dazu jedoch eine Verknüpfung zweier dargebotener Merkmale erfordert, so fällt das Lernen deutliche schwerer oder ist sogar praktisch unmöglich.

Dies ist deshalb überraschend, weil die Verknüpfung im zweiten Merkmalssatz „eigentlich“ eine einfache Erweiterung darstellt, die durch ein neuronales Netz einfach zu lernen sein sollte. Tatsächlich jedoch stellt sich heraus, dass dieses Konzept aufgrund der im bestärkenden Lernen verzögernd auftretenden Belohnung schwer zu erlernen ist. Die Ergebnisse sind zusammenfassend in Abbildung 1 dargestellt.

[2] Nim-3 ist ein fast triviales Spiel, bei dem M Spielsteine auf dem Tisch liegen und jeder Spieler 1, 2 oder 3 Steine nehmen darf. Ziel ist es, den letzten Stein zu bekommen. Dies gelingt dann, wenn man dem Gegner eine Anzahl von Steinen übrig lässt, die durch 4 teilbar ist.

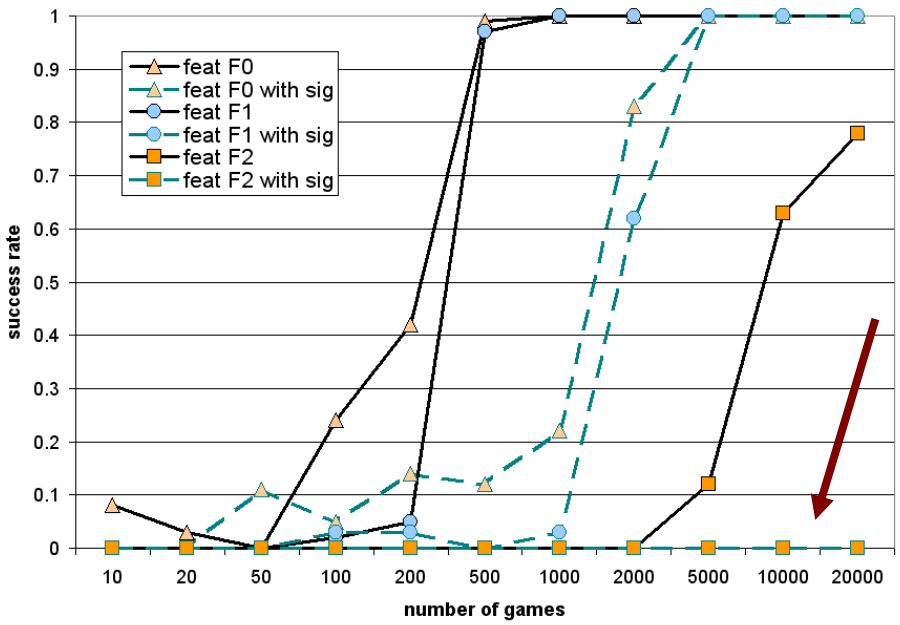

Abbildung 1: Lernerfolg des TD-Agenten im Spiel Nim-3. Die horizontale Achse zeigt die Anzahl der Trainingsspiele in logarithmischer Darstellung, die vertikale Achse die Erfolgsrate eines trainierten TD-Agenten. In Abhängigkeit von den Merkmalen (Feature Set F0 und F1: einfache Kodierung, vs. Feature Set F2: logische Verknüpfung erforderlich) steigt die Erfolgsrate bei F0, F1 entweder schnell an oder sie steigt bei F2 nur sehr langsam oder gar nicht (roter Pfeil) an. [„with sig" bedeutet, dass das Output-Neuron eine Sigmoidfunktion benutzt.]

Ein ähnliches Verhalten zeigt sich bei dem bekannten Spiel TicTacToe (Abbildung 2). Dies gehört auch noch zu den relativ einfachen Spielen, da es nur ca. 5.600 verschiedene Spielzustände gibt. Auch hier lernt unser TD-Agent [4], der zahlreiche Merkmale angeboten bekommt, deutlich schneller (um den Faktor 100) als ein TD-Agent aus der Literatur [5], welcher nur die Spielbrettkonfiguration als Input erhält.

Inzwischen arbeiten wir an dem anspruchsvolleren Brettspiel „Vier gewinnt" (Connect-4), bei dem die Zahl der Spielzustände bei über 10^{14} liegt. Dieses Brettspiel war schon Gegenstand verschiedener Diplomarbeiten an der FH Köln [6][7][8], die das Problem ohne TD-Learning bearbeiteten, sowie einer aktuellen Diplomarbeit [9], die erstmalig einen TD-Agenten einsetzt. Der TD-Agent kann bereits bestimmte Endspiele (Abbildung 3) erlernen, es fehlt allerdings noch der Einsatz von Merkmalen. Das Spielverhalten des TD-Agenten ist daher noch lange nicht optimal, wenngleich er gegen einen zufällig ziehenden Agenten sicher gewinnt.

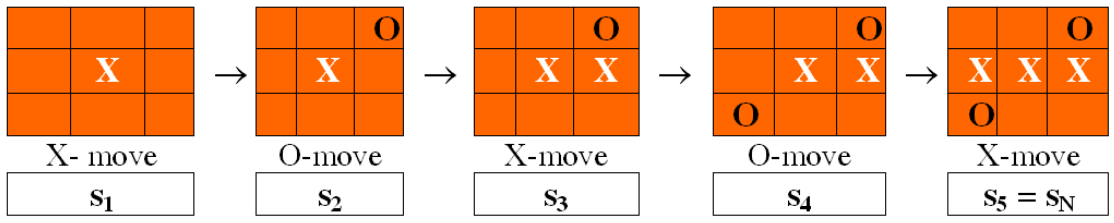

Abbildung 2: Ein beispielhafter Spielverlauf im Spiel TicTacToe, bei dem jeder Spieler versucht, drei Spielsteine in einer Reihe (horizontal, vertikal oder diagonal) zu platzieren.

Ausblick

Eine weitere Ausgestaltung des TD-Learning-Agenten für Connect4 ist geplant. Es sollen insbesondere Merkmale in den TD-Lernvorgang mit eingebracht werden. Hierbei ist die Frage der Evaluation, d.h. welche Merkmale für TD-Learning besonders geeignet sind, von großer Bedeutung.

Für die allgemeine Herangehensweise an strategische Lernsituationen mit einer hohen kombinatorischen Vielfalt von Zuständen ist es interessant, ein generisches Vorgehen zur Gewinnung von Merkmalen zu besitzen. Hierfür erscheinen uns NTupel-Systeme [10] besonders geeignet, deren Nutzen für TD-Learning wir in einem beantragten Forschungsprojekt SOMA (Systematische Optimierung von Modellen in Informatik und Automatisierungstechnik) untersuchen möchten.

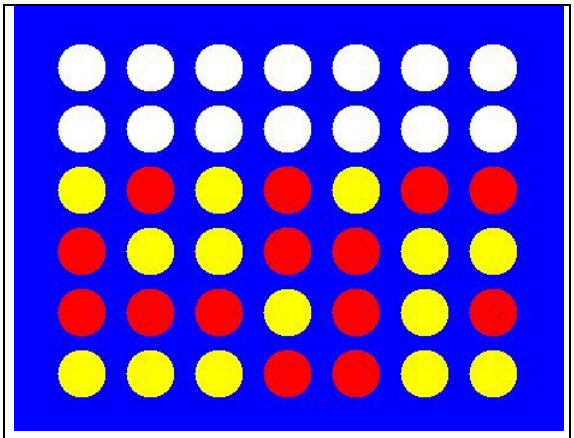

Abbildung 3: Endspielsituation aus dem Spiel Connect-4.

Danksagung: Teile dieser Arbeit wurden durch die FH Köln im Rahmen des anerkannten Forschungsschwerpunktes COSA gefördert.

Literatur

[1] Richard S. Sutton, Andrew G. Barto: Reinforcement Learning - An Introduction. MIT Press, Cambridge, 1998.
[2] Gerald Tesauro: Temporal Difference Learning and TD-Gammon, C*ommunications of the ACM,* March 1995 / Vol. 38, No. 3.
[3] W. Konen: Reinforcement Learning für Brettspiele: Der Temporal Difference Algorithmus, Techn. Report, Institut für Informatik, FH Köln, Okt. 2008. (PDF)
[4] W. Konen, T. Bartz-Beielstein: Reinforcement Learning: Insights from Interesting Failures in Parameter Selection. In: G. Rudolph et al. (ed.), 10th International Conference on Parallel Problem Solving From Nature (PPSN2008), Dortmund, September 2008, p. 478-487, Lecture Notes in Computer Science, LNCS 5199, Springer, Berlin, 2008. (PDF)
[5] M. Stenmark: Synthesizing Board Evaluation Functions for Connect4 using Machine Learning Techniques, Master Thesis, Department of Computer Science, Østfold University College, Norway, July 2005.
[6] T. Wende: *Entwurf und Anwendung künstlicher neuronaler Netze zum Lernen strategischer Brettspiele*, Diplomarbeit, FH Köln, Okt 2003.
[7] T. Rudolph: *Konzeption einer Entwicklungsumgebung lernender KNN für strategische Spiele*, Diplomarbeit, FH Köln, Sept 2003.
[8] A. Klassen: *Evaluation der Einsetzbarkeit lernfähiger neuronaler Netze für das strategische Brettspiel „4-Gewinnt“*, Bachelorarbeit, FH Köln, Feb 2005.
[9] J. Schwenck, *Einsatz von Reinforcement Learning für strategische Brettspiele am Beispiel von "4-Gewinnt“*, Diplomarbeit, FH Köln, Okt 2008.
[10] W. Bledsoe, I. Browning: Pattern recognition and reading by machine. In: Proceedings of the EJCC, pp. 225 232, 1959.