Bericht

über die Diplomarbeit zum Kolloquium

von

Jan Philipp Schwenck

Student der Allgemeinen Informatik

Gummersbach, im Oktober 2008

Betreuende Professoren:

Dr. Wolfgang Konen, Dr. Hartmut Westenberger

1. Thema der Diplomarbeit

Meine Diplomarbeit beschäftigt sich mit dem Thema, ob und wie man strategische Brettspiele mittels einer mathematischen Strategie erlernen kann. Die genutzte Strategie heißt „reinforcement learning“ und nutzt ein bewährtes Verfahren, welches schwerpunktmäßig aus der Pädagogik und Verhaltenspsychologie kommt und unter „Conditional Learning“ bekannt ist.

Das Schlagwort „Pavlow’scher Hund“ ist in diesem Kontext zu nennen, was vielen dadurch ein Begriff ist, da Hr. Pavlow seinerzeit seinem Hund durch Belohnungs- und Bestrafungsmechanismen bestimmte Verhaltensweisen beigebracht hat. Der Hund musste lernen, dass er erst etwas zu fressen erhält, wenn er zuvor das Klingeln einer Glocke hörte. So erlernte der Hund den Zusammenhang zwischen Glocke und Fressen (Belohnung) und reagierte später intuitiv, indem er bei dem Ertönen der Glocke zum Futternapf kam, obwohl nicht unweigerlich eine Belohnung in Form von Fressen dort erschien. Der Hund hatte den Zusammenhang zwischen Zeichen, Reaktion und darauffolgender Belohnung erlernt.

In der Forschung werden bis heute und sicher auch in der Zukunft weitere Experimente zu diesem Thema durchgeführt. Man erkennt recht schnell, das die Verwendung von Belohnungsmaßnahmen ebenso relevant wie alltäglich ist, wie das Nutzen bestimmter Strafen, um schlechte Eigenschaften, Reaktionen oder Verhaltensweisen abzutrainieren.

Reinforcement Learning (im Folgenden mit ‚RL‘ bezeichnet) ist im Grunde nichts anderes, als die Verwendung von Strafe und Belohnung im Kontext einer Anwendung oder eines Problemfalls, innerhalb dessen eine bestimmte Verhaltensweise erlernt werden soll. Im Zusammenhang mit strategischen Brettspielen versucht man RL als Strategie zu nutzen, um Spielsituationen oder –aktionen zu bewerten und daraus gute Entscheidungskriterien für Reaktionen auf eine andere Spielsituation oder –aktion zu erhalten.

2. Inhalte und Arbeitsschwerpunkte

In meiner Diplomarbeit werden konkret zwei Anwendungsfälle unterschieden, die aus unterschiedlichen Zielsetzungen in dieser Arbeit angewandt werden. Der erste (Haupt-)Teil der Diplomarbeit beschäftigt sich mit der unterschiedlichen Umsetzung von RL im Kontext des Spiels Nimm-3. Das kleine und den meisten bekannte Spiel „Nimm-3“ wird genutzt, um daran die Verwendung von RL zu testen – auch im Programmierkontext – und seine Funktionsweise zu erläutern. Hier liegt ein wesentlicher Schwerpunkt der Diplomarbeit, da das grundlegende

Verständnis der genutzten Methode natürlich von enormer Wichtigkeit – auch für eine eventuelle Weiterführung der Experimente – ist.

Nimm-3 ist ein nettes Spielchen, bei dem es darum geht, aus einer vorab definierten Startmenge von Spielsteinen – das können Streichhölzer, Figuren oder Striche auf Papier sein – durch sinnvolles und cleveres Ziehen dieser Figuren möglichst als letzter alle übriggebliebenen Spielsteine zu entfernen. Regel ist dabei, dass nie mehr als drei, und nie weniger als ein Spielstein entfernt werden darf, gezogen wird abwechselnd. Analysiert man das Spiel samt seiner Regel genau, wird einem auffallen, dass es bei der Frage nach dem Sieger des Spiels zum Einen auf die Anzahl der zu Beginn verfügbaren Spielsteine abkommt und zum Anderen davon abhängt wie die Züge der Spieler getätigt werden. Man gewinnt das Spiel immer dann, wenn es einem gelingt nach seinem Zug eine Anzahl an Spielsteinen übrig zu lassen, die durch vier teilbar ist. Warum? Weil der Gegner dadurch nicht die Change erhält (beispielsweise) alle vier Steine zu entfernen (siehe Regel), aber mindestens einen Stein entfernen muss. Tut er dies bleiben aber drei oder weniger Steine übrig, die zum nächsten Teiler von vier oder zum Spielgewinn führen. Somit gewinnt man das Spiel, egal ob der andere Spieler eine gute oder weniger gute Strategie hat. Hierbei fällt letztendlich aber auch auf, dass es neben der Anzahl an Initial-Spielsteinen darauf ankommt, wer der anziehende Spieler ist.

Das Spiel eignet sich gerade weil es so leicht zu erlernen ist gut zum exemplarischen Anwenden der RL-Strategie. Es ermöglicht klare Einteilung der Spielsituationen in gute und schlechte Situationen (sogenannte „States“). Es ist relevant, in Situationen vor oder nach einem Zug zu unterscheiden. Wichtiger sind zweitere (sog. „After States“), weil sie den Raum an Situationsvariationen deutlich geringer halten als erstere. RL dient nun zur Erzeugung aussagekräftiger Bewertungen möglichst aller After States, die in einem Spiel vorkommen. Erreicht man diesen Zustand aus eindeutig bewerteten After States so liegt eine Möglichkeit vor, das Spiel perfekt zu spielen, da ja jede Situation eindeutig charakterisiert und bewertet und entsprechend beim Spielzug agiert werden kann.

Ziel einer jeden Bewertungs- und Charakterisierungsmethode (wie RL) in Anwendung auf strategische Brettspiele sollte es daher immer sein, eine perfekte Spielfunktion zu erzeugen, mit der man das Spiel abbilden und somit spielen und gewinnen kann.

Wie schon zuvor erwähnt, könnte man alternativ zur Bewertung der Spielsituationen auch die Aktionen (also die Züge) in einem Spiel bewerten. Dort unterscheidet man nicht mehr, wie es noch bei den Situationen der Fall war. Man hat allerdings die Schwierigkeit, dass bei Nimm-3 jeder Spielzug immer die gleiche Menge an Variationsmöglichkeiten bietet (mit Ausnahme weniger

spielbeendender Züge), die Situation aber maßgeblich dazu beiträgt, welchen Zug man tätigen sollte. Somit erscheint es sinnvoller, die Situationsbewertende Variante von RL zu untersuchen, gleichwohl auch diese andere Möglichkeit existiert.

Der zweite große Schwerpunkt der Diplomarbeit ist die Untersuchung der RL-Strategie, wenn man sie auf das strategische Brettspiel „4-Gewinnt“ anwendet. Die Hauptfrage, mit der ich mich in dieser Arbeit auseinandergesetzt habe, ist, in wie weit RL das Spiel „4-Gewinnt“ erlernen kann oder ob RL zumindest bei der Erzeugung einer guten Spielfunktion helfen kann.

Das Spiel „4-Gewinnt“ ist dabei allerdings um einige Größenordnungen größer und mächtiger als das zuvor untersuchte Nimm-3. Im Gegensatz zum vorigen Fall, spielt die Positionierung eines einzelnen Spielsteines eine entscheidende Rolle beim Verlauf des Spiels.

„4-Gewinnt“ heißt so, weil es hierbei darauf ankommt, vier gleichfarbige Spielsteine – möglichst seine eigenen – in einer aufeinanderfolgenden, unterbrechungsfreien Reihe anzuordnen. Jeder Spieler darf abwechselnd ziehen, indem er einen seiner Spielsteine in eine Spalte eines sechs Reihen und sieben Spalten großen Spielfeldes positioniert. Dabei ist wichtig zu erwähnen, dass das Spielraster aus 42 (6x7) Spielfeldern aufrecht steht und der Schwerkraft ausgesetzt ist, sodass eingeworfene Spielsteine immer bis auf den untersten noch freien Platz in einer Spalte fallen. Wer zuerst vier eigene Spielsteine anordnen kann, ohne das der Gegner dazwischen wirft und die Reihe unterbricht, gewinnt das Spiel.

Hier ist der Raum aller Spielsituationsvariationen, die entstehen können um eine Vielzahl größer als bei Nimm-3 und hinzu kommt die Tatsache, dass die Position eines Steines einen großen Einfluss auf die Relevanz eines anderen Steines haben kann, was bei Nimm-3 nicht der Fall ist.

Die nun zu untersuchende Frage lautet, ob RL in der Lage ist, genügend spielrelevante Bewertungen vorzunehmen, um eine geeignete Spielfunktion zu trainieren, damit das Spiel „4-Gewinnt“ erlernt werden kann.

Neben den beiden genannten Schwerpunkten der Arbeit existiert natürlich eine ausführliche Einleitung zur Theorie von RL und dem Umgang mit dieser Bewertungmethode, die es ermöglicht die Vorgänge hinter RL zu verstehen und den Ablauf von Bewertungsschritten zu erkennen. Die jeweiligen Arbeitsschwerpunkte (Nimm-3 und 4-Gewinnt) werden durch die Erläuterung des Brettspiels mit sämtlichen Regeln übersichtlich dargestellt.

Die Arbeit wird im Grunde durch das gerade beschriebene Einleitungskapitel und einen Abschlussteil eingerahmt. Dieser Abschlussteil setzt sich aus mehreren Unterkapiteln zusammen, die sich aus der thematischen Zusammenfassung der Ergebnisse, einem Ausblick auf mögliche Weiterführungsschritte und einem persönlichen Fazit zusammensetzen. Hier wird abschließend ein Überblick über die getätigten Arbeiten gegeben, von Erfolgen und Rückschritten, sowie offenen Fragen berichtet und eine persönliche Bewertung der Arbeitsabläufe vorgenommen, die das Interesse des Themas sowie die Verläufe aller Arbeitsschritte während der Diplomarbeit erläutert.

3. Lernerfolg bei Nimm-3 durch RL

Beim Spiel Nimm-3 wird die RL-Strategie durch unterschiedliche Implementierungstechniken getestet. Es gibt eine tabellarische Lösung, die in Tabellenform alle möglichen Spielsituationen auflistet und über RL bewerten lässt. Eine weitere Lösung wird durch die Nutzung eines linearen Netzes dargestellt, was eine Performancesteigerung mit sich bringt, da das Netz durch eine alternative Codierung der Spielsituation, nicht mehr alle denkbaren Situationen auflisten muss. Es minimiert sich somit die Eingabegröße. Die dritte und zuletzt getestete Lösung wird durch ein neuronales Netz beschrieben, welches eine weitere Codierungsalternative nutzt. Dies wird auch ausgeführt, um darzustellen, dass durch die Wahl einer bestimmten Codierung, auch ein bestimmtes mathematisches Modell genutzt werden muss. Dennoch lässt sich durch alle drei Implementierungstechniken das Spiel so lösen, dass man mit der Auswertung der Ergebnisse in der Lage ist, das Spiel perfekt zu spielen.

Bei der tabellarischen Darstellung aller möglichen Spielsituationen erfahren diese durch RL jeweils unterschiedliche Bewertungen, so dass letztendlich eine gute Spielfunktion entsteht, die sich einzig damit beschäftigt, den nächsten Spielzug anhand der Tabelle (und der darin abgelegten Situationsbewertungen) abzulesen. Was hier passiert, ist nichts anderes als eine Bewertung der Spielsituation, die nach einem Zug eines Spielers auftritt. Die RL-Strategie bewertet nur rückwärts, das heißt, es werden zunächst Situationen bewertet, die zu terminierenden Spielzuständen führen. Trainiert man aber mit mehreren Spielen, erfährt die Tabelle auch in früheren Spielsituationen eine Bewertung, die zur Analyse des bestmöglichen Spielzugs betragen kann. Dadurch, dass „vorletzte“ Spielzüge nach einigen Trainingsspielen schon bewertet sind, kann sich eine Bewertung früherer Spielsituationen auf die bereits vorliegenden Bewertungen stützen. Somit wird die Tabelle aller Spielsituationen quasi rückwärts mit Bewertungen gefüllt, anhand derer das Spiel perfekt gespielt werden kann.

Das lineare Netz unterscheidet sich von der tabellarischen Lösung im Wesentlichen dadurch, dass es flexibler auf geänderte Problemstellungen übertragen werden kann, sich aber dabei die grundsätzliche Nutzung des Netzes nicht verändert. Die Übersetzung, die von der tabellarischen Lösung hin zum linearen Netz durchgeführt wird, beinhaltet, dass die möglichen Spielsituationen, die im Spiel vorkommen können, auf den Eingabevektor des linearen Netzes übertragen werden. Allerdings wird dies nur in einfacher Form gemacht, sodass man bei der Codierung des Spiels Nimm-3 soviele Eingabeneuronen hat, wie unterschiedliche Spielsituationen auftreten können, und zwar unabhängig vom Spieler. Der Spieler, der diese Situation hinterlassen hat, wird über die Wertebelegung des jeweiligen Neurons codiert. Hinterlässt beispielsweise Spieler Weiß den After-State „8 übrige Spielsteine“, so wird das Eingabeneuron, welches „8 übrige Spielsteine“ codiert, durch den Wert „1“ aktiviert. Der Wert gibt dabei den Spieler an. Ist der Spieler „Schwarz“ würde das entsprechende Neuron mit „-1“ aktiviert. Alle anderen Eingabeneuronen bleiben mit dem Wert „0“ belegt, damit sie inaktiv sind und keine Auswirkung auf das Ergebnis haben. Denn bei RL kommt es ja einzig auf die Bewertung einer einzelnen Situation an.

Trainiert das lineare Netz genügend Spiele, so erfährt es eine Vielzahl an Situationsbewertungen, die zu einer guten Spielfunktion führen können. Es hängt allerdings hierbei von einer größeren Anzahl von Parametern ab, ob man mittels des linearen Netzes auch eine perfekte Spielfunktion erhält. Hier sind vor allem die Lernschrittweite, der Discount-Faktor, für die Abwertung einer Zukunftssituation und der Zufallswahlfaktor zu nennen, die alle – je nach Wert – eine mehr oder weniger großen Einfluss auf das Ergebnis, und damit auf die Güte der Spielfunktion haben.

Das neuronale Netz wird verwendet, da es je nach Codierungswahl manchmal notwendig ist, ein mathematisch komplexeres Modell zu finden, da nur dieses in der Lage ist, das entstandene Problem zu lösen. Hier entspricht die Codierung einem solchen Problem: Dadurch, dass ergänzend zur Codierung beim linearen Netz ein Eingabeneuron hinzukommt, welches den Spieler repräsentiert, ist es notwendig, eine weitere Neuronenschicht (Hiddenschicht) im Netz zu ergänzen. Das wird notwendig, da nach jeder Spielsituation (After State) die Situation selber – also die hinterlassenden Spielsteine – über ein entsprechendes Neuron codiert werden und der Spieler, der diesen After State hinterlässt, durch ein weiteres Neuron codiert wird. Somit sind nach jeder Spielsituation zwei Neuronen aktiv; dieser Zusammenhang ist aber nicht mehr durch ein lineares Netz erlernbar. Also benötigt man eine erweiterte Form dieser Netzstruktur, was durch ein neuronales Netz erreicht wird.

Der Lernerfolg des neuronalen Netzes wird erst nach wesentlich mehr Trainingsspielen als beim linearen Netz erreicht. Das liegt unter anderem an der Tatsache, dass die Fehlerkorrektur der Kantengewichtungen zu schwach ist, um direkte, spürbare Effekte zu hinterlassen. Erst über eine Vielzahl an Trainingsspielen erreicht man durch die Kantenanpassung eine gute Spielfunktion. Das Problem hierbei besteht durch die zwei-phasige Kantenanpassung. Nach dem ersten Trainingsspiel wird eine Situation (samt des Spielerneurons), die zum Ende des Spiels führt, bewertet. Dabei erreicht die Bewertung – die über die Kantenanpassung geschieht – aber nicht die Kanten der entsprechenden Eingabeneuronen, die letztlich ebenso ausschlaggebend für den Outputwert sind, wie die Kanten zwischen Hidden- und Outputschicht. Im ersten Schritt werden nur letztere Kanten angepasst, was dazu führt, dass mindesten zwei Trainingsspiele durchgeführt werden müssen, um einen Lernerfolg bei den Eingabekanten der beteiligten Neuronen zu erkennen. Somit – und aus der nicht immer besten Wahl der Netzparameter – lässt sich begründen, dass das neuronale Netz viele Tausend Trainingsspiele benötigt, bis der Lernerfolg mit dem des linearen Netzes verglichen werden kann.

Insgesamt kann man aber sagen, dass der Lernvorgang für das Spiel Nimm-3 mit unterschiedlichen Implementierungstechniken funktioniert und – bei einem so einfachen Spiel – auch für die Praxis relevant sein kann. Dennoch entsteht die Frage, ob sich der Lernprozess beschleunigen lässt und die Eindeutigkeit der Ergebnisse auch bei komplexeren Problemstellungen noch gegeben ist.

4. Lernerfolg bei 4-Gewinnt durch RL

Die Schwierigkeit beim Spiel 4-Gewinnt besteht in der Komplexität des Spielgeschehens und der Dimensionsvergrößerung, was sich vor allem auf den Zustandsraum der möglichen Spielsituationen bezieht. Im Gegensatz zu Nimm-3 wird hier kein lineares (eindimensionales) Spielfeld mit jedem Zug verkleinert, sondern ein Zusammenhang zwischen vorherigen und kommenden Spielzügen aufgebaut – das heißt es kommt nicht nur auf die Anzahl übriger Situationsmöglichkeiten an, sondern vielmehr auf die Beziehung eines vorangegangenen Wurfes zum nachfolgenden. Bei 4-Gewinnt existiert im Gegensatz zu Nimm-3 ein zweidimensionales Spielfeld, bei dem es im Verlauf des Spiels besonders auf die Position jedes Spielsteins ankommt. Ziel ist es, vier aufeinanderfolgende Spielsteine einer Farbe zu positionieren, ohne das der Gegner diese Reihe unterbricht und somit verhindert. So ist neben der Spielsituation und dem aktuellen Spieler, im Grunde jede Position aller vorherigen Spielsteine interessant, da jede Position ein gewisses Ausbaupotential besitzt, was zum Gewinn des Spiels beitragen kann. Durch den sechs-

spaltigen und sieben-zeiligen Spielfeldaufbau, erhält das Spiel 42 Spielfelder, die je nach Spielverlauf vollkommen unterschiedliche Belegungsmuster aufweisen. Eine Spielsituation zu codieren erscheint somit wesentlich schwieriger und komplexer als es bei Nimm-3 der Fall ist.

Ein Problem, das während der Testphase auftrat, war die fehlende Zugterminierung. Ein terminierender Zug wurde nicht direkt als beste Zugmöglichkeit erkannt. So fehlte bei den verwendeten Quellen die Angabe, dass eine terminierende Spielsituation immer mit dem Reward (Belohnungswert) zu bewerten ist. Diese Korrektur konnte erst nach einer Vielzahl an Tests gefunden werden, ermöglichte aber nachfolgend bessere und deutlichere Ergebnisanalysen.

Zunächst muss man sich überlegen, ob es überhaupt Sinn macht, eine andere mathematische Methode zu verwenden, als ein neuronales Netz. Dies hängt jedoch sehr von der Codierung der Spielsituation ab, im günstigsten Fall findet man über Analysen und Berechnungen möglicherweise Features, die im Eingabevektor eines Netzes so abgebildet werden können, dass sie ein lineares Netz lösen kann. Aber selbst ein neuronales Netz hat es schwer – so zeigte es die Diplomarbeit – sinnvolle Zusammenhänge zwischen verschiedenen Eingaben zu erlernen, so dass man die Netzausgabe entsprechend einer guten Spielfunktion auswerten kann.

In der vorliegenden Diplomarbeit werden also zwei Codierungsalternativen untersucht, die sich vor allem in der Anzahl der Eingabeneuronen unterscheiden. Die erste Variante codiert jedes Spielfeld mittels dreier Neuronen. Das erste Neuron wird aktiviert – das heißt mit „1“ belegt –, wenn das entsprechende Feld mit einem roten Spielstein belegt ist, das zweite Neuron wird aktiviert, wenn das Feld mit einem gelben Spielstein belegt ist und das dritte Neuron wird aktiviert, wenn das Feld nicht belegt, also noch frei ist. Somit ergeben sich bei 42 Spielfeldern insgesamt 126 Eingabeneuronen, was durchaus einen riesigen Eingabevektor darstellt. Ziel dieser Codierung ist es, mit einheitlicher Wertebelegung der aktivierten Neuronen schnell passende Kantengewichtungen zu erzeugen, die sich positiv auf die Entwicklung einer guten Spielfunktion auswirken.

Die zweite, alternative Codierung stellt jedes Spielfeld durch ein Eingabeneuron dar. Dabei wird die Belegung dieses Feldes durch den angelegten Wert wiedergespiegelt. Dabei repräsentiert ein Neuron, was den Wert „1“ trägt, ein Spielfeld mit einem roten Spielstein, bei einer „-1“ ist das entsprechende Feld mit Gelb belegt. Eine „0“ bedeutet ein freies Feld. Diese Netzstruktur hat den Vorteil, dass nicht noch Zusammenhänge innerhalb der Codierung der Eingabeneuronen erlernt werden müssen, wie es bei der ersten Codierung der Fall ist. Weiterhin wird eine dritte Codierungsalternative angesprochen, die im Rahmen der Arbeit jedoch nur konzeptionell beschrieben werden kann. Dabei geht es um die Idee, das Spielfeld nicht in seiner vollen Belegung

zu codieren, sondern Muster herauszuarbeiten, die die Spielfeldsituation anschaulicher und zusammenfassender beschreiben, bei denen aber dennoch keine relevante Information verloren geht. Der Umfang dieser Überlegungen wird in Diplomarbeiten vorheriger Studierender aufgegriffen, die sich allerdings nicht mit der RL-Strategie auseinandersetzten.

Das Training des Netzes mit diesen zwei beschriebenen und programmierten Codierungsalternativen erweist sich als schwierig. Wie bereits erwähnt, ist es durch die Größe des möglichen Zustandsraums nicht möglich, das Netz auf das gesamte Spiel 4-Gewinnt zu trainieren. Früh zeigt sich, dass es notwendig ist, den Zustandsraum zu verkleinern, damit man überhaupt Lernerfolg erkennen kann. So werden Endspielsituationen erzeugt, die das Spielfeld, zu zwei Dritteln mit Spielsteinen gefüllt, darstellen. Somit wird nicht nur die Anzahl der restlichen, möglichen Spielzüge eingeschränkt, sondern auch die Anzahl der möglichen Spielsituation, die entstehen können – je nach dem, welche Züge getätigt werden. Diese Endspiele schränken allerdings gleichzeitig die Gewinnmöglichkeiten der Spieler ein. Die vorherigen Spielzüge sind als relevante Information zu berücksichtigen, schränken aber gleichzeitig die Kombinationsmöglichkeiten für vier aufeinanderfolgende Spielsteine ein. Das Training dieser Endspielsituationen ergibt nun Ergebnisse, die insgesamt zwar nicht unbedingt erfolgreich sind, dennoch aber mehrere Aussagen zu Trainingsprozess und Netzperformance zulassen.

Bei den durchgeführten Tests wird das zuvor trainierte neuronale Netz gegen zwei verschiedene Gegner auf seine Performance getestet. Man lässt einmal das Netz gegen einen zufallsgesteuerten Gegner spielen, der einzig durch die zufällige Wahl einer (erlaubten) Einwurfspalte eine Zugentscheidung trifft. Der andere Gegner des neuronalen Netzes ist ein sogenannter MinMax-Algorithmus, der das Spiel mittels einer algorithmischen Lösung analysiert und im Voraus den möglichen Spielverlauf berechnet. Aus diesen Informationen errechnet er für sich den besten Zug. In beiden Fällen zeigt das neuronale Netz noch keine perfekte Verhaltensweise. Gegen den zufallsgesteuerten Gegner schafft es allerdings eine Siegquote von knapp 90%. Die Tests gegen den MinMax-Algorithmus verliert das Netz deutlich, wobei große Schwankungen bei den Testergebnissen noch Fragezeichen innerhalb der Analyse dieser Ergebnisse aufzeigen. In manchen Fällen kann es erklärt werden, warum das neuronale Netz deutlich gegen den MinMax-Algorithmus verliert. Die Chance des Gewinnens setzt einfach ein durchtrainiertes Netz voraus, was auch aus eher ungewohnten Spielsituationen noch den Gewinn des Spiels ermöglicht. Teilweise treten aber auch etwas höhere Quoten zugunsten des Netzes auf, diese verändern den Durchschnitt der Testergebnisse zwar nur unwesentlich, sind aber bei der Analyse der Ergebnisse relevant, da sie auf mögliche Stärken des Netzes, aber auch auf gewinnbringende Parametereinstellungen hinweisen, die einen stärken Einfluss auf die

Netzperformance haben, als in anderen Fällen. Insgesamt bleibt aber die Erkenntnis, dass aufgrund der zeitlichen Knappheit, die Testphase wesentlichen zu kurz ist, um in geeigneter Weise festzuhalten, welche Einstellungen dem Netz zu stärkerer Performance verhelfen und wo eventuell noch fehlenden Informationsquellen vorhanden sind, die in der Gestaltung der Spielcodierung, der Netzstruktur oder der Parametereinstellungen berücksichtigt werden müssten.

Es bleibt die Erkenntnis, dass das Netz alleine, keine starke Spielfunktion erzeugen kann, da – selbst bei Endspielen – aus der Berechnung des neuronalen Netzes nicht immer die richtigen Spielzüge getätigt werden.

5. Fazit und Ausblick

Das Ziel, 4-Gewinnt mittels künstlicher Methoden zu Erlernen, kann aus dieser Arbeit heraus so nicht bestätigt werden. Dennoch werden wesentliche Erkenntnisse gewonnen, die deutlich machen, dass sie RL als Lernmethode für strategische Brettspiele unter gewissen Einschränkungen anbietet. Gerade bei der Umsetzung eines lernenden Gegners für das Spiel Nimm-3 wird deutlich, dass es möglich ist, perfekte Spielfunktionen zu generieren – wenn auch die Findung einer perfekten Spielweise dieses Spiels nicht schwer ist. Bei 4-Gewinnt ist die Informationsfülle, die aus einer Spielsituation, aber auch aus einem einzelnen Spielzug gezogen werden kann, wesentlich größer. Somit entsteht die Schwierigkeit, diese Informationsvielfalt angemessen zu codieren, so dass ein künstlicher Gegner entsprechende Reaktionen auf eine Situation auch erlernen kann. Fehlt relevante Information, ist auch die Lernmöglichkeit eingeschränkt. Ebenso kommt es darauf an, die systemeigenen Parameter optimal für einen Anwendungsfall einzustellen. Bei einem Brettspiel kommt es vielleicht mehr auf Details an, die häufig wiederholt werden müssen, bei anderen sind nur Zwischenschritte besonders interessant. So müssen auch die Lernparameter sowie die methodischen Größen korrekt und möglichst optimal gewählt werden.

Fasst man dies alles zusammen, wird deutlich, dass es wesentlich länger dauert und umfassendere Tests erfolgen müssen, um das Erlernen eines strategischen Brettspiels zu ermöglichen. Dennoch wird durch diese Arbeit deutlich, dass es grundsätzlich möglich ist, was an sich ja schon einen Erfolg darstellt. Koppelt man eine lernende Struktur an ein wissendes System, zum Beispiel einen Algorithmus, gewinnt der künstliche Gegner wesentlich an Spielstärke hinzu. Hier bietet sich viele Möglichkeiten der Nutzung an, die ja teilweise auch schon in der Praxis um- und eingesetzt werden.

Persönlich ist die Arbeit eine große Herausforderung gewesen, die mich durch stressige Situationen und kritische Momente führte. Sie ist aber auch eine Motivation, sich schwieriger Themen zu stellen, auf die es vielleicht noch keine eindeutige Antwort gibt. Ich hätte mir neben deutlich mehr Zeit auch ein Team gewünscht, mit dem man Ideen austauschen und einfach den Raum der Gestaltungs- und Umsetzungsmöglichkeiten vergrößern könnte. Ich war an vielen Stellen leider einfach an die bestehenden Ressourcen und die Zeit gebunden, so dass ich lange nicht alles umsetzen konnte, was an Ideen und Möglichkeiten besteht. Dennoch besteht durch diese Arbeit sicherlich ein fortgeschrittener Einstieg in die Thematik von RL und des Erlernen von Brettspielen mittels unterschiedlicher Methoden. In Zukunft darf diese Arbeit gerne Grundlage für weitere Arbeiten sein.