Der Unterschied zwischen Neurogammon und TD-Gammon

Moritz Lintner

Seminar KI: gestern, heute, morgen
Angewandte Informatik, Universität Bamberg

Zusammenfassung. In diesem Paper wird der Unterschied zwischen den beiden Programmen Neurogammon und TD-Gammon geklärt. Bei diesen handelt es sich um zwei künstliche Intelligenzen, welche auf neuronalen Netzen basieren und das Spiel Backgammon zu spielen gemeistert haben. Unterschiede bestehen hier hauptsächlich in der Art und Weise, wie das Spiel erlernt wurde. Um den Unterschied in den beiden Lernverfahren festhalten zu können, wurden die dahinterstehenden Methoden genau angeschaut und festgehalten. Neurogammon basiert hierbei auf der Methode, bei der nach Testdaten gelernt wird und das Programm Expertenspiele analysiert, anhand dessen das Neuronale Netzwerk trainiert wird. Das Programm TD-Gammon basiert hingegen auf einem Lernmechanismus, der Time-Difference-Learning (TD-Learning) heißt und geht hierbei fast komplett ohne Vorwissen an die Aufgabe ran. Es erlernt Strategien, indem es immer wieder im Spiel gegen sich selbst antritt, Fehler erkennt und sich somit immer weiter verbessert.

Schlüsselwörter: TD-Learning, Neurogammon, TD-Gammon, Neuronales Netz

1 Einleitung

Obwohl neuronale Netze anfänglich nicht weit verbreitet waren und recht erfolglos genutzt wurden, finden diese in den letzten Jahren immer häufiger Verwendung, wie beispielsweise bei der Spracherkennung, Bilderverarbeitung, aber auch in künstlichen Intelligenzen zu verschiedenen Spielen, wie Go oder auch Backgammon.
Neuronale Netze können viele verschiedene kognitive Aufgaben übernehmen und ersetzen, sowie ergänzen schon sehr viele herkömmliche Algorithmen, die bisher im Einsatz waren. Bei den meisten von diesen Ansätzen wurden die Probleme versucht durch eine möglichst exakte Beschreibung von Zuständen, Möglichkeiten und Merkmalen zu lösen und Rauschen durch statistische Methoden ab zu fangen. Aufgrund von vielfältigen Variationen, welche nicht allesamt modelliert werden können, sind diese Modelle allerdings sehr fehleranfällig (Trinkwalder, 2016). Dagegen werden neuronalen Netzen nicht auf bestimmte Merkmale oder Strategien abgerichtet, sondern erschaffen durch die automatische Analyse von vielen Datensätzen, verschiedene Muster und Strukturen, wodurch das Netz eine Strategie entwickelt, das Problem auf möglichst

abstrakter Ebene zu lösen. Beispielsweise: Trainiert man ein neuronales Netz mit gesprochener Sprache, wird es mathematische Funktionen bilden, die Phoneme unterscheiden und in Folgen von Phonemen sinnvolle Wörter und Sätze erkennen."(Trinkwalder, 2016).
Hierbei entstanden mehrere Methoden, um dem neuronalen Netz Trainingssätze zur Verfügung zu stellen. Einer davon ist das Trainieren anhand von Testdaten, ein anderer ist das Time-Difference-Learning, welche jeweils als Grundlage hergenommen wurden, um die beiden Programme TD-Gammon und Neurogammon zu entwickeln. Dadurch unterscheiden sich diese beiden Programme, was in nachfolgendem ausgeführt wird. Hierzu wurden verschiedene Artikel und Paper gesucht und die Methoden und Vorgehensweisen direkt miteinander verglichen.

2 Hintergrund

Die beiden Programme TD-Gammon und Neurogammon bauen beide auf neuronalen Netzwerke auf, auch wenn sie unterschiedliche Verfahren verwenden, um das Netz zu trainieren, um bessere Ergebnisse beim Backgammon zu erzielen.

2.1 Backgammon

Backgammon ist ein altes Brettspiel für zwei Personen, welches auf einer Eindimensionalen Strecke mit mehreren Stationen gespielt wird. Gespielt wird mit je 15 weißen und schwarzen Steine, deren Aufstellung zu Beginn fest vorgegeben ist. Jedem Spieler wird eine Farbe zugewiesen, von der Abhängig er die Steine in eine Richtung ans Ende der Strecke bewegen muss. 2.1
Derjenige, der es als erstes geschafft hat, seine Spielsteine an das Ende einer Strecke zu bringen, hat das Spiel gewonnen. Hierzu würfeln die Spieler mit zwei Würfeln und dürfen ihre Steine entsprechend der Augenzahl weiter ziehen. Hierbei kann man die gewürfelten Zahlen mit einem oder zwei verschiedenen Steinen ziehen. Bei einem Pasch darf man für jeden Würfel sogar zweimal gehen.
Durch zusätzliche Regeln, wie das blockieren von Positionen oder das Schmeißen von gegnerischen Steinen, sodass diese am anderen Ende der Strecke wieder ins Spiel gewürfelt werden müssen, wird das Spiel Backgammon um einiges komplexer gestaltet. Zusätzlich gibt es noch einen Verdopplungs-Würfel, mit welchem der Einsatz für das Spiel verdoppelt werden kann.
Es gibt drei verschiedene Arten von Siegen. Zum einen kann man einfach gewinnen, wenn der Gegner auch schon Steine bis ans Ende gebracht hat. Dann kann man Gammon gewinnen, wenn der gegnerische Spieler noch keine Steine in den letzten Feldern der Strecke hat. Und Backgammon ist die höchste Gewinnstufe, wenn der Gegner noch mindestens einen Stein ganz am Anfang der Strecke oder aus dem Spiel geworfen liegen hat.

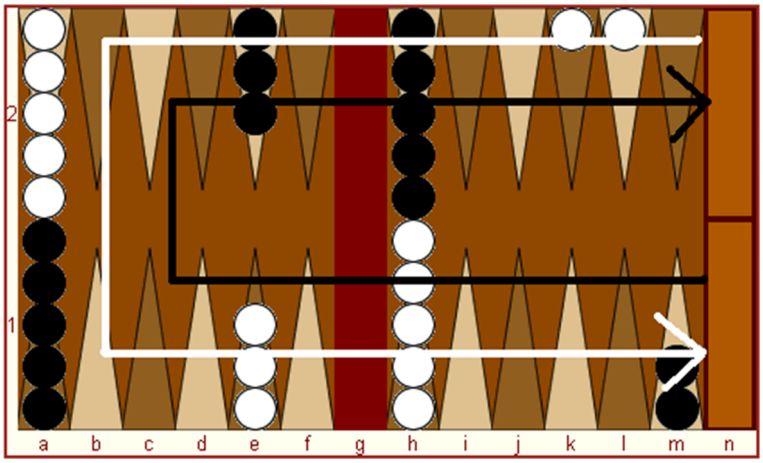


Abb. 1. Backgammon Spielfeld
`https://www.jijbent.nl/spelregels/backgammon.php`

2.2 Neuronales Netzwerk

Definition 1. *Ein neuronales Netzwerk ist eine parallele, verteilte Informationsverarbeitungsstruktur, die aus Verarbeitungselementen besteht (die einen lokalen Speicher verarbeiten können und lokalisierte Informationsverarbeitungsoperationen ausführen können), die miteinander durch unidirektionalen Signalkanälen, die als Verbindungen bezeichnet werden, verbunden sind. Jedes Verarbeitungselement hat eine einzige Ausgangsverbindung, die in beliebig viele zusätzliche Verbindungen verzweigt (fans out") (die jeweils das gleiche Signal - das Verarbeitungselement-Ausgangssignal - tragen). Das Verarbeitungselement-Ausgabesignal kann von irgendeinem gewünschten mathematischen Typ sein. Die gesamte Verarbeitung, die in jedem Verarbeitungselement abläuft, muss vollständig lokal sein, d.h. sie darf nur von den aktuellen Werten der Eingangssignale abhängen, die über auftreffende Verbindungen am Verarbeitungselement ankommen, und von, im lokalen Speicher der Verarbeitungselemente gespeicherten Werten. (Hecht-Nielsen, 1989)*

Ein neuronales Netzwerk besteht also aus einzelnen Verarbeitungselementen, welche Neuronen genannt werden, die Eingangssignale miteinander verrechnen, um ein Ausgangssignal zu erzeugen, welches dann wiederum an die nachfolgenden Elemente versendet wird.
Ein neuronales Netz besteht aus drei verschiedenen Layern. Einmal das Input-Layer, in welchem die Eingangsdaten übergeben werden, dann das Hidden-Layer, welches aus mehreren Schichten von miteinander verbundenen Neuronen bestehen kann und die Informationen verarbeitet, und das Output-Layer, welches

das Ergebnis liefert. Hierbei hat jedes Verarbeitungselement einer Verarbeitungsschicht nur Eingangswerte von vorangegangen Schichten. (vgl. 2.2) (Hecht-Nielsen, 1989)

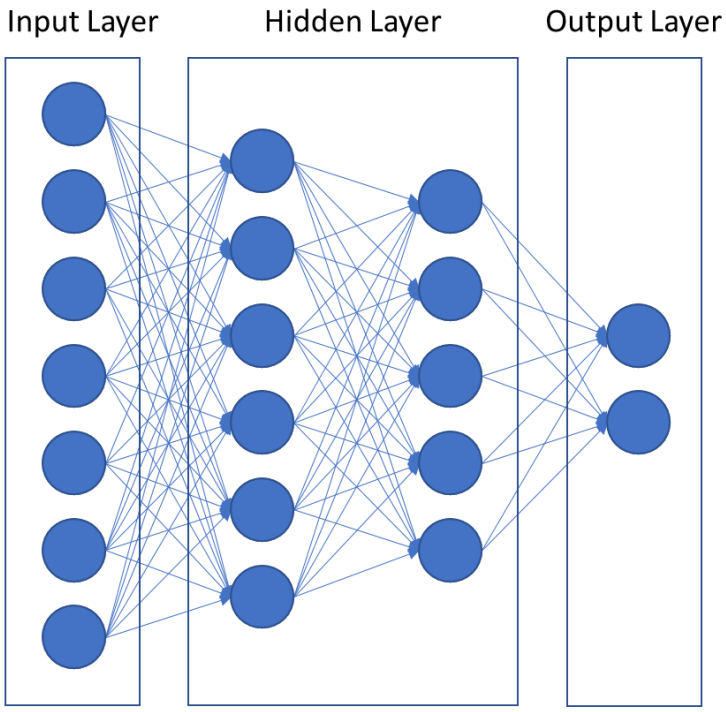


Abb. 2. Neuronales Netz

Beim Training des Netzwerks werden dem Netzwerk immer ein Input gegeben und ein erwarteter Output. Das Netz berechnet nun aufgrund der Eingabewerte ein Ergebnis und gleicht dies dann mit dem Soll-Wert ab. Anschließend wird durch Backpropagation geschaut, wie die Eingangswerte der verschiedenen Neuronen hätten verrechnet werden müssen, um das gewünschte Ergebnis zu erzielen und die Verarbeitungsprozesse jedes Elements werden dem ein wenig angepasst. (Hecht-Nielsen, 1989)

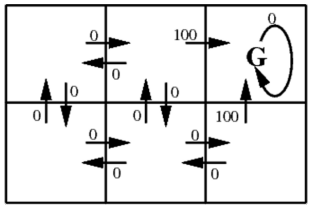
2.3 Q-Learning

Das Q-Learning ist stark mit dem TD-Learning verwandt, weshalb dieses hier auch kurz aufgeführt ist, um das Verständnis des TD-Learning etwas zu vereinfachen.

Wenn ein neuronales Netz ein Spiel erlernen soll, sollte das Netzwerk im besten

Fall bei jeder Aktion, die es ausgewählt hat, direkt ein Feedback bekommt, ob diese Aktion gut oder schlecht war. Bei vielen älteren Spielen, wie beispielsweise manchen Atari-Spiele ist dies noch direkt über den Score möglich. Allerdings ist es bei vielen Spielen so, dass sich der Score nicht verändert, bis man das Ziel erreicht oder andere, längere Aktionen zu Ende gebracht hat. Ähnlich ist es auch bei dem Spiel Backgammon. Hier ist es schwierig, während eines Spieles ohne große Spielerfahrung eine Aktion zu bewerten.

Somit ergibt sich die Ausgangssituation, dass man in mehreren Zuständen s_t aus verschiedenen Aktionen a_t wählen kann, bei denen man keine Belohnung (Reward) r_t erhält, in einem besonderen Zustand dann aber plötzlich eine große Belohnung bekommt (vgl. 5).
$r_t = r(s_t, a_t)$
Beim Q-Learning werden nun den Zuständen, die keine Belohnung liefern, eine erwartete Belohnung $V^\pi(s_t)$ (expected Rewards) hinzugefügt, indem der Reward der benachbarten Aktionen übernommen wird und je weiter dieser entfernt liegt, durch ein γ reduziert einfließt:
$V^\pi(s_t) \equiv r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} \ldots \equiv \sum_{i=0}^{\infty} \gamma^i r_{t+i}$
Somit ergeben sich für jeden Zustand erwartete Belohnungen wie in Abbildung 4. Um nun eine Aktion bewerten zu können, fließt neben der tatsächlich erhaltene Belohnung noch die Erwartete mit ein, sodass folgende Formel daraus resultiert (mit $\delta(s,a)$, die den Zustand s mithilfe der Aktion a in den nächsten Zustand überführt):
$Q[s,a] = r(s,a) + \gamma V^*(\delta(s,a))$
Um für einen aktuellen Zustand s nun die beste Aktion a zu finden, wird diese gewählt, welche das beste $Q[s,a]$ liefert:
$\pi^*(s) = argmax_a(Q[s,a])$
(Ute Schmid, 23.02.2015)



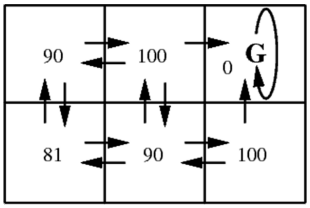


Abb. 4. Erhaltene Belohnungen **Abb. 5.** Erwartete Belohnungen
(Ute Schmid, 23.02.2015)

3 Forschungsmethode

In diesem Kapitel werden die Vorgehensweisen zur Ergebnisfindung zu dieser Forschungsarbeit näher beschrieben.

3.1 Methodenwahl

Ausgehend von bestehenden Forschungsarbeiten zeigte sich oftmals der fehlende direkte Vergleich von TD-Gammon und Neurogammon. Es wurde festgestellt, dass meist nur die Ergebnisse der beiden Algorithmen präsentiert wurden, aber die Fragestellung, warum die Ergebnisse sich unterscheiden offen bleibt. Es zeigte sich, dass die Forschungslücke, durch einen direkten Vergleich der Forschungsbestands, zusammen mit der Analyse der dahinter liegenden Methoden weiter erschlossen werden kann.
Es stellte sich zudem heraus, dass Gerald Tesauro selbst die Arbeiten zu Neurogammon und TD-Gammon schrieb und dass seine Paper aufeinander aufbauten, sodass die Suche nach weiterer Relevanter Literatur nur zu einer sehr begrenzen Referenzmenge führte. Aus diesem Grund musste sich hierbei auf einige wenige Paper über die Programme selbst und deren Lern-Methoden beschränkt werden.

3.2 Analyse relevanter Forschungsarbeiten

Zu Beginn des Forschungsprozesses wurden ausgehend von dem gegebenen Paper von Tesauro (1995) durch Rückwärtssuche und der Suche nach Schlagwörten nach geeigneten Forschungsarbeiten gesucht, die für die Erstellung dieser Forschungsarbeit von Relevanz waren. Der Informationsbestand, der nicht über diese Forschungsarbeiten abgedeckt wurde, wurde noch durch weitere, allgemeine Paper über die einzelnen Methoden noch ergänzt. Hierbei wurde zunächst in digitalen Bibliotheken für wissenschaftliche Arbeiten, Google Scholar und den Katalogen der Universitäten Bamberg und Erlangen nach in der nachfolgenden Tabelle gelisteten Stichworten durchsucht.
Als erstes wurden hierbei die Artikel nach den Titeln und der Verfügbarkeit selektiert, daraufhin wurde anhand des Abstract aussortiert, ob Artikel für die Arbeit wichtig sind und geschaut, welche Fragen von dem Artikel beantwortet werden. Anschließend wurde geschaut, ob von den referenzierten Artikeln noch etwas relevant sein könnte. Als letzte Prüfung, wurden die gesammelten Artikel überprüft, ob noch Fragen offen sind und letztendlich die Finale Bezugsmenge festgelegt.

In der nachfolgenden Tabelle sind ein paar der Suchbegriffe und die dadurch gefundene Literatur, welche schon aufgrund des Titels selektiert wurden, aufgelistet.

Suchmaschine	Suchbegriffe	Gefundene Dokumente
Google Scholar	Neurogammon	- Neurogammon wins computer olympiad - Neurogammon: a neural-network backgammon program
Google Scholar	Neurogammon Tesauro	- TD-gammon: A self-teaching backgammon program - Neurogammon wins computer olympiad - Practical issues in temporal difference learning - Temporal difference learning of backgammon strategy
Google Scholar	Q-Learning	- Q-Learning
Google Scholar	neural network approach game	- Neural Network Design
Google Scholar	backpropagation neural network	- Theory of backpropagation neural network
Bamberger Uni Katalog	TD-Gammon	Nichts
Bamberger Uni Katalog	Neurogammon	Nichts
Bamberger Uni Katalog	Gerald Tesauro	Nichts passendes
Erlangener Uni Katalog	TD-Gammon	- Reinforcement learning : an introduction
Erlangener Uni Katalog	Neurogammon	Nichts
Erlangener Uni Katalog	Gerald Tesauro	Nichts passendes
IEEE Xplore Digital Library	Neurogammon	- Neurogammon wins computer olympiad - Neurogammon: a neural-network backgammon program
IEEE Xplore Digital Library	TD-Gammon	- TD-gammon: A self-teaching backgammon program
IEEE Xplore Digital Library	TD-Learning	- Differential TD learning for value function approximation - TD-learning with exploration
IEEE Xplore Digital Library	Neural Network	Nichts passendes
IEEE Xplore Digital Library	Backpropagation Neural Network	Nichts passendes

Abb. 5. Suchtabelle

Ein Großteil der Literatur war jedoch nicht relevant oder konnte nicht angeschaut oder verwendet werden. Ein paar der Gründe sind nachfolgend aufgeführt:

- Neurogammon wins computer olympiad": Verfügbarkeit
- TD-Gammon: A self-teaching backgammon program": Von „Temporal Difference Learning and TD-Gammon" abgedeckt.
- Practical issues in temporal difference learning": Relevanter Teil wurde von Temporal Difference Learning and TD-Gammon"übernommen. (Gleicher Author)
- Temporal difference learning of backgammon strategy": Von Temporal Difference Learning and TD-Gammonäbgedeckt.
- "Q-Learning": Hier wurde entschieden, die Vorlesung von Frau Schmid zu nehmen.
- Neural Network Design": Theory of backpropagation neural networklieferte bereits das Gesuchte.
- ...

4 Ergebnisse

In dem folgenden Abschnitt werden die Inhalte der Paper im Bezug auf die Funktionsweisen der beiden Systeme zusammengefasst und präsentiert.

4.1 Neurogammon

Das Lernen des Neurogammon basiert auf der einfachen Methode, dass das Netz mehrere Testdaten durchschaut und sich anhand von diesen immer weiter verbessert.
Das Netzwerk bekam die Brettposition als Input und hatte hierbei neun Ausgabemöglichkeiten, welche die Wahrscheinlichkeit für die Bewertung anhand einer 9-Punkte Ratingskala repräsentierte.(Tesauro, 1990)

Training von Neurogammon Tesauro (1990) erklärt, dass das Programm Neurogammon anhand eines Datensets trainiert wurde. Dieses enthielt 3000 Brettpositionen, welche von 64 Expertenspielen genommen wurde. 225 dieser Positionen wurden als Testdaten beiseite gelegt, wohingegen der Rest als Trainingsdaten Verwendung fand.

Evaluation von Neurogammon Um das System leichter evaluieren zu können, wurden die Bewertungspunkte, die das Netzwerk berechnet hatte aufsummiert und ein Unterschied σ zwischen der berechneten und der gegebenen Bewertungen gebildet. Bei einem σ<0,5 galt die Situation als richtig bewertet, wohingegen

bei σ>1,5 die Bewertung signifikant ab wich.
Die besten Ergebnisse wurden hierbei mit einem 243-24-9 Netzwerk erzielt, welches 61% der Brettpositionen mit einem σ<0,5 bewertete und nur 6% mit einem σ>1,5. (Tesauro, 1990)

4.2 TD-Gammon

Das Programm TD-Gammon basiert auf einer Lernmethode, die Time-Difference-Learning (TD-Learning) heißt. Diese wird in dem Unterkapitel Training von TD-Gammon"genauer erklärt.
Bei TD-Gammon bekam das Netzwerk einen Input-Vektor, welcher die Anzahl von weißen und schwarzen Figuren an jeder Spielfeldposition abgespeichert hatte und lieferte als Output einen Vektor mit Vier Feldern, welcher die Aussage liefern sollte, wie hoch die Wahrscheinlichkeit ist, dass Spieler 1 oder Spieler 2 mit einem Gammon oder einem einfachen Sieg gewinnt. Der Sieg durch Backgammon wurde hierbei weg gelassen, da dieser nur in sehr seltenen Fällen auftritt. (Tesauro, 1995)

Training von TD-Gammon TD-Gammon wurde mithilfe des TD-Learning trainiert. Bei diesem Verfahren wird das Spiel gespielt und nach jedem Zug das Netzwerk angepasst. Hierbei wird davon ausgegangen, dass der aktuelle Zustand der wünschenswerte ist und das Netz so angepasst, dass der aktuelle Zustand das nächste mal mit einer höheren Wahrscheinlichkeit erzielt wird.

Die Anpassung der Gewichte wird hierbei mit folgender Formel berechnet.
$w_{t+1} - w_t = \alpha\,(Y_{t+1} - Y_t) \sum_{k=1}^{t} \lambda^{t-k} \nabla_w Y_k$

Y_i: der Output für die Brettposition s_i
w: Vektor der Gewichte des Neuronalen Netzwerkes
α: Eine Konstante, welche beschreibt, wie stark das System sich an den aktuellen Zustand anpassen soll. (Auch Lernrate genannt)
$\nabla_w Y_k$: Ist der Gradient des Outputs des Netzwerkes zu den jeweiligen Zuständen, welcher die aktuellen Gewichte des Netzes mit einbezieht.
λ: Eine Konstante, wie stark die Positionen gewertet werden sollen, die weiter zurückliegen.
Am Ende des Spieles wird das ganze Netzwerk nochmal mit dem finalen Ergebnis z angepasst, wobei hier $z - Y_f$ anstelle von $Y_{t+1} - Y_t$ in der Formel verwendet wird. Somit wird am Ende das Netzwerk nochmal korrigiert, falls es viele schlechte Züge gemacht hat.
(Tesauro, 1995)

Damit sich das Programm nun selbst trainieren kann, musste dieses viele Spiele durchführen. Damit es hierzu keinen menschlichen Gegner braucht, spielt das Programm immer wieder gegen sich selbst und wählt die Züge für beide Seiten.

Da fest gestellt wurde, dass komplett randomisierte Neuronale Netzwerke viel zu lange brauchen, bis ein solides Ergebnis heraus kommt, wurden dem Netzwerk noch einige wenige modellierte Grundkonzepte beigefügt, nach denen das Ergebnis des Netzwerkes noch abgewägt wurde, um schneller Ergebnisse erzielen zu können.(Tesauro, 1995)

Evaluation von TD-Gammon Da man bei TD-Gammon keine Testdaten hatte, musste das Programm beim tatsächlichen Spielen getestet werden. Hierzu gab es verschiedene Versionen, die gegen Profispieler mehrmals hintereinander angetreten sind.
Im Jahr 1991 spielte Version 1.0 in 51 Spielen gegen Robertie, Magriel und Malcolm Davis, welche zu den damals 11 besten Spielern der damaligen zeit gehörten und erzielte ein sehr respektables Ergebnis mit nur 13 Verlustpunkten, welches ca ein Viertel Verlustpunkt pro Spiel ist. Die zweite Version 2.0 des Programms, welches mehr als doppelt so viele Trainingsspiele absolviert hatte, wurde im Jahr 1992 auf der World Cup Backgammon - Meisterschaft vorgestellt. In 38 Vorführungsspielen gegen menschliche Top-Spieler wie Kent Goulding, Kit Woolsey, Wilcox Snellings, die ehemaligen World Cup Champions Joe Sylvester und Joe Russell, hatte das Programm nur 7 Verlustpunkte erzielt. Bei der Version 2.1, welche mit 1,5 Millionen Testspielen die des TDG 2.0 fast verdoppelte, hatte gegen Bill Robertie in 40 Spielen nur eine knappe Niederlage mit 1 Punkt. (Tesauro, 1995)

Program	Trainingsspiele	Gegner	Ergebnis
TDG 1.0	300.000	Robertie, Davis, Magriel	-13 pts/51 games (-0,25 ppg)
TDG 2.0	800.000	Goulding, Woolsey, Snellings, Russel, Sylvester	-7 pts/38 games (-0,18 ppg)
TDG 2.1	1.500.000	Robertie	-1 pts/40 games (-0,02 ppg)

(Tesauro, 1995)

4.3 Direkter Vergleich von TD-Gammon und Neurogammon

TD-Gammon und Neurogammon basieren beide auf neuronalen Netzwerken und haben gelernt das Spiel Backgammon zu meistern. Der Unterschied der beiden Programme liegt aber darin, wie sie das Spiel gelernt haben und wie sie sich verbessert haben.
Neurogammon lernte aufgrund von einigen Trainingsdaten das Spiel, wobei sich die Qualität des Programms auf die der Daten und deren Auswertung beschränkt. Dagegen lernt TD-Gammon anhand von selbst generierten Daten, da dir direkt aus seiner Anwendung resultieren.
Bei dem Programm Neurogammon war es essentiell, dass sich Menschen die Zeit

nehmen, um Testdaten zu erzeugen, welche eine möglichst gute Qualität haben müssen. TD-Gammon hingegen lernt das Programm ohne menschliches Zutun und ist auch vollkommen Unabhängig, ob die Lerndaten erfolgreich erzeugt werden konnten. Es lernt nur durch das Spielen und die Regeln des Spieles selbst, indem es seine Trainingsdaten selbst erzeugt, welche mit der Zeit eine immer höhere Qualität aufweisen.

Der Zweite Punkt, in dem sich die Programme unterscheiden ist die Art der Evaluation. Während bei Neurogammon diese automatisch statt finden kann, nur anhand von Testdaten, braucht es beim TD-Gammon ein anderes gegenüber, welches mehrere Spiele durchführt, um signifikante Ergebnisse zu bekommmen.

Bei dem Neurogammon ist es wichtig, welche Qualität die Daten haben, wohingegen beim TD-Gammon wichtig ist, wie lange dieses trainieren konnte, also wie viele Daten (Quantität) generiert werden konnten.

Für das Spiel Backgammon ist das TD-Gammon wohl der bessere Ansatz, da hier keine Objektivität bei der Bewertung der Spielbrettsituationen eintritt und es sich immer weiter Verbessern kann. Wenn für Neurogammon jedoch noch viele weitere, hochwertige Trainingsdaten erzeugt werden könnten, kann dies jedoch schneller ein noch höheres Niveau erreichen.

5 Diskussion

5.1 Implikationen für bestehende und Zukünftige Ansätze

Der gegenwärtige Forschungsstand zu neuronalen Netzen sowohl im Einsatzbereich der Spiele, als auch im generellen konnte durch diese Forschungsarbeit bezüglich nachfolgender Aspekte erweitert werden.

Es wurde fest gestellt, dass Neuronale Netze im Bereich von Backgammon gute Strategien anhand einiger Datensätze oder des Trainings mit sich selbst liefern können. Dabei beeinflussen bei den Datensätzen die Qualität von diesen das Ergebnis, wobei beim Training mit sich selbst die Quantität viel wichtiger ist, da sich hier die Qualität mit der Zeit erhöht.

Deshalb sollte sich bei der Implementierung von neuronalen Netzwerken anfänglich immer erst die Frage gestellt werden, was genau das Netz liefern soll, da hier schon entschieden werden kann, welcher Ansatz sinnvoller ist. Wenn aufgrund von Regeln und Vorschriften die Generierung eigener Testdaten möglich ist, sollte die TD-Learning-Methode bevorzugt werden, wohingegen, wenn man viele, qualitativ hochwertige Daten hat, das Lernen anhand von diesen sinnvoller ist.

Schlussendlich lässt sich auch sagen, dass neuronale Netze, die mithilfe der TD-Learning-Methode lernen und somit ihre Trainingsdaten selbst generieren

können, ziemlich unabhängig von dem Spiel selbst sind, es müssen nur die Regeln modelliert werden. Wohingegen die Netze, welche mit Datensätzen arbeiten, immer darauf angewiesen sind, dass solche geliefert werden können.
Allerdings muss bei den Selbst-Lernenden Programmen immer eine externe Evaluation stattfinden, wobei bei den Netzen mit Datensätzen eine gewisse Selbst-Evaluierung erfolgen kann.

6 Fazit/Zusammenfassung und Ausblick

Zusammenfassend ist TD-Gammon in der Lage sich eigenständig weiter zu entwickeln. Indem es davon aus geht, dass seine getroffenen Entscheidungen in richtig sind, produziert es, aufgrund von Spielen gegen sich selbst und dem Einhalten von Spielregeln, Trainingsdaten, die es zur Weiterentwicklung nutzt und die mit jedem weiteren Spiel eine immer höhere Qualität erlangen.
Für das Programm Neurogammon hat Tesauro händisch Brettpositionen ausgewertet, um Datensätze zum Lernen zu erzeugen. Ein Teil von diesen Daten wurde als Trainingsdaten verwendet und ein anderer Teil zur Auswertung. Hier konnten verschiedene Netze mit unterschiedlich vielen Schichten durchprobiert werden, um ein möglichst gutes Ergebnis zu erzielen. Jedoch ist Neurogammon immer auf die Qualität der gelieferten Testdaten angewiesen.
Bei hoher Qualität von vorhandenen Datensätzen ist somit der Ansatz des Neurogammon im Vorteil, wohingegen der von TD-Gammon im Vorteil ist, wenn dieser genug Zeit hat, sich selbst immer weiter zu verbessern und es möglich ist, allein aufgrund von Regeln oder Gesetzmäßigkeiten eigene Trainingsdaten zu erzeugen.

In der Zukunft kann eine Kombination aus beiden Systemen ausprobiert werden, indem man zuerst ein neuronales Netz anhand von Trainingsdaten trainiert und dieses dann gegen sich selbst immer weiter trainieren lässt. Somit kann man anhand der Datensätze zuerst schon einen hohes Niveau erreichen und verschiedene neuronale Netze ausprobieren, welche dann durch TD-Learning sich noch weiter verbessert.

Literatur

Hecht-Nielsen. (1989). Theory of the backpropagation neural network. In *International joint conference on neural networks* (p. 593-605 vol.1). IEEE. doi: 10.1109/IJCNN.1989.118638

Tesauro, G. (1990). Neurogammon: A neural-network backgammon program. In *1990 ijcnn international joint conference on neural networks* (p. 33-39 vol.3). IEEE. doi: 10.1109/IJCNN.1990.137821

Tesauro, G. (1995). Temporal difference learning and td-gammon. *Commun. ACM*, *38*(3), 58–68. Retrieved from http://doi.acm.org/10.1145/203330.203343 doi: 10.1145/203330.203343

Trinkwalder, A. (2016). *Netzgespinste: Die mathematik neuronaler netze: einfache mechanismen, komplexe konstruktion.* Retrieved 14.03.2018, from http://heise.de/-3120565

Ute Schmid. (23.02.2015). *Lecture 12: Reinforcement learning: Learning programs and strategies.* Retrieved from http://www.cogsys.wiai.uni-bamberg.de/teaching/ws1718/ml/slides/cogsysII-12.pdf