Untersuchung von stochastischen und nicht stochastischen Reinforcement-Learning-Algorithmen für Blackjack und Kuhn Poker

Bachelorarbeit zur Erlangung des akademischen Grades
Bachelor of Science
im Studiengang IT-Management
an der Fakultät für Informatik und Ingenieurwissenschaften
der Technischen Hochschule Köln

vorgelegt von:	Tobias Felix Marcus
Matrikel-Nr.:	11131684
Adresse:	Wiesenauel 25
	51491 Overath
	tobias_felix.marcus@smail.th-koeln.de

eingereicht bei:	Prof. Dr. rer. nat. Wolfgang Konen
Zweitgutachter*in:	Prof. Dr. Daniel Gaida

Overath, 05.07.2024

Kurzfassung

Das Ziel dieser Arbeit ist es, eine Einschätzung der Spielstärke von komplexeren Reinforcement-Learning-Algorithmen in den Spielen Blackjack und Kuhn Poker zu erstellen und diese mit den Ergebnissen vorangegangener Arbeiten zu diesen Spielen mit einfacheren Algorithmen aus dem General Board Game (GBG) Framework zu vergleichen. Mit Blick auf dieses Ziel werden die folgenden Forschungsfragen gestellt:

- Sind die Agenten in der Lage, die jeweilige optimale Strategie eines stochastischen Spiels zu lernen beziehungsweise sich dieser anzunähern?
    - Wählen sie in bestimmten Situationen die gleichen Aktionen, wie die optimale Strategie?
    - Können sie sich vom Gewinn- oder Verlustwert an die optimale Strategie annähern?
    - Kommen sie vom Gewinn- oder Verlustwert näher an die optimale Strategie als die Agenten des GBG-Frameworks?
- Macht es einen Unterschied, ob die Agenten stochastisch oder deterministisch agieren?

Um diese Fragen zu beantworten zu können, wurden in einem vorangegangenen Praxisprojekt Gymnasium Environments für die beiden Spiele entwickelt. Mit diesen Environments wurden die ausgewählten Stable-Baselines3-Agenten (SB3) trainiert und getestet.

Die Ergebnisse dieser Experimente zeigten klar, dass die Agenten der SB3-Bibliothek in der Lage sind sowohl Kuhn Poker als auch Blackjack zu einem hohen Level zu erlernen. Dabei konnten sie in Kuhn Poker eine perfekte Strategie gegen die optimale Strategie erlernen und sich in Blackjack vom Verlustwert an die Basic Strategy annähern. Aufgrund der Schwankungen in den Vergleichsdaten kann bei Blackjack keine definitive Aussage getroffen werden, ob die getesteten Agenten besser spielen als die Agenten aus dem General Board Game Framework. In Kuhn Poker konnten die SB3-Agenten vor allem als zweiter Spieler klar bessere Ergebnisse erzielen als die GBG-Agenten.

Die Ergebnisse zeigen, dass die komplexeren Agenten der SB3-Bibliothek die Spiele lernen können und im Vergleich zu den einfacheren Agenten aus GBG insgesamt besser abschneiden.

Inhalt

Tabellenverzeichnis

Abbildungsverzeichnis

1 Einleitung

In den letzten Jahrzehnten wurden in vielen Bereichen der künstlichen Intelligenz große Fortschritte erreicht. Im Alltag werden Angebote wie die Textverarbeitungs-KI ChatGPT (OpenAI, 2024) in immer mehr Bereichen genutzt. Auch auf dem Gebiet der Game Theory und des Game Learning werden immer mehr Meilensteine erreicht. So konnte 2017 der von DeepMind entwickelte Algorithmus AlphaGo erstmals einen Großmeister im sehr komplexen Spiel Go schlagen (Silver, Huang, & Maddison, 2016). Bei anderen Spielen gab es ähnliche Fortschritte, ebenfalls im Jahr 2017 konnte ein Team um Micheal Bowling einen Algorithmus entwickeln, der die Zwei-Spieler-Pokervariante Heads-Up Limit Hold'em lösen konnte, (Bowling, Burch, Johanson, & Tammelin, 2015) und eine optimale Strategie ermittelte. Im Jahr 2019 konnte das Programm Pluribus von Noam Brown und Tuomas Sandholm (Brown & Sandholm, 2019) in Runden gegen 5 professionelle Pokerspieler gewinnen.

Im General Board Game (GBG) Framework (Konen, 2019) sind unterschiedliche allgemeine KI-Agenten implementiert, die in den unterschiedlichen Spielen des Frameworks getestet werden können. In diesem Framework wurden von Meißner (Meißner, 2021) und Zeh (Zeh, 2021) jeweils die Spiele Blackjack und Kuhn Poker implementiert und mit den vorhandenen KI-Agenten getestet. Sowohl in Blackjack als auch in Kuhn Poker konnten für die eher einfacheren Agenten des GBG-Framework Lernerfolge verzeichnet werden, wobei die Lernerfolge in beiden Spielen noch Spielraum für Verbesserungen boten. So konnten die Agenten MC-N und MCTSE in Blackjack jeweils in unter 90% der zufälligen Spielsituationen die Aktion der optimalen Strategie treffen. In Kuhn Poker konnten vor allem Qlearn und Sarsa als Startspieler einen durchschnittlichen Gewinn annähernd an die optimale Strategie erreichen, doch als Folgespieler konnte kein Agent einen durchschnittlichen Gewinn von über null erzielen.

Im Vergleich zu diesen Agenten setzen die Reinforcement-Learning (kurz RL) Agenten der Stable-Baselines3 (SB3) (Raffin, et al., 2021) Bibliothek auf komplexere Zielfunktionen, um viele unterschiedliche Aufgaben lösen zu können. Nun sollen die Agenten der SB3 Bibliothek getestet werden, um zu erfahren, ob sie die Spiele Blackjack und Kuhn Poker besser lernen können als die Agenten des GBG-Frameworks. Um diesen Vergleich möglich zu machen, wurden die beiden Spiele in einem vorangegangenen Praxisprojekt (Marcus, 2024) als Gymnasium Environments implementiert.

Die folgenden Forschungsfragen versucht diese Arbeit zu beantworten:

- Sind die Agenten in der Lage, die jeweilige optimale Strategie eines stochastischen Spiels zu lernen beziehungsweise sich dieser anzunähern?
    - Wählen sie in bestimmten Situationen die gleichen Aktionen, wie die optimale Strategie?
    - Können sie sich vom Gewinn- oder Verlustwert an die optimale Strategie annähern?

    - Kommen sie vom Gewinn- oder Verlustwert näher an die optimale Strategie als die Agenten des GBG-Frameworks?

- Macht es einen Unterschied, ob die Agenten stochastisch oder deterministisch agieren?
    - Erwartet wird, dass es bei Blackjack keinen Unterschied macht, da die optimale Strategie mit festen Aktionen für jede Situation festgelegt ist. Bei Kuhn Poker wird jedoch ein Unterschied erwartet, da die optimale Strategie eine stochastische Auswahl an Aktionen vorsieht.
- Können die Agenten gegen einen nicht optimal spielenden Gegner optimal agieren?
    - Erwartet wird, dass die Agenten eine optimale Exploitation Strategie finden können, unsicherer ist, wie viele Runden sie dafür gegen den nicht optimal spielenden Gegner trainieren müssen, um diese zu finden.

Mit Bezug auf die gestellten Forschungsfragen soll diese Arbeit die Möglichkeiten von Reinforcement-Learning-Algorithmen untersuchen. Die Evaluation der Stable-Baselines3-Agenten in den Spielen Blackjack und Kuhn Poker dient als Grundlage, die Stärken dieser und unterschiedlicher anderer Optimierungsansätze zu vergleichen. Die darauf aufbauenden Ergebnisse bieten Einblicke in die Potenziale moderner RL-Algorithmen in der Spieltheorie und darüber hinaus.

Auf Basis der im Praxisprojekt entwickelten Gymnasium Environments werden die SB3-Algorithmen „PPO“, „A2C“ und „DQN“ in Kuhn Poker und „maskablePPO“ in Blackjack getestet. Stable-Baselines3 wurde ausgewählt, da es eine einfache Nutzung erlaubt und für Forschungsprojekte entwickelt wurde. Unter den in SB3 implementierten Algorithmen wurden die oben genannten ausgesucht, da die genutzten Environments bestimmte Formen als Action- und Observationspace nutzen, die von den anderen Algorithmen in SB3 nicht unterstützt werden. Im Folgenden werden die Experimente und Vergleiche dargestellt, die darauf abzielen, die Leistungsfähigkeit dieser Agenten in stochastischen Spielszenarien zu bewerten und zu vergleichen.

Dafür wird in Kapitel 2 ein Überblick über die wichtigste Literatur zum Thema KI mit Blackjack und Kuhn Poker aufgezeigt. Kapitel 3 gibt dann einen Überblick über die Regeln und optimalen Strategien der Spiele. Eine Darstellung der Methoden und Ergebnisse der Evaluation von Blackjack erfolgt in Kapitel 4 und eine solche Darstellung für Kuhn Poker liefert Kapitel 5. Ebenfalls in den Kapiteln 4 und 5 finden sich die jeweiligen Diskussionen zu den Ergebnissen. In Kapitel 6 wird die Arbeit mit einem Fazit und Ausblicken auf eine mögliche Fortführung der Forschung abgeschlossen.

2 Forschungsstand

In diesem Abschnitt werden einige der für diese Arbeit wichtigsten Veröffentlichungen und Standards im Reinforcement Learning und Game Learning vorgestellt.

2.1 Gymnasium

Gymnasium (Farama Foundation, 2023) (ehemals OpenAI Gym (Brockman, et al., 2016)) ist ein Schnittstellenstandard für Reinforcement Learning Umgebungen (Environments) und eine Bibliothek von bestehenden Environments. Diese Environments können genutzt werden, um unterschiedliche RL-Algorithmen zu trainieren und zu testen. Nach dem Standard können auch eigene Environments für die Nutzung mit RL-Algorithmen entwickelt werden. Durch die einheitlichen Schnittstellen lassen sich RL-Algorithmen einfach in unterschiedlichen Aufgaben testen. Dabei wird oft mit Bibliotheken wie Stable-Baselines 3 gearbeitet, die unterschiedliche RL-Algorithmen zur Verfügung stellen.

In der vorangegangenen Projektarbeit (Marcus, 2024) wurden für Blackjack und Kuhn Poker Environments in Gymnasium entwickelt bzw. für Blackjack ein bestehendes Environment um fehlende Funktionen erweitert, die in dieser Arbeit genutzt werden sollen, um die Agenten der SB3 Bibliothek in den Spielen trainieren und testen zu können.

2.2 PettingZoo

PettingZoo (Farama Foundation, 2024) ist, wie auch Gymnasium, ein Schnittstellenstandard für RL. Im Gegensatz zu Gymnasium ist PettingZoo für Environments vorgesehen, mit denen mehrere Agenten gleichzeitig oder abwechselnd interagieren. Dabei können die Environments sowohl kooperativer als auch kompetitiver Natur sein. Ebenfalls wie in Gymnasium gibt es sowohl fertige Environments, die in PettingZoo bereitgestellt werden, als auch die Möglichkeit neue Environments zu entwickeln. Unter den vorhandenen Environments befinden sich mehrere Varianten von Poker jedoch kein Kuhn Poker.

Bevor in der Projektarbeit das Gymnasium Environment für Kuhn Poker entwickelt wurde, wurde versucht ein Environment in PettingZoo zu entwickeln, da dieses für Environments mit mehreren Agenten konzipiert ist. Nach Versuchen mit dem eigenen Kuhn Poker Environment und anderen in PettingZoo bereitgestellten Environments konnte jedoch mit den Agenten der SB3 Bibliothek kein zu erwartender Ansatz eines Lernerfolgs erreicht werden. Für diese Versuche wurde den Anweisungen eines Tutorials der PettingZoo Entwickler gefolgt. In einer Antwort auf ein bereits bestehendes Issue (elliottower, 2024) im PettingZoo GitHub wurde von einem Entwickler erläutert, dass das besagte Tutorial nur ein Proof of Concept sei und nicht auf wesentliche Funktionalität getestet wurde. Seit dem Wechsel zu einem Gymnasium Environment für Kuhn Poker gab es Fortschritte bei der Behebung des Fehlers, ein erneuter Umstieg zu PettingZoo wäre zu diesem Zeitpunkt jedoch nicht mehr umzusetzen. In einem kurzen Test konnte bestätigt werden, dass der Workaround für das Vier-Gewinnt Beispiel funktioniert. Für

das selbst entwickelte Kuhn Poker Environment konnte der Workaround aufgrund anderer technischer Probleme und aufgrund der zeitlichen Begrenzung nicht getestet werden.

2.3 Stable-Baselines3

Stable-Baselines3 (Raffin, et al., 2021) kurz SB3 ist eine Sammlung von Implementationen von Deep-Reinforcement-Learning-Algorithmen in Python. Die Algorithmen werden meist mit Gymnasium Environments genutzt und sind durch das gleichbleibende Interface und die ausführliche Dokumentation einfach einzusetzen.

2.4 (Meißner, 2021)

In dieser Arbeit wurde das Spiel Blackjack mit einigen Algorithmen aus dem GBG-Framework getestet. Dafür wurde Blackjack zuerst im Framework implementiert. Die Algorithmen, die in Blackjack getestet wurden, sind MC und MCTSE, TD-N-Tuple 4, Qlearn-4 und SARSA-4. Die beiden Algorithmen Monte Carlo (MC) und Monte Carlo Tree Search Expectimax (MCTSE) konnten jeweils in über 90% der einfachen Spielsituationen den besten Spielzug nach der Basic Strategy finden. Bei einer Evaluation nach zufälligen Spielsituationen konnten diese beiden Agenten in über 80% der Fälle den von der Basic Strategy vorgegebenen besten Spielzug auswählen. Für den Agenten TD-N-Tuple 4 konnte kein Lernerfolg verzeichnet werden. Durch ein neu entwickeltes „Simple Game“ konnte bewiesen werden, dass TD-N-Tuple 4 aufgrund des Fehlens einer Mittelung der Zielfunktion nicht für stochastische Spiele wie Blackjack geeignet ist. Die Agenten Qlearn-4 und SARSA-4 konnten in einer kurzen Betrachtung ähnlich gute Ergebnisse erzielen wie MC und MCTSE.

2.5 (Vos, 2022)

In dieser Arbeit wurden die RL-Algorithmen Q-Learning und QV-Learning und die evolutionären Algorithmen Genetic Algorithm (GA) und Particle Swarm Optimization (PSO) in Blackjack untersucht. Dabei konnten die RL-Algorithmen die evolutionären Algorithmen in der gemessenen „Winrate“ übertreffen, wobei QV-Learning mit einer ε-greedy Erkundungsstrategie das beste Ergebnis erreichte. PSO lag mit einer „Winrate“ von etwas über 21% weit hinter den anderen Algorithmen, die sich alle in mindestens einer Parameterkonfiguration bis auf weniger als einen Prozentpunkt an das vorgegebene Ergebnis der optimalen Strategie annähern konnten.

2.6 (Zeh, 2021)

Ebenfalls in GBG umgesetzt wurden die Poker Varianten Kuhn Poker und Texas Hold’em. Diese Spiele wurden von Tim Zeh implementiert und in dieser Arbeit mit 8 Algorithmen aus dem Framework getestet. In Kuhn Poker konnten die Agenten TDS, Qlearn und SARSA als Startspieler die besten Ergebnisse erzielen und lagen mit dem durchschnittlichen Verlust nur knapp hinter der optimalen Strategie. Diese drei Agenten konnten als zweiter Spieler jedoch nicht die besten Ergebnisse erreichen. Als zweiter

Spieler schnitten die Agenten MCN und MCTSE am besten ab, konnten sich jedoch nur an einen Verlust von 0 annähern und nicht wie die optimale Strategie einen Gewinn erzielen. Im Durchschnitt über die Ergebnisse als Start- und als zweiter Spieler konnten TDS, SARSA und Qlearn die besten Ergebnisse erreichen, wobei für SARSA und Qlearn auch in Texas Hold'em die besten Ergebnisse gemessen werden konnten.

2.7 (Hoehn, Southey, & Holte, 2009)

Hoehn et al testeten in dieser Publikation Algorithmen und Strategien, die lernen sollten, gegnerische Fehler besser auszunutzen als die optimale Strategie, die von einem Nash-Gleichgewicht ausgeht. Dafür wurden sowohl Parameter Learning als auch Strategy Learning Methoden getestet. Die Agenten auf Basis dieser Methoden sollten dabei in wenigen Spielrunden eine Strategie finden, die gegen einen exploitable Gegner besser spielt als eine pessimistische Nash-Strategie, da zum Beispiel gegen einen menschlichen Spieler ein Spiel mit mehreren Millionen Spielrunden kein realistisches Szenario darstellt. Daher wurde die Anzahl der Runden auf 200 begrenzt. Einer der wichtigsten Parameter war dabei die Runde, in der von einer erkundenden Strategie (Exploration) zu einer ausnutzenden Strategie (Exploitation) gewechselt wird. Die Fehler eines Gegners perfekt auszunutzen ist selbst in einem einfacheren Spiel wie Kuhn Poker nicht möglich, doch es konnte gezeigt werden, dass bereits nach 50 Runden eine Exploitation Strategie gefunden werden kann, die höhere Gewinne erreicht als die optimale Strategie, die von einem Nash-Gleichgewicht ausgeht.

2.8 (Bowling, Burch, Johanson, & Tammelin, 2015)

Durch den Agenten Cepheus, der eine Strategie nutzt, die durch einen CFR+ Algorithmus errechnet wurde, konnte die Poker Variante Heads-Up-Limit-Hold'em (HULHE) gelöst werden. Mit der erreichten Exploitability von 0.986mbb/g (milli-big-blinds pro Spiel) kann das Spiel als schwach gelöst erklärt werden. Auch in einem online Test konnte Cepheus selbst gegen die besten Spieler im Durchschnitt 87 mbb/g gewinnen.

2.9 RLCard

RLCard (DATA Lab, 2019) ist ein Toolkit für Reinforcement Learning in Kartenspielen. Es beinhaltet 10 Spiele die als Environment implementiert sind und die Möglichkeit neue Spiele als Environment zu erstellen. Außerdem beinhaltet es eine Auswahl an Reinforcement Learning und Such-Algorithmen, die in den Spielen getestet werden können. Eines der vorhandenen Environments implementiert das Spiel Blackjack, wobei es eine ähnliche vereinfachte Variante darstellt, wie in Gymnasium. Außerdem enthält RLCard einige unterschiedliche Pokerspiele, jedoch kein Kuhn Poker.

3 Die Spiele

3.1 Blackjack

Blackjack ist ein Kartenspiel mit zufälligen Elementen, bei dem die Spieler auf Basis unvollständiger Informationen agieren müssen. Das Ziel des Spiels besteht darin, mit den Karten in der Hand eine höhere Summe als der Dealer zu erreichen, ohne dabei den Wert von 21 zu überschreiten. Da sowohl die zweite Karte des Dealers als auch mögliche weitere gezogene Karten verdeckt bleiben, müssen die Spieler bei Blackjack mit unvollständigen Informationen arbeiten. Obwohl das Spiel nicht deterministisch ist, existieren optimale Strategien, die in jeder Spielsituation die besten Gewinnchancen bieten. In diesem Abschnitt werden für Blackjack die wichtigsten Regeln und einige mögliche Varianten und die Basic Strategy des Spiels erklärt.

3.1.1 Regeln

Blackjack wird von ein bis sieben Spielern unabhängig voneinander gegen den Dealer gespielt.

3.1.1.1 Ablauf

Zu Beginn jeder Runde setzt jeder Spieler seinen Einsatz, der in den meisten Fällen sowohl eine untere als auch eine obere Begrenzung hat. In der für diese Arbeit genutzten Implementierung wird ein fester Einsatz von 1 simuliert, wobei die Gewinne oder Verluste als Reward ausgegeben werden. Nach den Einsätzen erhält jeder Spieler zwei Karten, die meist offen liegen, und der Dealer erhält eine offene Karte. In der „hole card“-Variante, bei der die zweite Karte des Dealers sofort gezogen und verdeckt abgelegt wird, überprüft der Dealer, wenn seine erste offene Karte ein Blackjack ermöglicht, ob er mit der verdeckten zweiten Karte ein Blackjack hat. Hat der Dealer ein Blackjack, wird die Runde beendet und ausgewertet, ohne dass die Spieler die Möglichkeit haben, mit ihrer Hand zu interagieren. In solchen Fällen können die Spieler, bevor der Dealer die zweite Karte ansieht, eine Insurance-Nebenwette abschließen. Hat der Dealer keinen Blackjack oder wird die „no hole card“-Variante gespielt, können die Spieler nun der Reihe nach ihre Aktionen ausführen. In der genutzten Implementierung kann mit der Option „peek“ die ebenso genannte Überprüfung, ob der Dealer ein Blackjack hat, aktiviert oder deaktiviert werden, wobei die Peek-Option standardmäßig aktiviert ist, wenn keine explizite Angabe erfolgt. Eine Insurance-Nebenwette ist nicht implementiert, könnte aber bei einer Erweiterung des Projekts hinzugefügt werden.

3.1.1.2 Kartenwerte

Die Zahlenkarten haben jeweils den Wert ihrer Zahl.

Bube, Dame und König haben jeweils den Wert zehn.

Ein Ass hat den Wert eins oder elf, zum Vorteil der Person, die es hält.

3.1.1.3 Spielzüge

Jeder Spieler wählt in seinem Zug einen der folgenden 5 Spielzüge.

(1) Hit: Der Spieler bekommt eine weitere Karte.

(2) Stand: Der Spieler nimmt keine weiteren Karten und beendet für sich selbst die Runde.

(3) Double Down: Der Spieler verdoppelt seinen Einsatz und erhält eine weitere Karte. Wie bei Stand ist die Runde für diesen Spieler danach beendet.

(4) Split: Bei zwei gleichen (oder gleichwertigen) Karten in der Starthand kann der Spieler die Karten aufteilen und mit zwei „Händen“ weiterspielen. Dafür muss für die zweite Hand ein weiterer Einsatz gemacht werden. Beide Hände bekommen eine weitere Karte, sodass beide Hände wieder mit einer Starthand weitergespielt werden. Es gibt Regelvarianten, die es verbieten Split mehrmals in einer Runde zu spielen, in der vorliegenden Implementation ist die häufigere Variante implementiert, dass ein Spieler in einer Runde auf zwei „Split“-Aktionen beschränkt ist. Das heißt eine Starthand kann maximal zu drei Händen gesplittet werden.

(5) Surrender: Ist nur als erste Aktion mit der Starthand möglich. Der Spieler gibt die Hand und jegliche Gewinnchancen auf, aber erhält dafür die Hälfte des Einsatzes zurück, im Casinobetrieb würden dafür gegebenenfalls die geringerwertigen Chips genutzt, im vorliegenden Environment werden halbe Punkte verwendet.

3.1.1.4 Gewinnbedingung

Überschreitet ein Spieler mit seiner Hand den Wert 21, verliert er sofort die Runde. Ein Unentschieden tritt ein, wenn der Spieler eine Hand hat, die denselben Wert wie die des Dealers aufweist. Hat der Spieler eine stärkere Hand oder überkauft sich der Dealer (überschreitet den Wert 21), gewinnt der Spieler die Runde.

3.1.1.5 Blackjack

In vielen Varianten, einschließlich der Implementierung im hier genutzten Environment, wird bei einem Blackjack (auch natural Blackjack genannt), also wenn der Wert 21 mit den ersten beiden Karten erreicht wird, der Gewinn im Verhältnis 3:2 ausgezahlt. Das bedeutet, der Spieler erhält seinen Einsatz zurück und zusätzlich einen Gewinn, der dem 1,5-fachen seines Einsatzes entspricht. Hat der Dealer ebenfalls ein Blackjack, gilt die Runde als unentschieden, und der Spieler erhält seinen Einsatz zurück.

3.1.1.6 Auszahlungen

Hat der Spieler eine schwächere Hand als der Dealer oder überkauft sich, verliert er seinen Einsatz. Bei einem Unentschieden erhält er den Einsatz zurück. Hat er eine bessere Hand als der Dealer erhält er seinen Einsatz zurück und einen gleichhohen Gewinn. Gewinnt der Spieler mit einem Blackjack wird meist im Verhältnis 3:2 ausgezahlt.

3.1.1.7 Insurance

Wird die „hole card“-Variante gespielt und die erste Karte des Dealers ist ein Ass, haben die Spieler die Möglichkeit, eine Nebenwette darauf abzuschließen, ob der Dealer ein Blackjack hat. Dies ist die einzige Interaktionsmöglichkeit für die Spieler, wenn der Dealer ein Blackjack hat. Hat der Dealer tatsächlich ein Blackjack wird die Insurance im Verhältnis 2:1 ausgezahlt, also der Spieler erhält den Einsatz auf die Nebenwette zurück und einen Gewinn in der doppelten Höhe des Einsatzes. In der Regel ist es für die Spieler nicht vorteilhaft, die Insurance zu setzen. Bei einer Wahrscheinlichkeit von mehr als einem Drittel, dass die verdeckte Karte eine Zehn ist, wäre es vorteilhaft, die Insurance zu setzen. Solche Situationen lassen sich mit bestimmten Kartenzählungstechniken erkennen, sind aber im genutzten Environment nicht möglich, da die Karten zufällig generiert und nicht aus einem festen Deck gezogen werden und die Wahrscheinlichkeiten daher immer gleichbleiben. Da die Insurance bei einem unendlichen Kartendeck statistisch nie von Vorteil ist, wurde sie im vorliegenden Environment nicht implementiert. Das Environment könnte um eine Insurance erweitert werden, um zu testen, ob und wie schnell die Agenten lernen, diese Nebenwette nicht einzugehen.

3.1.2 Basic Strategy

Die Basic Strategy wurde zuerst von Baldwin et al im Jahr 1956 berechnet (Baldwin, Cantey, Maisel, & McDermott, 1956) und bezeichnet die besten Spielzüge für jeden Spielstand. Die Basic Strategy wird meist als Tabelle dargestellt und kann in den jeweiligen Spielzügen und im durchschnittlichen Gewinn beziehungsweise Verlust variieren, abhängig von den Regelvarianten des Spiels. Für das genutzte Environment wurde ebenfalls im Praxisprojekt ein Agent implementiert, der die Spielzüge der Basic Strategy aus einer Tabelle ausliest und als Benchmark genutzt werden kann.

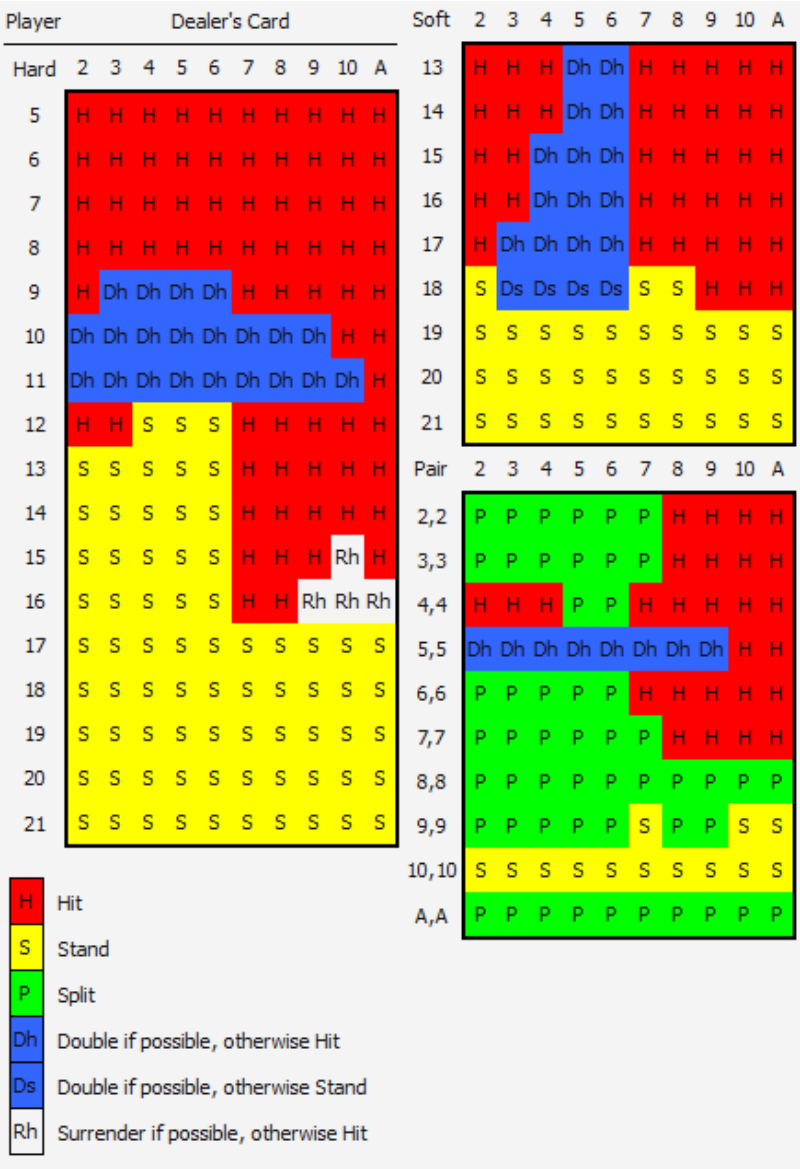


Abbildung 1: Basic Strategy Tabelle

Die abgebildete Strategietabelle wurde von WizardOfOdds (Wizard of Odds, 2010) mit der folgenden Konfiguration agberufen:

- Decks: 4 or more
- Soft 17: Dealer Stands
- Double After Split: Allowed
- Surrender: Allowed with any dealer upcard
- Dealer Peek: Dealer peeks for blackjack

Die Tabellen stellen die optimalen Aktionen für die möglichen Situationen dar. Dabei wird auf der vertikalen Achse der Handwert bzw. die Handkarten des Spielers dargestellt und auf der waagerechten Achse die offene Karte des Dealers. Die Tabelle mit „Soft“ in der oberen linken Ecke stellt die Situationen dar, in denen der Spieler ein Ass in der Hand hält, welches als 11 gewertet wird. Die Tabelle mit „Pair“ stellt die Situationen dar, in

denen die ersten beiden Karten des Spielers den gleichen Wert haben und somit die Aktion „Split“ erlaubt ist. Die Tabelle mit „Hard“ stellt die restlichen Situationen dar, in denen der Spieler kein Ass in der Hand hält oder dieses bereits als eins gewertet wird.

Im vorliegenden Environment konnte mit dem Basic Strategy Agent in 1.000.000 Spielrunden ein durchschnittlicher Reward von ≈-0,7% gemessen werden. Dieser Wert lässt sich nicht exakt in der Literatur finden, was jedoch von dem Problem abstammen kann, dass Blackjack viele Regelvarianten besitzt, die in den Quellen meist unterschiedlich angewandt werden. (Buzzi, 2020) beschreibt einen durchschnittlichen Verlust von 0,834% für einen Basic Strategy Spieler. (Thorpe, 1966) findet in seinen Experimenten einen durchschnittlichen Gewinn von ≈0.1% für einen Basic Strategy Spieler. Dieser Unterschied lässt sich unter anderem dadurch erklären, dass die Experimente mit unterschiedlich vielen Kartendecks gemacht wurden.

3.2 Kuhn Poker

Kuhn Poker (Kuhn, 1950) ist ein einfaches Spiel, das zur Sammlung der Pokerspiele gehört und von zwei Spielern gegeneinander gespielt wird. Es wurde entwickelt, um die wesentlichen Herausforderungen komplexerer Pokervarianten beizubehalten, dabei jedoch die Komplexität durch eine stark reduzierte Anzahl an Karten zu minimieren. Diese Vereinfachung ermöglicht es, die optimalen Strategien für beide Spieler vollständig zu analysieren und zu verstehen. Im folgenden Abschnitt werden die Regeln von Kuhn Poker detailliert erläutert und die optimalen Strategien für die Spieler dargestellt.

3.2.1 Regeln

Das Deck für Kuhn Poker besteht aus drei unterschiedlich wertigen Karten, typischerweise einem Buben, einer Dame und einem König. Die Karten haben die übliche Wertigkeit wie in anderen Pokerspielen: König ist höher als Dame, und Dame ist höher als Bube. Eine Runde beginnt mit einem verpflichtenden Einsatz von beiden Spielern, der meist auf einen Chip festgesetzt ist. Anschließend werden die Karten zufällig verteilt, sodass jeder Spieler eine Karte erhält und die dritte Karte verdeckt bleibt.

Die Spieler können sich abwechselnd für eine der beiden Aktionen „BET“ oder „PASS“ entscheiden, beginnend mit dem Startspieler. Da die Runden meist unabhängig voneinander betrachtet werden, gibt es keine feste Regel, ob immer derselbe Spieler beginnt oder abgewechselt wird. „BET“ entspricht dem „RAISE“ bzw. „CALL“ in anderen Pokervarianten und „PASS“ entspricht dem „CHECK“ bzw. „FOLD“.

Wenn beide Spieler nacheinander die gleiche Aktion wählen oder eine Aktion „PASS“ auf eine Aktion „BET“ folgt, wird die Runde beendet und ausgewertet. Haben beide Spieler die gleiche Aktion gewählt, gewinnt der Spieler mit der höherwertigen Karte einen Chip bei „PASS“ oder zwei Chips bei „BET“ vom anderen Spieler. Wenn ein Spieler „BET“ wählt und der andere mit „PASS“ antwortet, gewinnt der Spieler, der „BET“ gewählt hat, einen Chip vom anderen Spieler.

Abbildung 2: Übersicht der möglichen Spielabläufe in Kuhn Poker (Kuhn, 1950)

3.2.2 Optimale Strategie

Aufgrund dieser Regeln ergeben sich sechs mögliche Aufteilungen der Karten und die fünf unterschiedlichen Aktionsverläufe, die in Abbildung 2 dargestellt sind. Daraus ergeben sich 30 mögliche Spielverläufe. Durch diesen geringen Umfang konnte Kuhn die optimalen Strategien für beide Spieler berechnen.

a. Als Startspieler

Tabelle 1: Optimale Strategie in Kuhn Poker vgl. [Zeh, 2021]

Hand	**Aktion**	**Reaktion**
Bube	α BET (1 – α) PASS	PASS
Dame	PASS	$(\alpha + \frac{1}{3})$ BET $(1 - (\alpha + \frac{1}{3}))$ PASS
König	(3 * α) BET (1 – 3 * α) PASS	BET

b. Als zweiter Spieler

	Aktion Gegner	
Hand	**PASS**	**BET**
Bube	$\frac{1}{3}$ BET $\frac{2}{3}$ PASS	PASS
Dame	PASS	$\frac{1}{3}$ BET $\frac{2}{3}$ PASS
König	BET	BET

In der linken Tabelle in der Spalte Aktion werden die optimalen ersten Aktionen des Startspielers beschrieben und in der Spalte Reaktion die optimalen Aktionen, wenn der Gegner auf ein anfängliches „PASS“ mit „BET“ reagiert hat. In der rechten Tabelle werden die optimalen Aktionen des zweiten Spielers als Reaktion auf die jeweilige erste Aktion des ersten Spielers dargestellt.

Ein zweiter Spieler, der der optimalen Strategie folgt, würde also mit einem Buben auf ein „PASS“ mit einer Wahrscheinlichkeit von $\frac{1}{3}$ mit „BET“ reagieren und mit einer Wahrscheinlichkeit von $\frac{2}{3}$ mit „PASS“ reagieren. Wählt sein Gegner „BET“, würde er immer mit „PASS“ reagieren.

Während sich für den zweiten Spieler exakt eine optimale Strategie ergibt, gibt es für den ersten Spieler unendlich viele optimale Strategien, die von einem Parameter α abhängig sind, wobei $0 \leq \alpha \leq \frac{1}{3}$. Da diese möglichen Strategien für den Startspieler alle Teil des Nash-Gleichgewichts sind, könnte der Startspieler in jedem Spiel ein neues α wählen und denselben Gewinn erwarten. Wenn diese beiden Strategien in einem Spiel aufeinandertreffen, erwartet der Startspieler einen durchschnittlichen Verlust von $-\frac{1}{18}$.

Da es sich um ein Nullsummenspiel handelt, beträgt der erwartete Gewinn für den zweiten Spieler $+\frac{1}{18}$.

Die vorgestellte optimale Strategie stellt ein Nash-Gleichgewicht dar. Das bedeutet, keiner der beiden Spieler kann seine erwarteten Gewinne durch eine andere Strategie erhöhen, solange der Gegner bei der optimalen Strategie bleibt. Da das Ziel beim Poker jedoch nicht nur das Vermeiden von Verlusten, sondern auch die Maximierung der eigenen Gewinne ist, können Strategien, die gegnerische Fehlentscheidungen ausnutzen, effektiver sein. Diese als "exploitative Poker" bezeichneten Strategien nutzen gegnerische Schwächen gezielt aus (888 Poker, 2023). Hierbei werden alle Spielrunden, in anderen Pokervarianten auch solche, in denen man selbst gefoldet hat, genutzt, um die Spielzüge des Gegners zu beobachten. Daraus lassen sich Rückschlüsse ziehen, ob der Gegner zu oft callt oder foldet oder anfällig für aggressive Bluffs ist.

Hoehn et al. (Hoehn, Southey, & Holte, 2009) konnten solche exploitativen Strategien für Kuhn Poker darstellen und Agenten entwickeln, die automatisch passende exploitative Strategien lernen können, wenn der Gegner von der optimalen Strategie abweicht.

4 Blackjack

In diesem Abschnitt werden die Vorgehensweisen und Ergebnisse der Evaluation für Blackjack beschrieben. Da Blackjack eine wesentlich größere Anzahl an möglichen Spielsituationen hat als Kuhn Poker, werden mehr Spielrunden und mehr Zeit benötigt, um einen Agenten zu trainieren und zu testen.

4.1 Evaluationsmethoden

Für das erfolgreiche Lernen und Spielen von Blackjack werden mehrere Kriterien in Betracht gezogen. Ein abstrakteres Kriterium ist die Vergleichbarkeit der so genannten Action Cards. Um diese Action Cards zu messen, müssen die jeweiligen Situationen simuliert und die jeweiligen Aktionen gespeichert werden. Diese Messungen konnten leider aus Gründen der zeitlichen Begrenzung dieser Arbeit nicht durchgeführt werden. Leichter zu messen und zu vergleichen sind dagegen der durchschnittliche Reward und die durchschnittliche Winrate. Um einen besseren Vergleich mit den Ergebnissen von (Meißner, 2021) zu ermöglichen, wird außerdem gemessen, in wie vielen Spielrunden die Aktion des Agenten mit der Vorgabe der Basic Strategy übereinstimmt. Dieses Kriterium wird von Herr Meißner als zufällige Situationen aus der Basic Strategy benannt. Um diese Zahlenwerte zu messen, werden eine Million Spielrunden auf dem Gymnasium Environment gespielt, die Gewinne/Verluste und gewonnene/verlorene Runden aufgenommen und im Durchschnitt dargestellt.

4.2 Action Masking

In Blackjack gibt es viele Spieleraktionen, die nur zu bestimmten Zeitpunkten in einer Runde erlaubt sind. So kann Split nur bei einem Pärchen mit zwei Karten und maximal 2-mal in einer Runde gewählt werden und Double Down und Surrender nur als erste Aktion mit der Starthand. Aufgrund dieser Einschränkungen wurde die neue Methode Action Masking implementiert. Dadurch können dem Algorithmus die jeweiligen Aktionen übergeben werden, die nicht erlaubt sind. Der Einsatz dieser Technik begrenzte die Auswahl der möglichen Algorithmen noch weiter.

4.3 Training und Parametertuning

Aufgrund der zeitlichen Begrenzung dieser Arbeit und der Implementation des Action Masking konnte in Blackjack nur der Algorithmus maskablePPO trainiert und getestet werden. Beim Training des Agenten in Blackjack wurden mit jeder Parameterkombination zwei Millionen Spiele gespielt. Die Parameter, die betrachtet wurden, sind hierbei die Architektur des Policy Netzwerks, die Lernrate, der Discount Faktor „gamma“ und die „clip_range“.

4.4 Evaluation des Agenten

In diesem Abschnitt werden die Ergebnisse des PPO-Agenten dargestellt und diskutiert. Dabei werden die Ergebnisse der Evaluationen mit und ohne deterministische Auswahl der Aktionen miteinander verglichen und mit den Agenten aus GBG verglichen.

4.4.1 Allgemeine Spielstärke von PPO in Blackjack

In Blackjack konnte für den besten PPO-Agenten aus dem Parametertuning im deterministischen Modus ein durchschnittlicher Reward von -0,0119 mit einer Standardabweichung von 0,00089 in fünf Messrunden gemessen werden. Im Vergleich zum gemessenen durchschnittlichen Reward der Basic Strategy von -0,0078 ist der Verlustwert etwa um die Hälfte höher. Verglichen mit dem Ergebnis eines zufällig agierenden Agenten, das bei etwa -0,51 liegt, verliert der PPO-Agent ein Vielfaches weniger.

4.4.2 Vergleich deterministic=True/False

In Abbildung 3 wird erkennbar, dass der Parameter, der einstellt, ob der Agent deterministisch handelt oder nicht, in Blackjack eine große Auswirkung auf die Ergebnisse hat. Dabei verliert der Agent im deterministischen Modus fast ein Drittel weniger an Chips und trifft etwa ein Prozent öfter die optimale Aktion nach der Basic Strategy. Beim Graphen für die Zufälligen Situationen nach Basic Strategy wird auch deutlich, dass dieses Kriterium im Vergleich zur Höhe der Werte und zu den Unterschieden der Varianten wesentlich geringere Schwankungen hat als der durchschnittliche Reward.

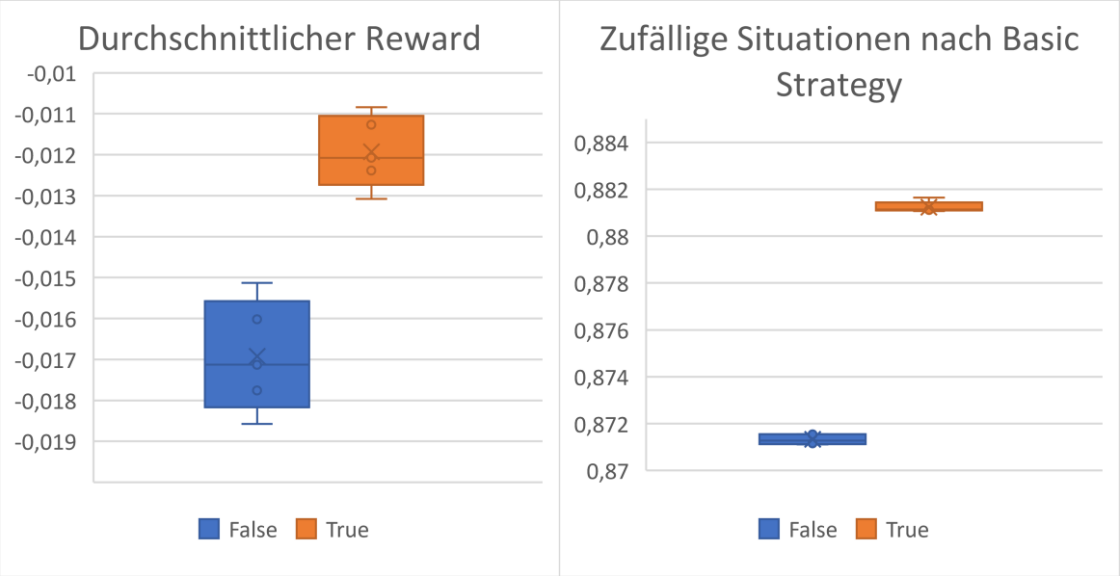


Abbildung 3: Durchschnittlicher Reward und zufällige Situationen nach der Basic Strategy für deterministic=True/False

4.4.3 Vergleich der SB3-Agenten mit den Agenten aus GBG

Der Vergleich mit den GBG-Agenten in Blackjack wird dadurch erschwert, dass die Zahlen in (Meißner, 2021) teilweise durch die lange Laufzeit der Agenten nur mit sehr großer Schwankung erfasst werden konnten.

Tabelle 2: Durchschnittlicher Reward und Zufällige Situationen nach der Basic Strategy PPO im Vergleich zu MC-N und MCTSE

Agent	Durchschnittlicher Reward	Zufällige Situationen der Basic Strategy
Basic Strategy	-0,0078	100%
PPO (SB3)	-0,0119	88,12%
MC-N (GBG)	-0,5 bis 0,6	Höchstwert über 87%
MCTSE (GBG)	-0,6 bis 0,1	Höchstwert über 87%
Random	-0,5118	20,05%

4.4.4 Diskussion

Als durchschnittlicher Reward des PPO-Agenten wurde -0,0119 gemessen. Damit ist der PPO-Agent um ein Vielfaches besser als ein zufällig agierender Agent. Im Vergleich zur Basic Strategy verliert der PPO-Agent etwa 50% mehr.

Die erste Frage, ob der Agent in bestimmten Situationen die gleichen Aktionen wählt, wie die Basic Strategy, kann nicht in Bezug auf einzelne bestimmte Situationen beantwortet werden. Jedoch wählt der PPO-Agent in etwa 88 Prozent der Situationen die gleiche Aktion, wie die Basic Strategy. Dieser Wert spricht dafür, dass der PPO-Agent einen Großteil der Strategie richtig erlernen konnte und ist in etwa vergleichbar mit den Ergebnissen aus Meißners Experimenten (Meißner, 2021).

Zur zweiten Frage, ob der Agent in der Lage ist sich vom Gewinn- oder Verlustwert der Basic Strategie anzunähern, lässt sich feststellen, dass der PPO-Agent sich nicht bis auf die Schwankungsungenauigkeit an die Basic Strategy annähern konnte. Jedoch ist der Abstand zur Basic Strategy sehr gering, im Vergleich zu dem, was zum Beispiel ein zufällig agierender Agent oder ein durchschnittlicher menschlicher Blackjack Spieler erzielt (Thorpe, 1966, S. 33).

Die dritte Frage, ob der Agent näher an die Basic Strategy kommt als die Agenten des GBG-Frameworks, lässt sich nur begrenzt beantworten, da die Ergebnisse der GBG-Agenten aufgrund von hoher Laufzeit eine hohe Ungenauigkeit aufweisen. Anhand der beschriebenen Zahlen lässt sich vermuten, dass der PPO-Agent das Spiel in etwa gleichgut oder geringfügig besser spielt als die GBG-Agenten. Es lässt sich vor allem feststellen, dass der PPO-Agent eine wesentlich kürzere kombinierte Trainings- und Testzeit benötigt, als die beiden Monte Carlo basierten Agenten aus GBG und dennoch ähnlich gute oder bessere Ergebnisse erzielt.

Bei der vierten Frage, ob es einen Unterschied macht, ob der Agent stochastisch oder deterministisch agiert, wurde erwartet, dass es bei Blackjack keinen Einfluss auf das Ergebnis hat. Entgegen dieser Erwartung machen die Ergebnisse deutlich, dass es einen Unterschied macht, ob der Agent stochastisch oder deterministisch handelt. Dabei erzielt der Agent bessere Ergebnisse, wenn er deterministisch agiert. Das lässt sich dadurch erklären, dass der Agent, wenn er deterministisch agiert, immer die gleiche

Aktion wählt, welche öfter die Richtige als die Falsche ist. Agiert er stochastisch, wählt er zu einem gewissen Prozentsatz die Aktion, die er selbst als schlechter einstuft und damit wählt er, auch wenn er die richtige Aktion besser einstuft, teilweise die falsche Aktion.

5 Kuhn Poker

In diesem Abschnitt werden die Methoden und die Ergebnisse der Evaluation für das Spiel Kuhn Poker beschrieben.

5.1 Evaluationsmethoden

Als Kriterium für das erfolgreiche Lernen und Spielen von Kuhn Poker wird der durchschnittliche Gewinn gegen die optimale Strategie gewählt. Dieser wird sowohl als Startspieler als auch als zweiter Spieler gemessen. Dafür werden die Agenten zuerst auf dem Gymnasium Environment trainiert, wobei auch Parametertuning betrieben wird, um die Agenten jeweils bestmöglich auf das Environment einzustellen. Für die Evaluation werden jeweils 200.000 Spiele als Startspieler und 200.000 Spiele als zweiter Spieler gegen die optimale Strategie gespielt. Dabei werden die Gewinne und Verluste gespeichert und im Durchschnitt dargestellt. Diese Evaluation wird zehnmal wiederholt und aus den Ergebnissen der Mittelwert und die Standardabweichung berechnet.

Abgesehen vom durchschnittlichen Reward sollten auch die Action Cards mit der erwarteten besten Strategie verglichen werden. Dieser Vergleich wurde auch für Agenten in (Zeh, 2021) durchgeführt, wodurch ein weiterer Vergleich neben dem durchschnittlichen Reward möglich ist.

5.2 Spiel gegen die optimale Strategie

Beim Spiel gegen die optimale Strategie in Kuhn Poker sind die Aktionen die besten, die der optimalen Strategie entsprechen. Dabei kommt es jedoch nicht darauf an, ob das exakte Verhältnis eingehalten wird, das in der optimalen Strategie beschrieben ist.

Zum Beispiel gibt die optimale Strategie für den Buben als erste Aktion des Startspielers an mit einer Wahrscheinlichkeit von α „BET“ und mit einer Wahrscheinlichkeit von 1-α „CHECK“ zu wählen. „Falls der Spieler mit einem Buben CHECK wählt, ist der Gewinn -1, sollte der Gegner ebenfalls CHECK wählen, so gewinnt der Gegner, sollte der Gegner BET wählen muss der Spieler FOLD wählen, um weitere Verluste zu vermeiden.“ (Zeh, 2021, S. 43) Wählt der Spieler „BET“, würde der Gegner bei einem König immer auch „BET“ wählen und mit einer Dame mit einer Wahrscheinlichkeit von $\frac{1}{3}$ ebenfalls „BET“ wählen und ansonsten „CHECK“. Daraus ergibt sich: (vgl. Zeh, 2021, S. 43/44)

$$Reward(König, BET) = -2$$

$$Reward(Dame, BET) = \frac{1}{3} * (-2) + \frac{2}{3} * 1 = -\frac{2}{3} + \frac{2}{3} = 0$$

$$Gewinn(BET) = \frac{Gewinn(König, BET) + Gewinn(Dame, BET)}{2} = -1$$

Das bedeutet, dass beide Aktionen den gleichen Erwartungswert haben und es somit egal ist, welche der beiden Aktionen vom Agenten gewählt wird. Die Wahrscheinlichkeiten der optimalen Strategie haben erst einen Einfluss, sobald der Gegner seine Strategie

ändern kann. Für die anderen Felder der Tabelle der optimalen Strategie hat Herr Zeh ähnliche Beweise dargestellt, die hier nicht alle wiederholt werden.

Durch die Beweise, die Herr Zeh vorlegt entsteht die in Tabelle 3 dargestellte Strategie, die die Agenten bei perfektem Lernverhalten erreichen sollten. Die linke Tabelle stellt die Aktionen als Startspieler dar, wobei Reaktion für die Aktion in der zweiten Runde steht, falls der Agent als erstes „CHECK“ gewählt und der Gegner mit „BET“ reagiert hat. Die rechte Tabelle stellt die Aktionen als zweiter Spieler dar. Dabei spielt es keine Rolle, ob ein Agent immer die gleiche Aktion oder mit einer gewissen Wahrscheinlichkeit beide Möglichkeiten wählt, solange die gewählten Aktionen im jeweiligen Feld der Strategie dargestellt sind.

Tabelle 3: Strategie gegen die optimale Strategie für Kuhn Poker

Hand	**Aktion**	**Reaktion**
Bube	BET|CHECK	CHECK
Dame	CHECK	BET|CHECK
König	BET|CHECK	BET

	Aktion Gegner	
Hand	**CHECK**	**BET**
Bube	BET|CHECK	CHECK
Dame	CHECK	BET|CHECK
König	BET	BET

5.3 Training und Parametertuning

Für das Training der Agenten in Kuhn Poker werden jeweils 200.000 Spiele als Startspieler und als zweiter Spieler gespielt. Dabei werden die Algorithmen mehrmals mit unterschiedlichen Parameterkonfigurationen trainiert und jeweils getestet, um eine möglichst gute Konfiguration für das Environment zu finden. Die Algorithmen, die getestet werden, sind PPO, A2C und DQN aus der Stable-Baselines3 Bibliothek. Die Parameter, die dabei betrachtet werden, sind die Architektur des Policy Netzwerks, die Lernrate und der Discount Faktor gamma. Bei PPO kommt dazu noch die „clip_range“, die eine Besonderheit bei diesem Algorithmus ist.

Unter den Algorithmen und Parameterkonfigurationen, die für Kuhn Poker getestet wurden, wurden die besten ausgewählt und genauer untersucht. Dabei konnten geringe Unterschiede festgestellt werden, aufgrund derer für jeden Algorithmus eine Konfiguration ausgewählt werden soll, die für die weitere Evaluation genutzt wird.

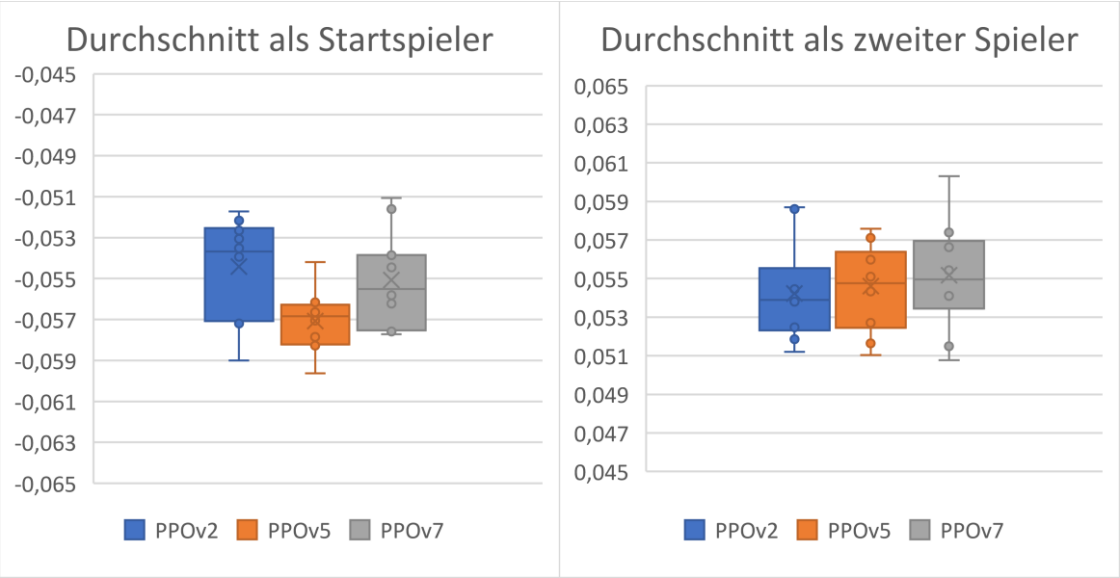


Abbildung 4: Messung der unterschiedlichen Konfigurationen aus dem Parametertuning von PPO

In Abbildung 4 (links) wird deutlich, dass die PPO-Agenten in der Evaluation als Startspieler gewisse Unterschiede in der Leistung aufzeigen. Jedoch sind die Unterschiede in der Leistung nicht signifikant einzustufen im Vergleich zu den Schwankungen der einzelnen Messungen.

In Abbildung 4 (rechts) werden die Ergebnisse der Evaluation als zweiter Spieler dargestellt. Dabei sind sich die unterschiedlichen Parametervarianten noch näher als bei der Evaluation als Startspieler.

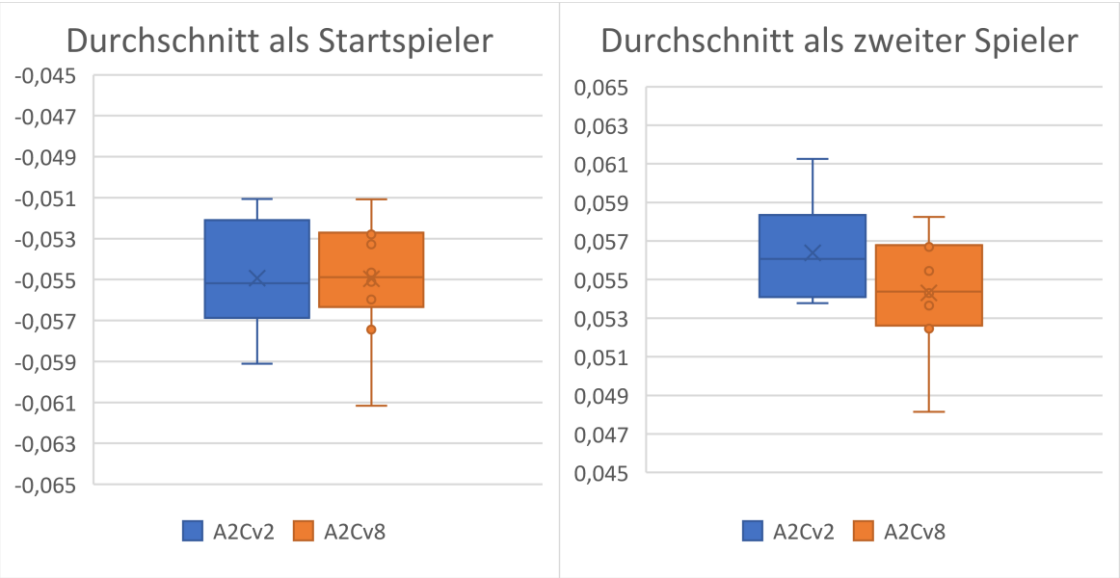


Abbildung 5: Messung der unterschiedlichen Konfigurationen aus dem Parametertuning von A2C

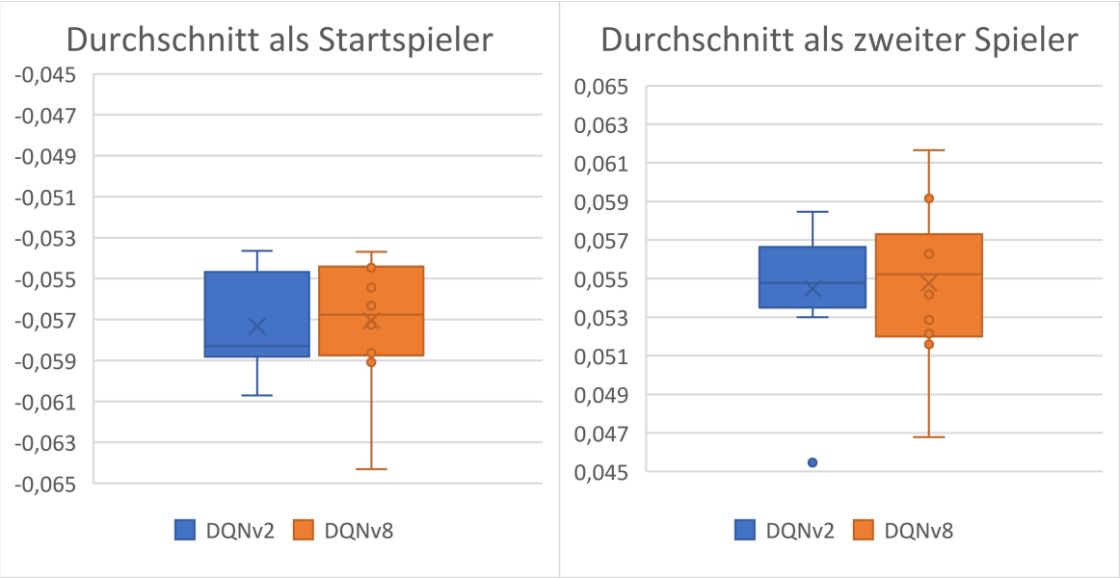


Abbildung 6: Messung der unterschiedlichen Konfigurationen aus dem Parametertuning von DQN

Aus den Abbildungen 5 und 6 geht hervor, dass sich sowohl bei A2C als auch bei DQN ähnliche bzw. noch geringere Unterschiede zwischen den Parameterkonfigurationen zeigen als bei PPO. Dazu sei gesagt, dass bei diesen beiden Algorithmen im Gegensatz zu PPO auch Konfigurationen getestet wurden, die wesentlich schlechtere Ergebnisse lieferten, in der genaueren Betrachtung wurde nur auf die stärkeren Varianten eingegangen, da für den Vergleich mit anderen Algorithmen und Ergebnissen aus anderen Veröffentlichungen möglichst starke Varianten dargestellt werden sollten.

5.4 Evaluation der Agenten

Nachdem die besten Varianten ausgewählt wurden, sollten die Stable-Baselines3-Agenten untereinander und mit den Algorithmen aus GBG verglichen werden. Außerdem sollte untersucht werden, ob die Stable-Baselines3-Agenten unterschiedlich stark sind, je nachdem ob sie deterministisch oder nichtdeterministisch agieren.

5.4.1 Vergleich der Stable-Baselines3 Agenten

Beim Vergleich der SB3-Agenten untereinander liegen die Ergebnisse von PPO und A2C sehr nah zusammen, wobei DQN mit einem kleinen, aber merklichen Abstand dahinter liegt. Die Standardabweichungen der Messungen lagen bei allen drei Algorithmen zwischen 0,002 und 0,003. Im Vergleich dazu sind die Unterschiede zwischen den Algorithmen sehr gering. Ebenfalls liegen die Ergebnisse bei einer nichtdeterministischen Auswahl der Aktionen unter den Ergebnissen, bei denen die Aktionen deterministisch ausgewählt wurden.

Tabelle 4: Durchschnittliche Rewards der SB3 Algorithmen gegen die optimale Strategie

Algorithmus	Deterministisch	Reward als Startspieler	Reward als zweiter Spieler
PPO	Ja	-0,055	0,0552
	Nein	-0,0562	0,0528
A2C	Ja	-0,0549	0,0564
	Nein	-0,0566	0,055385
DQN	Ja	-0,0571	0,0548
	Nein	-0,0659	0,046

5.4.1.1 Strategien der SB3-Agenten

Die folgenden Tabellen stellen die Strategien der SB3-Agenten dar. Dabei stellt die linke Tabelle die Aktionen als Startspieler und die rechte Tabelle die Aktionen als zweiter Spieler dar. Reaktion steht für die zweite Aktion als Startspieler, die nur auftritt, wenn der Agent als erstes „CHECK“ und der Gegner darauf „BET“ wählt. Die Farbe Grün bedeutet, dass in der Situation immer eine der richtigen Aktionen gewählt wurde. Bei den gelb hinterlegten Feldern hat der Agent in den meisten Fällen die richtige Aktion gewählt und würde diese im deterministischen Modus wählen, hat jedoch im nichtdeterministischen Modus zu einer geringen Prozentzahl auch die falsche Aktion gewählt. Bei allen Agenten wird deutlich, dass auch in den Situationen, in denen beide Aktionen den gleichen Reward versprechen, eine Aktion ebenso stark bevorzugt wird, wie in den Situationen, in denen eine Aktion besser sein sollte.

Tabelle 5: Anteil der Aktionen von PPO mit deterministic=False

Hand	Aktion	Reaktion
Bube	>97% CHECK	>99% CHECK
Dame	>98% CHECK	>99% CHECK
König	<1% CHECK	0% CHECK

	Aktion Gegner	
Hand	**CHECK**	**BET**
Bube	>97% CHECK	>99% CHECK
Dame	>98% CHECK	>99% CHECK
König	<1% CHECK	<2% CHECK

Tabelle 6: Anteil der Aktionen von A2C mit deterministic=False

Hand	Aktion	Reaktion
Bube	100% CHECK	100% CHECK
Dame	>99% CHECK	100% CHECK
König	<1% CHECK	-

	Aktion Gegner	
Hand	**CHECK**	**BET**
Bube	>99% CHECK	100% CHECK
Dame	>95% CHECK	100% CHECK
König	0% CHECK	<1% CHECK

Tabelle 7: Anteil der Aktionen von DQN mit deterministic=False

Hand	Aktion	Reaktion
Bube	>97% CHECK	>97% CHECK
Dame	>97% CHECK	>97% CHECK
König	>97% CHECK	<3% CHECK

	Aktion Gegner	
Hand	**CHECK**	**BET**
Bube	>97% CHECK	>97% CHECK
Dame	>97% CHECK	>97% CHECK
König	<3% CHECK	<3% CHECK

5.4.2 Vergleich der Stable-Baselines3 Agenten mit den Agenten aus GBG

Für diesen Vergleich wurden die Ergebnisse der Agenten aus Herr Zehs Arbeit (Zeh, 2021) entnommen, da eigene Experimente mit den GBG-Agenten zeitlich nicht möglich waren. Außerdem wurden einige Agenten aus GBG ausgewählt, da zum Beispiel die Monte Carlo basierten Agenten sehr ähnliche Ergebnisse erzielten und es die Übersichtlichkeit einschränken würde, jeden Agenten aufzuzählen. Bei den SB3-Agenten wurden nur die Ergebnisse mit deterministischer Auswahl der Aktionen gewählt, da diese in jedem Fall besser sind.

Tabelle 8: Übersicht der Ergebnisse der SB3-Agenten und ausgewählter GBG-Agenten

Agent	Reward als Startspieler	Reward als zweiter Spieler
Random vs alpha=$\frac{1}{3}$	-0,1673	-0,1669
PPO (SB3)	-0,055	0,0552
A2C (SB3)	-0,0549	0,0564
DQN (SB3)	-0,0571	0,0548
MCTSE (GBG)	-0,1105	-0,0101
Sarsa-4 (GBG)	-0,0548	-0,0321
Optimal	-0,0552	0,0552

In Tabelle 8 lässt sich erkennen, dass sowohl Sarsa-4 aus dem GBG-Framework als auch die SB3-Agenten mit deterministischer Auswahl der Aktionen sich als Startspieler sehr nah an den perfekten Reward gegen die optimale Strategie annähern konnten. Anhand der Strategietabellen lässt sich erkennen, dass diese Agenten als Startspieler sogar eine nach den Berechnungen im Abschnitt Spiel gegen die optimale Strategie perfekte Strategie finden konnten. Als zweiter Spieler konnten die SB3-Agenten ebenfalls eine perfekte Strategie finden, was den GBG-Agenten jedoch nicht möglich war.

5.4.3 Diskussion

Aus den Strategietabellen wird klar, dass die SB3-Agenten alle in allen Situationen eine richtige Tendenz aufweisen. A2C hat dabei die höchste Anzahl an Situationen, in denen es nur die richtigen Aktionen auswählt. Dafür weicht es mit 4-5% in der Situation als zweiter Spieler mit einer Dame gegen ein „CHECK“ vom Gegner öfter ab als PPO oder DQN. Zieht man zusätzlich die durchschnittlichen Rewards in Betracht, lässt sich

vermuten, dass DQN in dieser Metrik schlechter ist, da es in insgesamt mehr Fällen falsche Aktionen auswählt. A2C und PPO sind sich in der Stärke anhand der Rewards sehr ähnlich, wobei sich die höhere Abweichung in einer Situation und die geringeren Abweichungen in dafür mehr Situationen auszugleichen scheinen. Da alle Felder in den Strategien entweder gelb oder grün sind, lässt sich der leichte Verlust der Ergebnisse bei einer nichtdeterministischen Auswahl der Aktionen im Vergleich zu den anderen Ergebnissen dadurch erklären, dass die Agenten, wenn sie nichtdeterministisch handeln, zu unter 5% immer noch die schlechtere Aktion wählen, handeln sie deterministisch, wählen sie immer die gleiche und in diesem Fall die bessere Aktion.

Die Frage, ob die Agenten in bestimmten Situationen die gleichen Aktionen wie die optimale Strategie wählen, kann somit bejaht werden. Die Agenten geben zwar nicht in allen Situationen die gleichen Wahrscheinlichkeiten für die Aktionen an wie die optimale Strategie, wählen jedoch im deterministischen Modus immer die oder eine der Aktionen, die auch von der optimalen Strategie gewählt würde.

Die Frage, ob die Agenten sich vom Gewinn oder Verlustwert an die optimale Strategie annähern können, kann ebenfalls bejaht werden, da die gemessenen Ergebnisse der Agenten unter Betrachtung der Messungsschwankungen gleichwertig sind, wie die der optimalen Strategie. Außerdem treffen die Agenten im deterministischen Modus eine perfekte Strategie nach den Berechnungen im Abschnitt Spiel gegen die optimale Strategie und solch eine Strategie erreicht bei ausreichender Rundenzahl immer den gleichen Wert der optimalen Strategie. Damit sind die SB3-Agenten auch näher am Gewinn-/Verlustwert als die Agenten aus GBG, von denen keiner als zweiter Spieler einen positiven Reward erreichen konnte.

Die Frage, ob es einen Unterschied macht, ob die Agenten stochastisch oder deterministisch agieren, kann ebenfalls bejaht werden. Jedoch ist das Ergebnis dieser Untersuchung umgekehrt zur Erwartung. Anstatt einen Vorteil durch das stochastische Auswählen der Aktionen zu haben, wählen die Agenten öfter in den falschen Situationen die falschen Aktionen. In der Untersuchung Spiel gegen die optimale Strategie wurde bewiesen, dass es in den Situationen, in denen beide Aktionen in der optimalen Strategie abgebildet sind, keinen Unterschied macht mit welcher Wahrscheinlichkeit man welche Aktion auswählt. Das bedeutet, dass die deterministische Auswahl der Aktionen gegenüber der stochastischen Auswahl einen Vorteil bringt, da die Agenten sich dabei in jeder Situation für eine Aktion entscheiden und somit in den wichtigen Situationen nur die richtige Aktion wählen.

Ein nach einer nicht optimalen Strategie handelnder Agent konnte in der gegebenen Zeit leider nicht umgesetzt werden, weshalb die Frage, ob die Agenten gegen einen solchen Agenten eine optimale Exploitation Strategie finden können, nicht beantwortet werden kann.

6 Fazit und Ausblick

Im Vorfeld dieser Arbeit wurden im Rahmen eines Praxisprojekts (Marcus, 2024) zwei Environments nach der Gymnasium API implementiert, die die Spiele Blackjack und Kuhn Poker darstellen. Die Spiele sind beide als rundenbasierte Spiele mit imperfekten Informationen und nichtdeterministischen Ereignissen einzustufen. Spiele mit dieser Einstufung waren in der Vergangenheit für Reinforcement-Learning-Agenten eine große Herausforderung. In dieser Arbeit sollte daher überprüft werden, ob moderne, komplexe Reinforcement-Learning-Algorithmen diese Spiele besser lernen können.

Die vorliegende Arbeit hat gezeigt, dass moderne Reinforcement-Learning-Algorithmen in verschiedenen Spielkontexten eingesetzt werden können, um optimale oder annähernd optimale Strategien zu entwickeln. Dabei wurde deutlich, dass es in den beiden untersuchten Spielen einen Vorteil bietet, die Aktionen nicht stochastisch auszuwählen, da die Algorithmen die bestmöglichen Aktionen in einer Situation erlernen können und bei einer stochastischen Auswahl öfter die schlechteren Aktionen wählen.

Die Frage, ob die Agenten in der Lage sind, die jeweilige optimale Strategie eines stochastischen Spiels zu lernen beziehungsweise sich dieser anzunähern, kann nach den Ergebnissen dieser Arbeit für beide Spiele bejaht werden. Die Teilfrage, ob sie in bestimmten Situationen die gleichen Aktionen, wie die optimale Strategie wählen, kann für Kuhn Poker klar bejaht werden, da für alle Agenten eine optimale Aktionstabelle erfasst werden konnte. Bei Blackjack konnte keine Aktionstabelle erfasst werden, doch für die Auswertung nach zufälligen Situationen nach der Basic Strategy konnte eine Übereinstimmung der Aktionen in über 87% der Fälle gemessen werden. Das heißt der Agent konnte die Basic Strategy zwar nicht komplett erlernen, aber er konnte sich dieser wesentlich annähern. Bei den beiden Teilfragen zum Gewinn- oder Verlustwert, konnten die Ergebnisse zeigen, dass die Agenten sich in Kuhn Poker bis auf die Abweichungen aufgrund der stochastischen Eigenschaften des Spiels an die Verlustwerte der optimalen Strategie annähern konnten. In Blackjack konnte der Agent sich nicht perfekt an den Verlustwert der Basic Strategy annähern, jedoch konnte der Agent die Strategie gut genug lernen, um ein Vielfaches näher an die Basic Strategy zu kommen als zum Beispiel ein zufällig agierender Agent oder die Zahlenwerte aus der Literatur für einen durchschnittlichen menschlichen Blackjack Spieler.

Die Frage, ob es einen Unterschied macht, ob die Agenten stochastisch oder deterministisch agieren, kann ebenfalls klar bejaht werden. Anders als in der anfänglichen Erwartung macht es in beiden Spielen einen Unterschied, ob die Agenten stochastisch oder deterministisch agieren, und in beiden Spielen haben die Agenten schlechter abgeschnitten, wenn sie stochastisch agiert haben.

Die Frage, ob die Agenten gegen einen nicht optimal spielenden Gegner optimal ausnutzend agieren können, konnte in dieser Arbeit leider nicht beantwortet werden. Das Thema der Exploitation könnte jedoch ein spannender Einstiegspunkt für eine Weiterführung dieser Forschung sein.

Eine weitere Möglichkeit das Thema fortzusetzen wäre, Kuhn Poker als Pettingzoo Environment zu implementieren. Damit könnten Experimente einfacher umgesetzt werden, die zum Beispiel unterschiedliche Agenten im direkten Vergleich antreten lassen. Außerdem könnte damit ein exploitable-Agent einfacher unabhängig vom Environment entwickelt werden.

Außerdem könnte eine Java-Python-Bridge implementiert werden, um die Möglichkeit zu eröffnen, komplexere Agenten, die oft in Python implementiert sind, wie auch die der Stable-Baselines3 Bibliothek, direkt im GBG-Framework zu testen, sodass nicht für jedes Spiel ein entsprechendes Environment in Python entwickelt werden muss.

Literaturverzeichnis

888 Poker. (26. 11 2023). *888 Poker*. Abgerufen am 30. 04 2024 von Wie spielt man mit einer exploitativen Poker Strategie | 888 Poker: https://www.888poker.de/magazine/strategy/exploitative-strategie

Baldwin, R. R., Cantey, W. E., Maisel, H., & McDermott, J. P. (1956). The Optimal Strategy in Blackjack. *Journal of the American Statistical Association 51.275*, 429-439.

Bowling, M., Burch, N., Johanson, M., & Tammelin, O. (2015). Heads-up limit hold'em poker is solved. *Science 347.6218*, S. 145-149.

Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., & Zaremba, W. (2016). *OpenAI Gym.* arXiv preprint. Abgerufen am 04. 07 2024 von https://arxiv.org/abs/1606.01540

Brown, N., & Sandholm, T. (2019). Superhuman AI for multiplayer poker. *Science 365.6456*, 885-890.

Buzzi, A. (2020). *The statistics of Blackjack & optimal strategy*. Abgerufen am 30. 04 2024 von Towards Data Science: https://towardsdatascience.com/the-statistics-of-blackjack-e3b5fc29e67d

DATA Lab. (2019). *RLCard: A Toolkit for Reinforcement Learning in Card Games -- RLCard 0.0.1 documentation*. Abgerufen am 30. 04 2024 von RLCard documentation: https://rlcard.org/

elliottower. (11. 04 2024). *[Bug Report] SB3 Connect four tutorial does not train properly. Issue #1147 Farama Foundation*. Abgerufen am 03. 07 2024 von PettingZoo GitHub: https://github.com/Farama-Foundation/PettingZoo/issues/1147#issuecomment-1866408508

Farama Foundation. (2023). *Gymnasium Documentation*. Abgerufen am 30. 04 2024 von Gymnasium Documentation: https://gymnasium.farama.org/

Farama Foundation. (2024). *PettingZoo Documentation*. Abgerufen am 30. 04 2024 von PettingZoo Documentation: https://pettingzoo.farama.org/

Hoehn, B., Southey, F., & Holte, R. C. (2009). Effective Short-Term Opponent Exploitation in Simplified Poker. *Machine Learning 74*, S. 159-189. doi:10.1007/s10994-008-5091-5

Konen, W. (2019). General Board Game Playing for Education and Research in Generic AI Game Learning. *IEEE Conference on Games (CoG)* (S. 1-8). London: IEEE.

Kuhn, H. W. (1950). A simplified two-person poker. In H. W. Kuhn, & A. W. Tucker, *Contributions to the Theory of Games (AM-24), Volume I* (S. 97-103). Princeton: Princeton University Press.

Marcus, T. (2024). Implementation von Blackjack und Kuhn Poker als Gymnasium Environments. *(Praxisprojekt)*.

Meißner, S. (2021). Untersuchung des Spiel- und Lernerfolgs künstlicher Intelligenzen für ein nichtedeterministisches Speil mit imperfekten Informationen. *(BA Thesis)*. Gummersbach, Deutschland: TH Köln. Abgerufen am 03. 07 2024 von https://www.gm.fh-koeln.de/~konen/research/PaperPDF/BA-Meissner-final-2021.pdf

OpenAI. (25. 06 2024). *Models - OpenAI API*. Abgerufen am 18. 06 2024 von OpenAI API: https://platform.openai.com/docs/models

Raffin, A., Hill, A., Gleave, A., Kanervisto, A., Ernestus, M., & Dormann, N. (2021). Stable-Baselines3: Reliable Reinforcement Learning Implementations. *Journal of Machine Learning Research 22.268*, 1-8. Abgerufen am 04. 07 2024 von https://jmlr.org/papers/volume22/20-1364/20-1364.pdf

Silver, D., Huang, A., & Maddison, C. e. (2016). Mastering the game of Go with deep neural networks and tree search. *Nature 529.7587*, 484-489. doi:https://doi.org/10.1038/nature16961

Thorpe, E. O. (1966). *Beat The Dealer A Winning Strategy For The Game Of Twenty One.* New York: Random House.

Vos, T. (2022). Reinforcement Learning and Evolutionary algorithms in the Stochastic Environment of Blackjack. *(BA Thesis)*. Groningen, Niederlande: Reichsuniversität Groningen. Abgerufen am 03. 07 2024 von https://fse.studenttheses.ub.rug.nl/27515/1/BachelorThesiss3162443ThomasVos.pdf

Wizard of Odds. (2010). *Blackjack Basic Strategy*. Abgerufen am 05. 07 2024 von Wizard of Odds: https://wizardofodds.com/games/blackjack/strategy/calculator/

Zeh, T. (2021). Untersuchung von Allgemeinen KI-Agenten für das Spiel Poker im General Board Games Framework. *(MA Thesis)*. Gummersbach, Deutschland: TH Köln. Abgerufen am 03. 07 2024 von https://www.gm.fh-koeln.de/~konen/research/PaperPDF/MA_Zeh_final_Poker-GBG-2021.pdf

Anhang

Das GitHub Repository mit den Environments und Evaluationen:

https://github.com/Amonshi284/BlackjackKuhnPokerRL

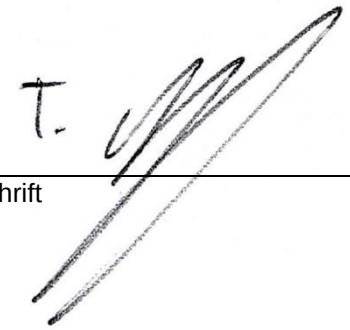
Erklärung

Ich versichere, die von mir vorgelegte Arbeit selbstständig verfasst zu haben. Alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten oder nicht veröffentlichten Arbeiten anderer oder der Verfasserin/des Verfassers selbst entnommen sind, habe ich als entnommen kenntlich gemacht. Sämtliche Quellen und Hilfsmittel, die ich für die Arbeit benutzt habe, sind angegeben. Die Arbeit hat mit gleichem Inhalt bzw. in wesentlichen Teilen noch keiner anderen Prüfungsbehörde vorgelegen.

Anmerkung: In einigen Studiengängen findet sich die Erklärung unmittelbar hinter dem Deckblatt der Arbeit.

Overath, 05.07.2024

Ort, Datum



Unterschrift