Hochschule für angewandte Wissenschaften Coburg
Fakultät Elektrotechnik und Informatik

Studiengang: Informatik Bachelor

Bachelorarbeit

KI-Entwicklung für das Spiel "Ganz schön clever": Ein Deep Reinforcement Learning Ansatz

Schubert, Sander

Abgabe der Arbeit: 26.10.2023

Betreut durch:

Prof. Dr. Mittag, Florian, Hochschule Coburg

Inhaltsverzeichnis

1 Zusammenfassung

Zunehmend prägen das maschinelle Lernen und KI die Arbeit und das Leben der Menschen. Besonders präsent sind im Jahr 2023 unter Anderem potente Chatbots wie ChatGPT 4. Solche Tools ermöglichen es Benutzern komplexe sowie komplizierte Aufgaben deutlich einfacher und schneller abzuarbeiten. Hervorzuheben ist hierbei auch, dass man mit solchen Tools deutlich weniger Fachwissen benötigt um in einem Bereich aufgaben effizient lösen zu können, da es einem eine Vielzahl von Informationen zum gewünschten Thema auf anfrage bereitstellen kann. Je komplexer das Problem oder die Fragestellung allerdings sind, desto unverlässlicher werden diese Tools. Man muss seine Anfragen deshalb möglichst präzise formulieren und die Problemstellung in für das Tool angemessene Teilaufgaben zerlegen.

Auch in der Spielentwicklung spielen maschinelles Lernen und KI schon seit langem eine bedeutende Rolle. In den meisten Spielen gibt es sogenannte Bots, welche man als KI bezeichnen kann. Diese sollen bestimmte Aufgaben im Spiel erfüllen um den Spieler zu unterstützen oder im als Widersacher zu dienen. Je komplexer die Aufgabe, umso schwerer ist es einen solchen Bot zu erstellen, welcher die Aufgabe auf zufriedenstellende Weise erfüllen kann.

Das Gesellschaftsspiel "Ganz schön clever" ist ein Würfelspiel, welches eine hohe Komplexität aufweist. Diese kommt vor allem durch die vielen Aktionsmöglichkeiten des Spielers und die multiplen zusammenhängen innerhalb des Belohnungssystems zustande. Außerdem weist es eine hohe Stochastizität auf, welche die Komplexität weiter erhöht. Ziel dieser Arbeit ist es eine KI beziehungsweise einen Bot für dieses Spiel zu entwickeln, der das Spiel effizient spielen kann, sowie zu analysieren welche Aspekte der Entwicklung dabei relevant und zu beachten sind.

Dazu mussten Spielumgebung sowie KI zunächst implementiert werden. Dies geschah mithilfe von Bibliotheken wie Stable-Baselines3 und Gymnasium. Insgesamt ergab sich dabei, dass sich mithilfe des PPO-Algorithmus von Stable-Balseslines3 auf relativ einfache Weise ein effizientes Modell für das Spiel entwickeln lässt.

2 Einleitung

2.1 Hinführung zum Thema

In den vergangenen Jahren gewann maschinelles Lernen und insbesondere auch KI zunehmend an Bedeutung, Tendenz steigend. Im Jahr 2023 ist eines der am meisten präsenten neuen Tool ChatGPT 4. Dieses Tool ist ein Chat-Bot, welches es dem Benutzer ermöglicht mit ihm zu kommunizieren und ihm, dem Chat-Bot, vor allem Fragen oder Aufgaben zu stellen. Solche Tools ermöglichen es Benutzern zunehmend ihre Tätigkeiten zu vereinfachen und prägen somit das Leben der Menschen zunehmend. Auch in dieser Arbeit wurde ChatGPT 4 als unterstützendes Tool verwendet. Es wurde vor Allem dafür benutzt um Fachliche Fragen zu beantworten, aber auch anfangs um Code für den Prototypen zu generieren. Mit zunehmender Komplexität der zu bearbeitenden Aufgabe sinkt die Verlässlichkeit solcher Tools. Daher ist es wichtig die Anfragen an den Chat-Bot möglichst präzise zu formulieren und den Aufgabenbereich angemessen einzuschränken um das Tool nicht zu überfordern.

Auch in der Spielentwicklung nimmt das maschinelle Lernen und KI schon seit langem eine zentrale Rolle ein. In den meisten spielen gibt es eine oder meist mehrere Künstliche Intelligenzen, welche bestimmte Aufgaben erfüllen um den Spieler bei Spiel zu unterstützen oder ihm als Widersacher zu dienen. Auch hier gilt je komplexer die Aufgabenstellung desto schwieriger ist es einen solchen Bot zu generieren, welcher diese effizient und richtig lösen kann.

Das Gemeinschaftsspiel "Ganz schön clever" ist ein Würfelspiel, welches eine hohe Komplexität aufweist. Diese kommt vor allem durch die hohe Anzahl an möglichen Aktionen (der sogenannte Aktionsraum) für den Spieler und die vielen Zusammenhänge des Belohnungssystems im Spiel zu Stande. Das Spiel weist allerdings auch eine hohe Stochastizität auf, welche die Komplexität zusätzlich erhöht.

Interessant ist wie man für ein solch komplexes Spiel einen Bot oder eine KI entwickeln kann um dieses effizient spielen zu können. Ist die Komplexität möglicherweise zu groß um vom Bot erfasst zu werden und wenn nicht, wie kann man einen solchen Bot implementieren und was gilt es dabei zu beachten?

2.2 Zielsetzung

Ziel der Arbeit ist es einen Bot beziehungsweise eine KI zu entwickeln, welche das Spiel "Ganz schön clever" möglichst effizient spielen kann. Dabei soll analysiert und erforscht werden, welche Aspekte es dabei zu beachten gilt und wie sich unterschiedliche Ansätze auf das Verhalten und die Performance des Modells (des Bots) auswirken.

In den vergangenen Jahren hat sich viel getan, weshalb deutlich mehr möglich geworden ist. Mit neuen Möglichkeiten ergeben sich auch bessere oder einfachere Ansätze, die zu einem gewünschten Ergebnis führen. Ziel ist es auch einen solchen Ansatz zu finden und zu vervollständigen.

Es gibt des Weiteren noch keine Untersuchungen zu einer Spiel-KI für das Spiel "Ganz schön clever" daher ist es interessant Aufschlüsse darüber zu gewinnen welche Schwierigkeiten sich hierbei ergeben und wie man diese überwinden kann.

2.3 Aufgabenstellung

Es ist eine KI für das Spiel "Ganz schön clever" zu implementieren. Hierbei sollen der Vorgang sowie Ergebnisse des Prozesses analysiert und bewertet werden. Hierzu wird zunächst ein Prototyp entwickelt, welcher eines der fünf Felder des Spiels beinhaltet. Dieser soll Einsichten über die Machbarkeit und die Rahmenbedingungen des Projektes geben. Daraufhin werden das Modell und die Spielumgebung schrittweise um ihre jeweiligen Funktionalitäten erweitert, bis das Spiel vollständig und möglichst effizient von der KI gespielt werden kann. Dieser Prozess wird analysiert und bewertet um Schlüsse darüber zu ziehen was vorteilhaft und was nachteilig für ein solches Vorhaben ist.

2.4 Aufbau der Arbeit

3 Grundlagen

3.1 Allgemeine Grundlagen

3.1.1 Ganz schön clever

Die folgende Abbildung zeigt das Spielbrett des Spiels "Ganz schön clever" zu dem im Rahmen dieser Arbeit eine KI implementiert werden soll:

Abb. 1: Ganz schön clever

Quelle: [Google Play Store, de.brettspielwelt.ganzschoenclever]

Im folgenden werden der Spielablauf und die wesentlichen Regeln des Spiels erklärt.

Es gibt sechs farbige Würfel, wobei jeder bis auf den weißen einem der 5 farbigen Spielfelder zuzuordnen ist. Der weiße Würfel ist ein Sonderwürfel und kann als einer anderen Würfel betrachtet werden. Wenn bestimmte Bedingungen erfüllt sind kann der Spieler einen Würfel wählen und das entsprechende Subfeld ausfüllen. Die Felder verfügen über Belohnungen, welche freigeschaltet werden, wenn eine bestimmte Kombination oder Anzahl an Feldern freigeschaltet worden ist. Beim orangenen und lila Feld kommt es bei der Belohnung zudem darauf an wie hoch das Würfelergebnis des gewählten Würfels ist, da die einzelnen Subfelder hier je nach Würfelergebnis zusätzlich belohnt werden.
Das Spiel teilt sich in bis zu sechs Runden mit jeweils bis zu frei Würfen ein. Nach jeder Runde

bekommt der Spieler zudem die Möglichkeit einen Würfel auf dem Tablett eines Mitspielers zu wählen. Diese Wahl verhält sich so wie bei der Wahl eines eigenen Würfels bei den Würfen des Spielers selbst. Die Anzahl der Runden werden von der Spielerzahl festgelegt. Bei ein bis zwei Spielern sind es sechs Runden. Bei drei Spielern sind es fünf Runden und bei vier Spielern sind es vier Runden. Die Anzahl der Würfe pro Runde ist immer drei, allerdings kann diese Anzahl reduziert werden, wenn kein Würfel mehr zum Würfeln zur Verfügung steht. Dann wird der Spielablauf fortgesetzt als hätte der Spieler seinen dritten Wurf in der Runde beendet und er kann einen Würfel vom Tablett des Mitspielers wählen bevor dann die neue Runde für ihn beginnt. Zu beginn der ersten, zweiten, dritten und vierten Runde bekommt jeder Spieler zudem eine Belohnung, welche oben rechts auf Abbildung 1 bei der jeweiligen Rundenzahl zu finden ist.
Es gibt zwei Arten von Belohnungen im Spiel. Punktebelohnungen und Boni. Bei Punktebelohnungen handelt es sich um Punkte welche auf dem Score des Spielers addiert werden, welcher am Ende des Spiels entscheidet wer gewonnen hat. Der Spieler mit dem höchsten Score gewinnt das Spiel. Punktebelohnungen erhält man beim gelben Feld indem man eine Spalte vollständig ausfüllt. Die Anzahl der Punkte findet sich am Ende der Spalte. Im blauen Feld mit zunehmender Anzahl der ausgefüllten blauen Subfelder. Die Anzahl der Punkte findet sich oben im blauen Feld. Im grünen Feld ebenso mit zunehmender Anzahl der ausgefüllten grünen Subfelder. Die Anzahl der Punkte findet sich auch hier oben im grünen Feld. Im orangenen und lila Feld muss man dafür lediglich ein Subfeld ausfüllen. Die Punktebelohnung entspricht der Augenzahl des entsprechenden ausgefüllten Subfeldes. Beim orangenen Feld wird diese Augenzahl bei einigen Feldern zusätzlich mit zwei oder drei multipliziert. Außerdem gibt es eine Boni Belohnung, welche indirekt eine Punktebelohnung darstellt. Es handelt sich hierbei um den sogenannten Fuchs beziehungsweise die Füchse. Die Anzahl der freigeschalteten Füchse wird am Ende des Spiels mit der Anzahl der erzielten Punkte des Feldes mit den niedrigsten erreichten Punktewert multipliziert und zum Gesamtpunktestand addiert.

Bei Boni handelt es sich um Belohnungen, welche der Spieler nutzen kann oder muss um sich im Spiel einen Vorteil zu verschaffen. Die Boni sind bei den entsprechenden Subfeldern beziehungsweise am Rande von Spalten und Zeilen eingezeichnet und können freigeschaltet werden indem man diese ausfüllt. Eine Ausnahme bildet hier die Boni welche beim gelben Feld freigeschaltet wird indem alle diagonalen Felder von links oben nach rechts unten ausgefüllt werden.

Jede Boni hat ihr eigenes Symbol. Nun folgt eine Aufzählung und Erklärung der verschiedenen Boni mit Ausnahme der Füchse. Boni werden bei der Benutzung aufbraucht. Man kann mehr als eine dieser Boni auf einmal besitzen.

Extra Wahl: Bei der Extra Wahl wird es dem Spieler ermöglicht am Ende seiner Würfe beziehungsweise nachdem er einen Würfel vom Silbertablett des Gegners gewählt hat erneut Würfel

zu wählen und die entsprechenden Felder dafür anzukreuzen. Würfel die so gewählt wurden können mithilfe der Extra Wahl Boni im selben Wurf nicht erneut gewählt werden. Es können alle Würfel gewählt werden, nicht nur die, welche unter normalen Umständen gültig zur Wahl stehen. Das Symbol ist die +1.

Neuer Wurf: Der Neue Wurf ermöglicht es dem Spieler einen seinen Würfe zu wiederholen ohne dabei einen der Würfel auszuwählen. Dies ermöglicht es ihm Würfe mit ungünstigen Ergebnissen neu auszurichten. Das Symbol sind die drei Pfeile die im Kreis angeordnet sind.

Gelbes Kreuz: Ermöglicht es dem Spieler direkt nach erhalten der Boni eines der gelben Subfelder nach eigener Wahl auszufüllen. Das Symbol ist ein Kreuz auf gelbem Hintergrund.

Blaues Kreuz: Ermöglicht es dem Spieler direkt nach erhalten der Boni eines der blauen Subfelder nach eigener Wahl auszufüllen. Das Symbol ist ein Kreuz auf blauem Hintergrund.

Grünes Kreuz: Ermöglicht es dem Spieler direkt nach erhalten der Boni das nächste freie Grüne Subfeld auszufüllen. Das Symbol ist ein Kreuz auf grünem Hintergrund.

Orangene Vier: Ermöglicht es dem Spieler direkt nach erhalten der Boni das nächste freie orangene Subfeld mit einer vier auszufüllen. Das Symbol ist eine vier auf orangenem Hintergrund.

Orangene Fünf: Ermöglicht es dem Spieler direkt nach erhalten der Boni das nächste freie orangene Subfeld mit einer fünf auszufüllen. Das Symbol ist eine fünf auf orangenem Hintergrund.

Orangene Sechs: Ermöglicht es dem Spieler direkt nach erhalten der Boni das nächste freie orangene Subfeld mit einer sechs auszufüllen. Das Symbol ist eine sechs auf orangenem Hintergrund.

Lila Sechs: Ermöglicht es dem Spieler direkt nach erhalten der Boni das nächste freie lila Subfeld mit einer sechs auszufüllen. Das Symbol ist eine sechs auf lila Hintergrund.

Nun folgen die Regeln nach denen bestimmt wird ob ein Würfel gewählt werden kann um ein Subfeld auszufüllen und ob er ein gültiger Würfel ist:

Ist ein Würfel ungültig kann dieser auch nicht gewählt werden. Würfel werden ungültig indem sie in dieser Runde bereits gewählt worden sind. Außerdem werden Würfel, welche eine geringere Augenzahl aufweisen als der aktuell gewählte Würfel auch automatisch ungültig. Eine Ausnahme ist hierbei die Wahl eines Würfels vom Silbertablett des Gegenspieler oder Wahlen von Würfeln durch die Extra Wahl Boni. Würfel werden wieder gültig am Anfang der Runde, nach Abschluss des dritten Wurfes (beinhaltet Wahl des Würfels), sowie nachdem Wahlen mit dem Extra Wahl Boni erfolgt sind.

Jedes Feld hat seine eigenen Regeln, die bestimmen, wann ein Subfeld ausgefüllt werden darf. Beim gelben Feld muss die Augenzahl des Würfel mit der Zahl des Subfeldes übereinstimmen. Beim Blauen Feld muss die Summe der Augenzahlen des blauen und des weißen Würfels mit der

Zahl des Subfeldes übereinstimmen. Beim grünen Feld muss die Augenzahl des Würfels größer oder gleich der Zahl im Subfeld sein. Zudem kann immer nur das nächste freie Feld ausgefüllt werden, beginnend von links. Im orangenen Feld kann immer das nächste Subfeld eingetragen werden. Auch hier beginnend von links. Beim lila Feld muss die Augenzahl des Würfels größer sein als die Zahl im zuletzt ausgefüllten Subfeld. Eine Ausnahme bildet hier der Fall in dem eine sechs im zuletzt ausgefüllten Subfeld steht. Dann kann das nächste Feld mit jeder beliebigen Augenzahl gewählt werden. Die sechs setzt die Voraussetzung für das lila Feld bis zum nächsten ausfüllen sozusagen aus. Auch hier gilt die Reihenfolge von links nach rechts.

3.1.2 Machine Learning

"Maschinelles Lernen heißt, Computer so zu programmieren, dass ein bestimmtes Leistungsmerkmal anhand von Beispieldaten oder Erfahrungswerten optimiert wird" [Maschinelles Lernen, Seite 3].

Es gibt bis heute nach wie vor viele Problemstellungen, die von Menschen auf einfache Art und weise lösbar sind, für die es aber keine Algorithmische Lösung zu geben scheint. Hier kommt das maschinelle Lernen zum Einsatz. Durch die Mustererkennung aus Trainingsdaten können Programme lernen solche Problemstellungen zu lösen indem sie präzise Vorhersagen über bestehende Sachverhalte aus beliebigen Daten des selben oder eines ähnlichen Sachverhaltes, der beim Training vorhanden war, zu treffen. Ein besonders weit verbreiteter Anwendungsfall ist die Herleitung von Kundenverhalten und möglicher Optimierungsmöglichkeiten für den Verkauf. Wenn man ein Programm mithilfe von maschinellem Lernen trainiert hat, nennt man dieses dann Modell. Ein solches Modell wird häufig erst auf allgemeinen Datensätzen und später auf immer spezifischeren trainiert, sodass es schließlich auf eine konkrete Aufgabe zugeschnitten werden kann [Maschinelles Lernen, Seite 1f].

Maschinelles Lernen ermöglicht es zwar nicht einen gesamten Prozess mit all seinen Einzelheiten zu verstehen, aber es ermöglicht relevante Merkmale zu erkennen und Schlüsse über den gesamten Sachverhalt zu schließen und auf Grund dessen zu agieren zu können oder zumindest Vorhersagen über diesen zu treffen [ML, Seite 2].

Die Anwendungsgebiete von maschinellem Lernen sind zahlreich. Unter anderem ist es relevant für den Einzelhandel und Finanzdienstleister, um Kreditgeschäfte abzuwickeln, Betrugsversuche zu erkennen, oder den Aktienmark einzuschätzen. Aber auch Fertigung wird es zur Optimierung, Steuerung und Fehlerbehebung eingesetzt. Auch in der Medizin erweisen sich medizinische Diagnoseprogramme mithilfe von Modellen, die mit maschinellem Lernen trainiert wurden als nützlich [ML, Seite 3].

Und das sind nur einige wenige der möglichen Bereiche in denen maschinelles Lernen bereits Anwendung findet.

Die Datenbestände und das World Wide Web werden immer größer und die Suche nach relevanten Daten kann nicht mehr manuell vorgenommen werden [ML, Seite 3].

"Das maschinelle Lernen ist aber nicht nur für Datenbanken relevant, sondern auch für das Gebiet der künstlichen Intelligenz" [ML, Seite 3].

Von Intelligenz spricht man dann, wenn das System selbstständig in einer sich verändernden Umgebung lernen und sich anpassen kann. Dadurch muss der Systementwickler nicht jede erdenkliche Situation vorhersehen und passende Lösungen dafür entwickeln [ML, Seite 3].

Maschinelles Lernen findet seine Anwendung in dieser Arbeit in Form von Deep Reinforcement Learning. Was Reinforcement Learning ist und wie es sich von Deep Reinforcement Learning unterscheidet wird im Folgenden beschrieben.

3.1.3 Reinforcement Learning

Reinforcement Learning (im deutschen Bestärkendes Lernen) heißt so, weil es die Aktionen des Agenten (beziehungsweise des Modells) bestärkt. Man kann sich das in etwa so vorstellen, wie das Training eines Hundes im Park. Dieser wird jedes mal wenn er einen Trick richtig ausführt mit einem Leckerli belohnt. Diese Belohnung bestärkt das Verhalten des Hundes und das Tier lernt dieses in Zukunft zu wiederholen. Eine negative Aktion kann hingegen bestraft werden, damit sie in Zukunft nicht wiederholt wird [RL, Seite 11].

Im Reinforcement Learning sind vor allem Folgende Begriffe wichtig:

Agent (oder Modell): Dabei handelt es sich um die Entität, welche mit der Umgebung interagiert und die Entscheidungen trifft. Dabei kann es sich zum Beispiel um einen Roboter oder autonomes Fahrzeug handeln [RL, Seite 11]. In dieser Arbeit ist diese Entität ein Modell, welches mithilfe der Bibliothek Stable-Baselines3 erstellt wird.

Umgebung: Dabei handelt es sich um die Außenwelt des Agenten [RL, Seite 11]. der Agent interagiert mit dieser und erhält je nach Zustand der Umgebung und seiner gewählten Aktion ein entsprechendes Feedback.

Aktion: Eine Aktion beschreibt das Verhalten des Agenten [RL, Seite 11]. In dieser Arbeit wählt der Agent Felder des Spielbrettes zum ausfüllen als Aktionen aus. Außerdem entscheidet er ob er bestimmte Boni nutzen möchte oder nicht.

Zustand: Der Zustand beschreibt den Zusammenhang zwischen Umgebung und Agent [RL, Seite 11]. In dieser Arbeit ist der Zustand von den Eigenschaften des Spielbrettes, der Würfel, der Rundenanzahl und der erspielten Boni abhängig.

Belohnung: Positive oder negative Vergeltung je nachdem wie gut der Zustandswechsel von Zustand x nach Zustand y gewesen ist [RL, Seite 11]. In dieser Arbeit wird dies durch die

jeweiligen Punktebelohnungen im Spiel verkörpert. Eine Ausnahme bildet hier eine negative Belohnung, wenn der Agent in einen Zustand gerät in dem er keine gültige Aktion tätigen kann.

Policy: Die Policy ist die Strategie des Agenten, nach welcher er seine nächsten Aktionen wählt [RL, Seite 11]. In dieser Arbeit wird die Policy durch ein Multilayer Perceptron (siehe Deep Learning) abgebildet.

Episode: Eine Menge an Zusammenhängenden Aktionen, welche endet, wenn das Ziel erreicht worden ist [RL, Seite 11]. In dieser Arbeit ist eine Episode ein kompletter Spieldurchlauf von Ganz schön clever.

Die folgende Abbildung beschreibt einen Lernzyklus im Reinforcement Learning:

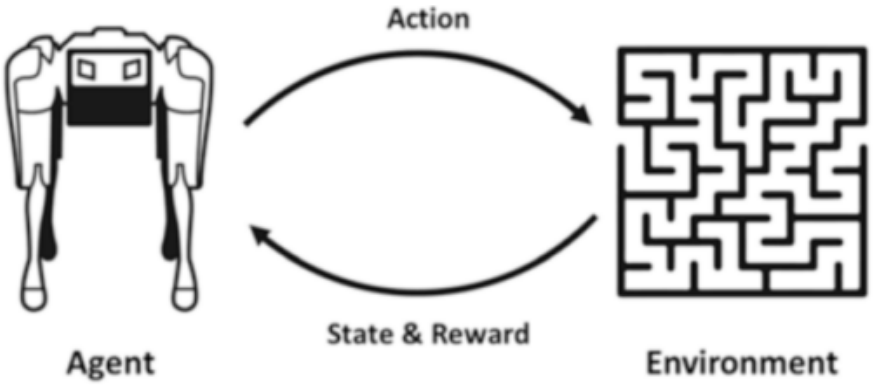

Abb. 2: Reinforcement Learning

Quelle: [RL, Seite 12]

Der Agent führt eine Aktion in der Umgebung aus und erhält daraufhin eine Belohnung und den neuen Zustand der Umgebung als Feedback. Daraufhin aktualisiert er seine Policy, um in Zukunft bessere Aktionen tätigen zu können.

Hierbei ist das Ziel des Agenten die gesamte erreichte Belohnung zu maximieren. Demnach wird die Policy dementsprechend angepasst, dass dies begünstigt wird [RL, Seite 12f].

Doch dabei gibt es einiges Schwierigkeiten, die es zu beachten gilt. Belohnungen die in kurzer Zeit erreicht werden können, könnten wichtiger oder weniger wichtig sein als Belohnungen, die erst innerhalb vieler Schritte erreicht werden können. Ein gutes Beispiel hierfür wäre ein Balanceakt, bei dem es besonders wichtig ist kurzzeitige Belohnungen zu bevorzugen, da die Episode endet, wenn das Balancieren fehlschlägt, was eine starke negative Belohnung zur Folge haben kann.

Ein weiterer Faktor ist die Balance zwischen Erkundung und Ausbeutung. Diese zwei Prinzipien sind wesentlich für das Reinforcement Learning. Dabei geht es darum, wie stark die Policy es

vorzieht neue oder selten gesehene Zustände und Aktionen auszuprobieren oder bereits bekannte, welche eine gute Belohnung zu bringen scheinen auszunutzen beziehungsweise auszubeuten [RL, Seite 13].

3.1.4 Deep Learning

Was unterscheidet Reinforcement Learning von Deep Reinforcement Learning? Im wesentlichen handelt es sich um das selbe Konzept, allerdings ist das eine Deep Learning und das andere nicht. Was ist also Deep Learning?

Von Deep Learning spricht man sobald ein Neuronales Netz mehrere versteckten Schichten hat.

Ein solches Netz besteht aus einer Vielzahl von Neuronen. [DRL, Seite 75] Ein solches Neuron setzt sich zusammen aus Inputs, Outputs, Gewichtungen dieser In- und Outputs, sowie einer Aktivierungsfunktion, welche auch gewichtet sein kann. Ein Spezialfall eines solchen Neuronalen Netzes ist ein sogenanntes Multilayer Perceptron. Bei diesem sind alle Neuronen in einer Schicht mit allen Neuronen der folgenden Schicht verbunden. Häufig haben auch alle Neuronen der versteckten Schichten die selbe Aktivierungsfunktion. Was den Beitrag der einzelnen Neuronen zum Gesamtergebnis des Netzes steuert sind im wesentlichen die unterschiedlichen Gewichtungen der Neuronen und ihre Position im Netz.

Die folgende Abbildung zeigt ein solches Multilayer Perceptron:

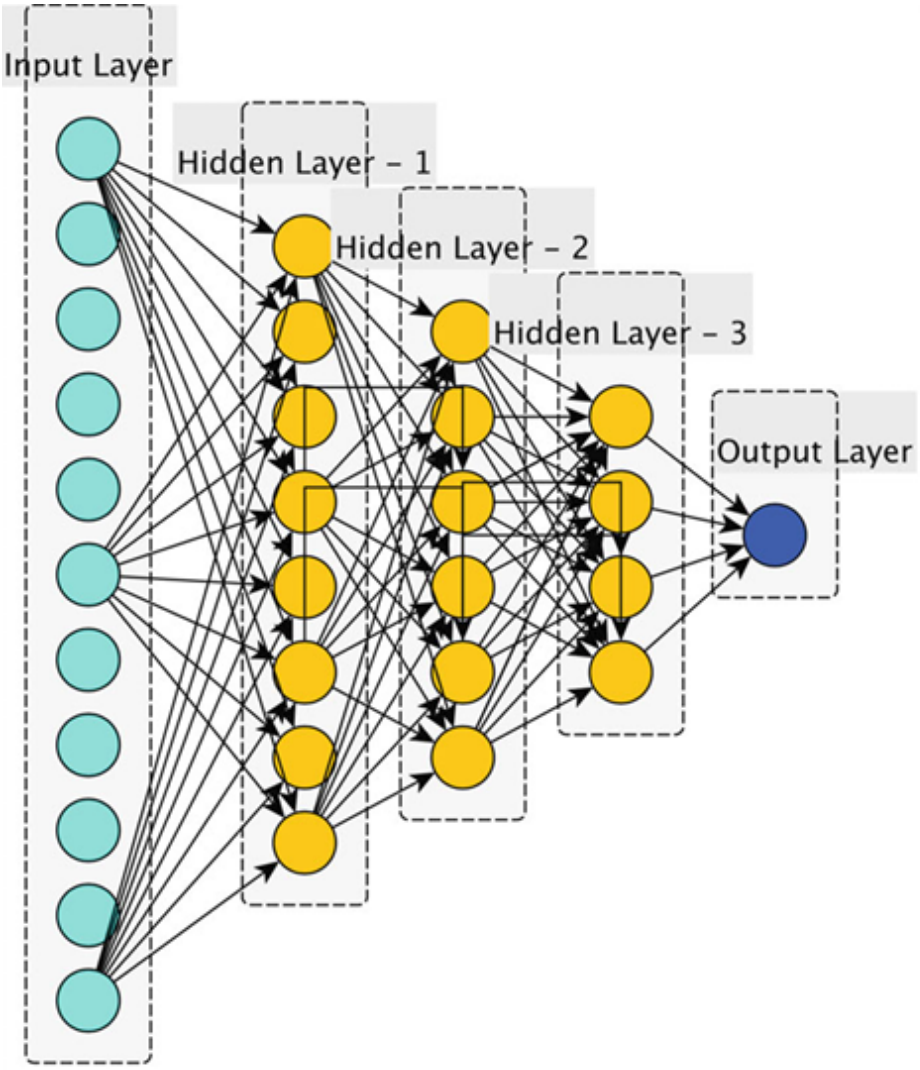

Abb. 3: Multilayer Perceptron
Quelle: [DRL, Seite 78]

Die Inputs des Netzes sind die Werte von Variablen der zur Verfügung stehenden Beobachtungen der Umgebung. Die versteckten Schichten verarbeiten diese Inputs dann mit ihren Funktionen und Gewichtungen. Das Output des Netzes ist hingehen eine Gewünschte Vorhersage, die mit ihrer eigenen Aktivierungsfunktion hergeleitet wird.

3.1.5 Proximal Policy Optimization

Proximal Policy Optimization oder kurz PPO ist eine neue Policy Gradient Methode für Reinforcement Learning. Im Gegensatz zu standardmäßigen Policy Gradient Methoden ist es mit dieser Methode möglich mehrere Policy Updates pro Datenpaket durchzuführen. PPO hat einige der Vorteile der Trusted Region Policy Optimization (kurz TRPO), ist aber simpler zu implementie-

ren und hat auch andere Vorteile, wie bessere Generalisierbarkeit und niedrigere Stichproben Komplexität. PPO zeigt eine gute Balance zwischen Stichproben Komplexität, Einfachheit und Trainingsgeschwindigkeit [PPO, Seite 1].

Die Stichprobenkomplexität beschreibt wie groß ein Datenpaket sein muss, um dem Algorithmus ein gewisses Level an Performance zu ermöglichen. Dies ermöglicht es mehr Updates mit weniger Gesamtdaten durchzuführen, was vor allem hilfreich ist, wenn nicht genügend Daten zur Verfügung stehen.

Policy Gradient Methoden berechnen einen Gradienten und passen die Policy in Richtung der negativen Steigung an. Das kann man sich ähnlich wie einen Punkt auf einer Parabel vorstellen, die Policy entspricht dem Punkt und wird so lange angepasst beziehungsweise verschoben, bis sie möglichst das Minimum der Parabel erreicht.

Bei der Trusted Region Policy Optimization gibt es eine vertrauenswürdige Region innerhalb die Policy abgeändert werden darf. Das soll sicherstellen, dass die Policy nicht zu stark abgeändert wird, um ein stabileres Training zu gewährleisten. Im Gegensatz zur PPO werden Änderungen, die zu stark abweichen verworfen.

Bei der PPO kommt es zum sogenannten Clipping. Hierbei werden zu starke Änderungen abgeschnitten im übertragenen Sinne und es kommt zu einer abgeschwächten Änderung der Policy.

PPO ist ein verhältnismäßig simpler, einfach verstehender und dennoch effizienter Algorithmus. Er hat eine gute Balance zwischen Stabilität und Effizienz. Außerdem erzielt er gute Ergebnisse bei einer großen Bandbreite an Aufgaben. Daher eignet sich PPO besonders gut für Einsteiger, die bisher nicht viel mit Deep Reinforcement Learning gearbeitet haben.

3.2 Verwendete Technologien

3.2.1 Gymnasium

3.2.2 Stable Baselines 3

3.2.3 Matplotlib

3.2.4 ChatGPT 4

4 Anforderungen und Konzeption

4.1 Anforderungen

4.1.1 Das Spiel

4.1.2 Die AI

4.1.3 Rahmenbedingungen

4.2 Konzeption

4.2.1 Die Spielumgebung

4.2.2 Die AI

5 Implementierung

5.1 Spielumgebung

5.1.1 Klassenattribute

5.1.2 Methoden

5.1.3 Einzelne Methoden...

5.2 AI

5.2.1 Model Learn

5.2.2 Model Predict

5.2.3 Init Envs

5.3 Darstellung

5.3.1 Make Entry

5.3.2 Plot History

5.4 Verwendung

6 Ergebnisse

6.1 Trainingshistorie

6.1.1 Version 1.1.0

6.1.2 Version 2.0

6.1.3 Version 3.0

6.1.4 Version 4.0

6.2 Finale Ergebnisse

6.2.1 Performance

6.2.2 Hyperparameter

6.2.3 ChatGPT 4

Literaturverzeichnis

HOCHSCHULE COBURG

Persönliche Angaben / *Personal details*

Schubert, Sander

Familienname, Vorname / *Surnames, given names*

02.12.1994

Geburtsdatum / *Date of birth*

Informatik Bachelor

Studiengang / *Course of study*

01550217

Matrikelnummer / *Student registration number*

Eigenständigkeitserklärung

Declaration

Hiermit versichere ich, dass ich diese Arbeit selbständig verfasst und noch nicht anderweitig für Prüfungszwecke vorgelegt habe. Ich habe keine anderen als die angegeben Quellen oder Hilfsmittel benutzt. Die Arbeit wurde weder in Gänze noch in Teilen von einer Künstlichen Intelligenz (KI) erstellt, es sei denn, die zur Erstellung genutzte KI wurde von der zuständigen Prüfungskommission oder der bzw. dem zuständigen Prüfenden ausdrücklich zugelassen. Wörtliche oder sinngemäße Zitate habe ich als solche gekennzeichnet.
Es ist mir bekannt, dass im Rahmen der Beurteilung meiner Arbeit Plagiatserkennungssoftware zum Einsatz kommen kann.
Es ist mir bewusst, dass Verstöße gegen Prüfungsvorschriften zur Bewertung meiner Arbeit mit „nicht ausreichend“ und in schweren Fällen auch zum Verlust sämtlicher Wiederholungsversuche führen können.

I hereby certify that I have written this thesis independently and have not submitted it elsewhere for examination purposes. I have not used any sources or aids other than those indicated The work has not been created in whole or in part by an artificial intelligence (AI), unless the AI used to create the work has been expressly approved by the responsible examination board or examiner. I have marked verbatim quotations or quotations in the spirit of the text as such.

I am aware that plagiarism detection software may be used in the assessment of my work.

I am aware that violations of examination regulations can lead to my work being graded as "unsatisfactory" and, in serious cases, to the loss of all repeat attempts.

Unterschrift Studierende/Studierender / *Signature student*

, den 26.10.2023

Ort, Datum / *Place, date*