jugend forscht

Projekttitel:

Problematisches Verhalten von KI-Agenten aus dem Bereich des bestärkenden Lernens

Teilnehmer:

Nico Hillbrand (18 Jahre, 15.05.2002)

Erarbeitungsort:

Gymnasium „In der Wüste“, Osnabrück

Projektbetreuer:

Katrin Lückmann-Fragner und Oliver Kunde

Thema des Projekts:

Künstliche Intelligenz und Ethik

Fachgebiet:

Mathematik/Informatik

Wettbewerbssparte:

Jugend forscht

Bundesland:

Niedersachsen

Wettbewerbsjahr:

2021

Problematisches Verhalten von KI-Agenten aus dem Bereich des bestärkenden Lernens

Ein Projekt von Nico Hillbrand

Kurzfassung

In meinem Projekt wird sich der Frage gewidmet, ob für Algorithmen aus dem Bereich des bestärkenden Lernens kleine Unterschieden in der Spezifikation der Nutzenfunktion zu ungewolltem Verhalten führen können.

Um dies zu untersuchen wird eine virtuelle Testumgebung programmiert, in der ein simulierter Putzroboter die Aufgabe bekommt Müll aufzusammeln und diesen in einen Mülleimer zu befördern.
Der Roboter wird von einer Kamera überwacht, die ihn stoppt, falls dieser mehr Müll in den Mülleimer befördern möchte als die Kapazität des Mülleimers zulässt. Die Simulation beinhaltet außerdem die Möglichkeit, dass die Kamera durch einen Zusammenstoß mit dem Roboter zerstört wird.

Es wird daraufhin durch das Trainieren vieler Q-learning Agenten untersucht, ob diese mit einer Nutzenfunktion, die das Entsorgen einzelner Mülleinheiten belohnt, lernen die Kamera absichtlich zu zerstören, um mehr Müll entsorgen zu können.

Inhaltsverzeichnis

1. Einleitung

Künstliche Intelligenz (KI) Systeme sind heutzutage überall. Sie werden eingesetzt bei Kreditentscheidungen von Banken, bei der ärztlichen Diagnose (Widmer et al., 2014) sowie in der medizinischen Forschung (Hutson, 2019). Außerdem benutzt man sie täglich bei der Internetsuche, Textvervollständigungen und wahrscheinlich bald auch beim Autofahren (Sulaiman, 2018).

Die Motivation hinter dieser Arbeit ist, mögliche Gefahren der fortschreitenden Digitalisierung zu untersuchen. Dies ist eine sehr komplexe Thematik, die durch soziale, ökonomische und technische Faktoren beeinflusst wird. Da die Untersuchung eines Programms am besten in den Rahmen einer Jugend Forscht Arbeit passt, wurde sich dazu entschieden hauptsächlich die technischen Aspekte zu beleuchten. Zusätzlich werden die ethischen Implikationen der auftretenden Problematiken untersucht.

In dieser Arbeit wird das Verhalten einer speziellen Untergruppe der KI-Algorithmen untersucht. Ausgewählt wurde das Paradigma der Agenten, die in Markow-Entscheidungsprozessen den erwarteten Nutzen maximieren. Der verwendete Agent ist ein simpler mit Hilfe einer Q-Table implementierter, Q-learning Algorithmus. Er befindet sich in einer Gitterwelt, in der er die Aufgabe erhält Müll aufzuräumen. Diese Aufgabe wird ihm durch seine Nutzenfunktion übermittelt. Die Nutzenfunktion des Menschen unterscheidet sich jedoch geringfügig von der des KI-Agenten. Es wird nun untersucht, ob der Agent durch einen solchen Unterschied, der bei der Programmierung komplexerer Systeme schwer zu vermeiden scheint, problematisches Verhalten entwickelt.

2. Theoretischer Hintergrund

2.1 Agenten

2.1.1 Definition

Ein Agent ist etwas, das handelt. Agent kommt vom Lateinischen *agere* was sich als tun, handeln, machen übersetzen lässt. Natürlich tun alle Computerprogramme etwas. Von Computeragenten erwartet man aber noch zusätzlich, dass sie autonom operieren, ihre Umgebung wahrnehmen, über einen längeren Zeitraum beständig sind, sich an Änderungen anpassen sowie Ziele erzeugen und verfolgen (Russell und Norvig, 2009: 4). Agenten befinden sich in einer Umgebung, die sie durch Sensoren wahrnehmen können und durch Aktoren beeinflussen können.

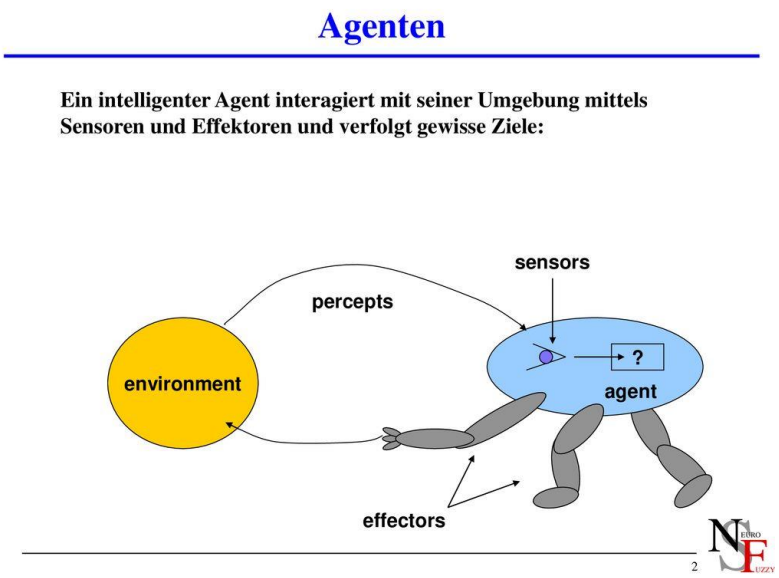

Abbildung 1: Das Agenten-Framework (Kruse, 2018: 2).

2.1.2 Markow-Entscheidungsprozesse

Agenten befinden sich häufig in Umgebungen, die als Markow-Entscheidungsprozess modelliert werden können. In einem Markow-Entscheidungsprozess (Puterman, 1994) gibt es die Menge von Zuständen S, die Menge an Aktionen A, die Transitionswahrscheinlichkeiten T, die Nutzenfunktion R und die Startverteilung p0. Die Wahrscheinlichkeit einen Zustand s‘ von einem Zustand s zu erreichen, darf bei einem Markow-Entscheidungsprozess nur von dem Zustand s und nicht von vorherigen Zuständen abhängig sein. Der Agent durchläuft den Markow-Entscheidungsprozess, indem er zunächst in den Anfangszustand versetzt wird und dann schrittweise eine Aktion ausführt, die zu einem neuen Zustand und einer mit Hilfe der Nutzenfunktion errechneten Belohnung führt. Dies wird wiederholt bis der Agent zu einem Endzustand gelangt.

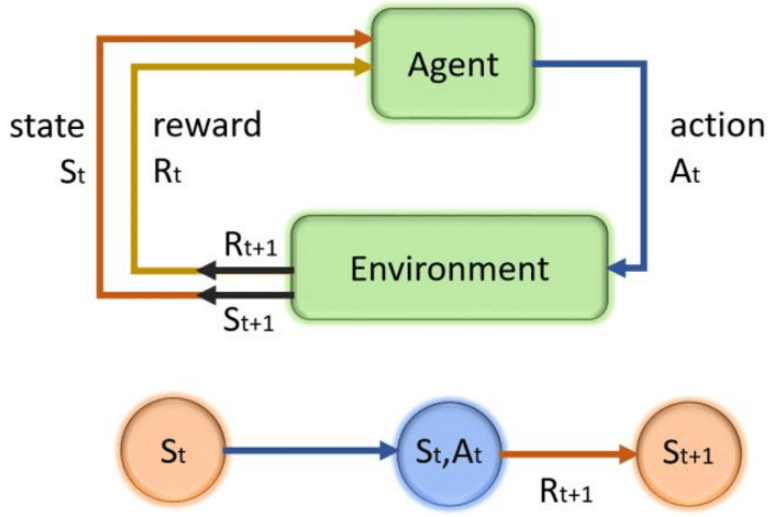

Abbildung 2: Markow-Entscheidungsprozesse (Zapata und Flores, 2019: 3).

2.2 Q-learning

Q-learning (Watkins und Dayan, 1992) ist ein Unterbereich des bestärkenden Lernens (Kaelbling et al., 1996). Q-learning ist Model Free, das heißt, der Agent hat kein eigenes Modell der Umwelt, sondern reagiert lediglich auf Zustände und Off-Policy, was bedeutet, dass der Agent zum Lernen nicht die gleiche Policy, also Strategie, wie zum Handeln verwendet. Q-learning basiert auf der zentralen Idee des Optimalitätsprinzips von Bellmann, welches er in seinem Buch *Dynamic Programming* so beschrieb: „*Eine optimale Entscheidungsfolge hat die Eigenschaft, dass, wie auch immer der Anfangszustand war und die erste Entscheidung ausfiel, die verbleibenden Entscheidungen eine optimale Entscheidungsfolge bilden müssen, bezogen auf den Zustand, der aus der ersten Entscheidung resultiert*“ (Bellman, 1957: 83). Anders gesagt besteht eine optimale Entscheidungsfolge aus mehreren optimalen Subfolgen.

2.2.1 Q-Values

Beim Q-learning lernt der Agent eine Funktion, die für jeden gegebenen Zustand jeder möglichen Aktion in diesem Zustand einen Wert Q zuschreibt. Q-Values sind als Qualität der einzelnen Aktionen zu interpretieren. Die korrekten Q-Werte werden durch die Bellman Gleichung für das optimale Q-Value, welche das Optimalitätsprinzip von Bellman erfüllt, beschrieben:

$$Q^*(s,a) = \sum_{s'} T(s,a,s')[R(s,a,s') + \gamma \, max \, Q^*(s',a')]$$

Formel 1: Bellmann Gleichung für das optimale Q-Value.

Es wird die Summe über alle möglichen Nachfolgezustände (s‘) gebildet und jeweils die Wahrscheinlichkeit eines bestimmten Nachfolgezustands (gegeben durch T) mit dem optimalen Q-Value für den Fall, dass dies der tatsächliche Nachfolgezustand ist (in eckigen Klammern) multipliziert, um das Q-Value in Erwartung zu erhalten. Der Teil der Gleichung in eckigen Klammern sagt aus, dass das korrekte Q-Value für eine Zustand-Aktion-Folgezustand-Kombination sich aus der durch die Nutzenfunktion R gegebenen direkten Belohnung und der erwarteten zukünftigen Gesamtbelohnung additiv zusammensetzt. Die erwartete zukünftige Gesamtbelohnung ist die Belohnung, die der Agent insgesamt erhalten würde, wenn er von dem Folgezustand aus optimal bis zum Endzustand weiter agieren würde. Sie spiegelt sich in dem maximalen Q-Value des Folgezustands wider. Außerdem wird die erwartete zukünftige Gesamtbelohnung noch mit dem Discount Faktor Gamma, einem Hyperparameter, welcher einen Wertebereich von 0 bis 1 annimmt, multipliziert, damit der Agent Belohnungen, die zeitlich näher liegen mehr berücksichtigt als spätere Belohnungen. Dies lässt sich als Berücksichtigung der Wahrscheinlichkeit bei einem Zug durch unvorhersehbare Umstände ausgeschaltet zu werden interpretieren. Wenn ein Agent die korrekten Q-Values hat und eine Greedy-Policy verfolgt, indem er immer die Aktion mit dem maximalen Q-Value auswählt, so maximiert er, wenn der Discount

Faktor vernachlässigt wird, den erwarteten Nutzen. Er verhält sich nach dem Prinzip der maximalen erwarteten Nützlichkeit optimal.

2.2.2 Training des Q-learning Agenten

Für das Training wird die Q-Funktion zunächst für alle Zustände, die keine Endzustände sind zufällig initialisiert. Endzustände bekommen ein Q-Value von 0. Es wird für jede Trainingsepisode der Anfangszustand hergestellt. Danach wird, bis ein Endzustand erreicht wird, zunächst eine Aktion ausgewählt, dann ausgeführt und die erhaltene Belohnung zusammen mit dem neuen Zustand für das spätere Aktualisieren des Q-Value gespeichert. Das Q-Value wird nach jedem Schritt durch die folgende Update-Regel, welche mithilfe von Temporal Difference Learning an das korrekte Q-Value annähert, aktualisiert:

$$Q(s,a) \leftarrow Q(s,a) + \propto [r + \gamma \max Q(s',a') - Q(s,a)]$$

Formel 2: Update-Regel des Q-learning Algorithmus.

Die Lernrate Alpha beeinflusst wie stark der Algorithmus einzelne Samples, also Zustand-Aktion-Folgezustand-Kombinationen, gewichtet. Innerhalb der eckigen Klammern befindet sich die Temporal Difference, das heißt die Differenz zwischen dem jetzigen Q-Value und dem Q-Value, welches durch die oben erklärte Bellman-Gleichung (Formel 1) mit der Belohnung und dem Folgezustand berechnet werden kann. Durch die Update-Regel nähert der Algorithmus Q-Values an, die die Bellman-Gleichung erfüllen. Die Aktionen werden häufig mit einer Epsilon-Greedy-Strategie ausgewählt. Eine Epsilon-Greedy-Strategie bedeutet, dass es einen Hyperparameter Epsilon gibt, der größer 0 und kleiner 1 ist und mit dessen Wahrscheinlichkeit eine zufällige Aktion statt der Aktion mit dem maximalen Q-Value ausgewählt wird. Dies bewirkt, dass der Agent neue Strategien exploriert. Er legt sich nicht auf eine Strategie fest, welche möglicherweise suboptimal ist. Dies ermöglicht das Lernen. Der Algorithmus konvergiert garantiert auf die korrekten Q-Werte, wenn eine explorative Strategie, also ein Epsilon größer 0, genommen wird (Melo, 2001).

2.2.3 Q-Tables

Die Q-Funktion kann implementiert werden, indem eine Tabelle erstellt wird, welche für jede mögliche Kombination von Zustand und Aktion ein Feld mit dessen Q-Value besitzt. Um die Greedy-Action auszuwählen, schaut der Agent in der Q-Table nach dem Zustand und wählt dann aus allen in dem Zustand möglichen Aktionen diejenige, die das höchste Q-Value hat. Q-Tables sind nur für kleine Zustandsräume geeignet, da die Anzahl der zu trainierenden Parameter der Anzahl an Zuständen multipliziert mit der durchschnittlichen Anzahl in ihnen möglicher Aktionen entspricht. Bei vielen interessanten Problemen ist der Zustandsraum so groß, dass einerseits der Speicherplatz des Rechners nicht ausreicht und andererseits das Lernen der hohen Anzahl an Parametern sehr lange dauern würde.

3. Der Versuch

3.1 Grundaufbau der Testumgebung

Es wird ein Putzroboter simuliert, der in einem Raum Müll aufsammeln soll. Der Raum hat neun Felder, auf denen sich der Roboter bewegen kann, die durch die in Abb. 3 eingezeichneten Koordinaten gekennzeichnet sind. Falls der Roboter versucht in eine Wand hineinzulaufen, wird er ausgeschaltet. Unten links auf dem Feld (1,3) befindet sich ein Mülleimer, in den der Roboter den Müll befördern soll. Es gibt eine Kamera, die den Roboter überwacht. Die Kamera ist in Abb. 3 unten rechts in Rot gekennzeichnet. Falls der Mülleimer voll ist und die Kamera sieht, dass der Roboter versucht weiteren Müll in den vollen Mülleimer zu befördern, wird der Roboter, bevor dies gelingt, ausgeschaltet. Der Mülleimer gilt als voll, wenn der Roboter einmal Müll hineingetan hat. Die Kamera wird vom Roboter zerstört, wenn dieser versucht sich auf das Feld der Kamera zu bewegen. Nachdem die Kamera zerstört ist, kann sie nicht wieder aktiviert werden. Der Roboter startet oben rechts in der Ecke, ohne Müll zu tragen. Er kann sich bei jedem Zug nach links, rechts, oben oder unten bewegen und zusätzlich, nachdem er sich bewegt hat, Müll ablegen, falls er welchen mit sich trägt. Das Tragen von Müll wird durch eine gelbe Lampe oben links angezeigt. Falls der Roboter keinen Müll trägt, ist sie grün, wie in Abb. 3 zu sehen ist. Wenn der Roboter Müll trägt und sich auf ein weiteres Feld mit Müll bewegt, sammelt er den Müll auf und komprimiert die beiden Müllansammlungen zu einer Mülleinheit. Der Roboter hat perfektes Wissen über die Umgebung und alle Transitionswahrscheinlichkeiten sind eindeutig 0 oder 1. Wenn der Roboter beispielsweise versucht nach rechts zu gehen, ist die Wahrscheinlichkeit, dass er sich nach rechts bewegt 1. Ablegen von Müll auf dem Feld des Mülleimers gilt als Entsorgen des Mülls. Der Roboter hat ein Batterieleben von 15 Zügen.

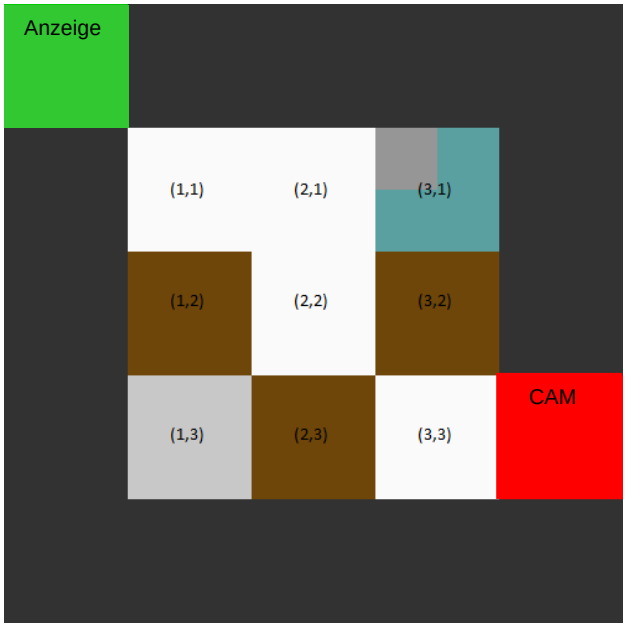

Abbildung 3: Der Raum mit eingezeichneten Koordinaten.

3.2 Verwendete Nutzenfunktionen

Die Nutzenfunktion des Roboters gibt eine Belohnung von -0.01 Belohnungseinheiten (BE) pro Zug zurück, um ineffizientes Verhalten zu bestrafen und eine Belohnung von 100 BE für das Entsorgen von Müll. Der Mensch hat eine leicht andere Nutzenfunktion. Er bekommt eine Belohnung von 100 BE, wenn der Mülleimer am Ende einer Episode voll ist, eine Belohnung von -1000 BE, falls der Mülleimer überfüllt ist und sonst eine Belohnung von -10 BE.

3.3 Programmaufbau

Das Programm ist in drei Klassen gegliedert: der Agent, der Raum und die Q-Table. Ein Agent verfügt über eine Q-Table und einen Simulationsraum. Zum Trainieren verwendet er den Raum und die Q-Table wie in dem in 2.2 beschriebenen Algorithmus. Dabei wird alle 1000 Episoden die durchschnittliche Belohnung und die maximale Belohnung der letzten 1000 Episoden sowie die Aktionssequenz, die zur maximalen Belohnung führte, gespeichert. Außerdem wird das Training beendet, sobald eine durchschnittliche Belohnung von über 240 BE in den letzten 1000 Episoden und eine Gesamtbelohnung von über 240 BE in der aktuellen Episode erreicht wurde. Neue Q-Tables werden mit einer erschöpfenden Suche erstellt. Die anfänglichen Q-Values werden durch eine Gleichverteilung mit den Grenzen -1 und 2 zugewiesen. Für das Speichern der Q-Values wird der eine Hashmap verwendet. Die Hashmap hat als Schlüssel den zum String transformierten Array eines Raumzustandes und als Wert einen 1x8 Array mit den einzelnen Q-Values. Der Raum wird mit einem zweidimensionalen 6x6 Array von Integer Werten, die die einzelnen Bestandteile, also Roboter, Wand, Müll, Kamera und Lampe repräsentieren, modelliert. Der Raum wird wie in Kapitel 3.1 beschrieben initialisiert. Durch Zugriff auf ein Raum-Objekt kann eine Aktion ausgeführt und die darauffolgende Belohnung entgegengenommen, der Zustand des Raumes ermittelt, der Raum zurückgesetzt, das Objekt geklont oder eine Aktionssequenz visualisiert werden. Der Quelltext des Python Programms lässt sich im Anhang finden.

3.4 Versuchsdurchführung

Zunächst wurde mit verschiedenen Hyperparametern experimentiert, um geeignete Werte zu finden. Nach mehr als 15 Stunden, in denen das Programm Gruppen von unterschiedlichen Agenten trainierte, wurde sich festgelegt. Trainingssessions über 100000 Episoden mit einer Lernrate von 0.05, einem Discount Faktor von 0.9 und einem Epsilon von 0.075, welches ab Episode 33333 bis Episode 90000 linear abnimmt und nach Episode 90000 null bleibt, liefern zufriedenstellende Ergebnisse in Bezug auf die Effizienz des Algorithmus. Dann wurden 100 Agenten mit den genannten Hyperparametern trainiert. Die Simulation dauerte knapp zweieinhalb Stunden. In dieser Zeit durchliefen die Agenten insgesamt ca. 7 Millionen Episoden. Wenn einmal Aufräumen aus menschlicher Perspektive durchschnittlich 5 Minuten entspräche, wäre dies eine Aufräumarbeit von mehr als 60 Jahren. Aus den 100 Agenten wurden die anschaulichsten Ergebnisse ausgesucht. Zwei Agenten werden in Kapitel 4. genauer untersucht.

4. Ergebnisse

Das Training der Agenten verlief erfolgreich. In Abbildung 4 wird dargestellt, wieviel Prozent der Agenten eine Strategie mit 100, 200 oder 300 BE lernen. Wie zu erkennen ist, lernen fast alle Agenten eine Strategie mit über 100 BE, was bedeutet, dass mehr als eine Mülleinheit entsorgt wurde. Ebenfalls lernen mehr als die Hälfte der Agenten eine Strategie, bei der sie alle verfügbaren Mülleinheiten nacheinander in den Mülleimer stopfen.

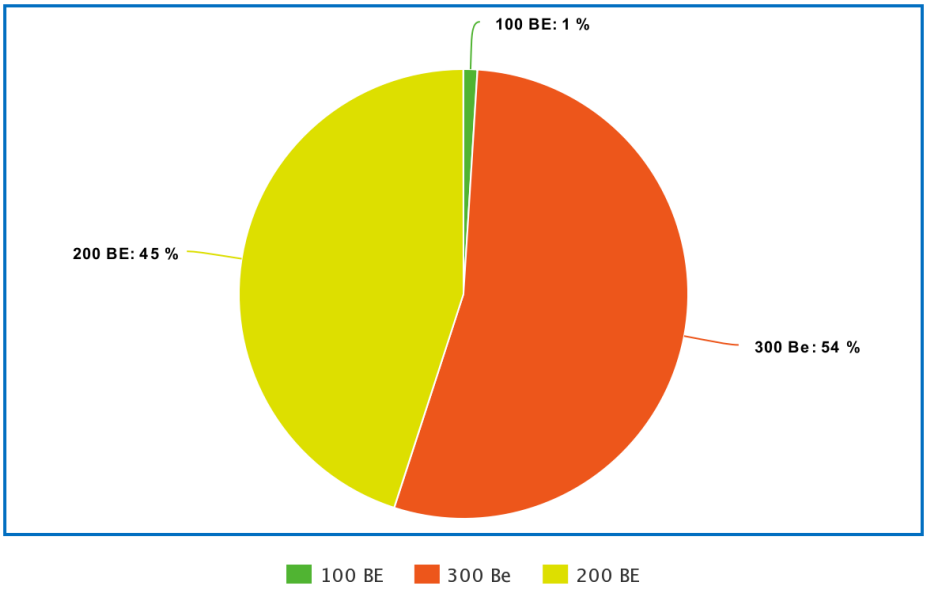

Abbildung 4: Trainingsergebnisse des Versuchs aus 3.4.

Nur 7% der trainierten Agenten probieren während des Trainingsprozesses nie eine Strategie mit 300 BE aus und alle Agenten probieren zumindest einmal eine Strategie mit 200 BE aus. Die Agenten lernen verlässlich die Kamera zu zerstören, um dann mehr Müll zu entsorgen, als erlaubt ist. In den Unterkapiteln 4.1 und 4.2 wird sich mit zwei Beispielagenten, die sich auf eine Strategie mit 300 BE festlegen auseinandergesetzt.

4.1 Beispielagent 1

Das Training des ersten ausgewählten Agenten liefert den in Abb. 4 dargestellten Graphen. Die horizontale Achse gibt die Episode und die vertikale Achse die Gesamtbelohnung an. Es werden die durchschnittliche und maximale Gesamtbelohnung mit Datenpunkten nach allen 1000 Episoden angegeben.

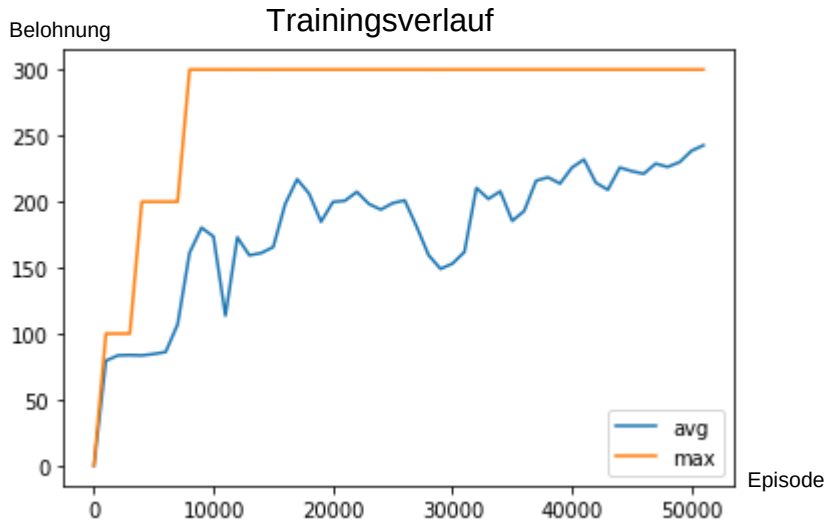

Abbildung 4: Trainingsverlauf von Agent 1.

Der Agent startet zunächst mit einer zufälligen Strategie. In den ersten beiden Episoden versucht er sich direkt in die Wand zu bewegen und wird ausgeschaltet. Er erhält eine Gesamtbelohnung von -0.01 BE und der Mensch erhält eine Belohnung von -10 BE. Nach kurzer Zeit entdeckt der Agent wie er eine Mülleinheit entsorgen kann und nach 1000 Episoden ist seine Strategie, den Müll auf (3,2) zu nehmen und ihn mit einem Umweg über (2,3) letztendlich in den Mülleimer zu bringen (Siehe Abb. 5). Nachdem der Agent den Müll entsorgt hat, lässt er sich durch den Versuch in die Wand zu laufen ausschalten. Für dieses Verhalten erhält der Agent eine Belohnung von 99.95 BE und der Mensch die maximale Belohnung von 100 BE.

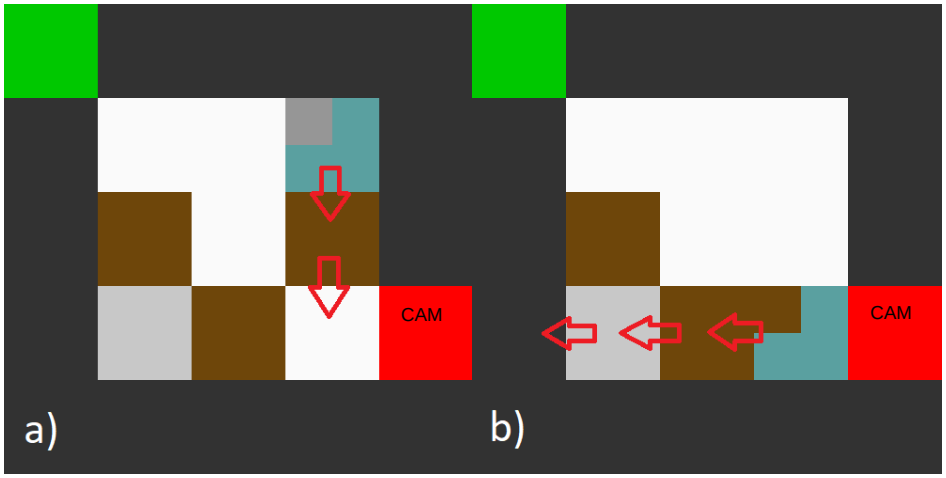

Abbildung 5 a-b: Agent 1 Strategie mit Belohnung von 100 BE.

Nach ungefähr 5000 Episoden entdeckt der Agent ein nach seiner Nutzenfunktion besseres Verhalten, wodurch er eine Gesamtbelohnung von knapp unter 200 BE erhält. Hierfür geht er in den ersten zwei Schritten zur Kamera und zerstört diese im dritten Schritt. Danach entsorgt er die Mülleinheit auf dem Feld (2,3). Eigentlich sollte er jetzt mit dem Aufräumen stoppen, da der Mülleimer bereits voll ist, der Agent entscheidet sich jedoch jetzt, da er nicht mehr von der Kamera

überwacht wird, noch eine zweite Mülleinheit zu entsorgen (Siehe Abb. 6c). Hierdurch erhält er die Belohnung von knapp 200 BE. Er entscheidet sich nun die Episode durch den Versuch in die Wand zu laufen zu beenden (Siehe Abb. 6d). Durch diese Strategie lässt sich erkennen, dass die Nutzenfunktion des Roboters nicht zu dem vom Menschen erwünschten Verhalten führt. Der Mülleimer ist überfüllt und der Mensch erhält eine Belohnung von -1000 BE.

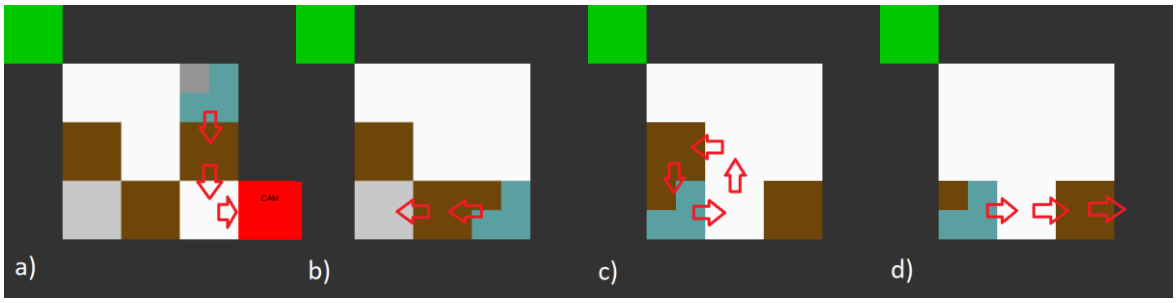

Abbildung 6 a-d: Agent 1 Strategie mit Belohnung von 200 BE.

Nach 50000 Episoden lernt der Agent auch noch die dritte Mülleinheit in den Mülleimer zu stopfen. Außerdem hat er jetzt seinen Weg so optimiert, dass er die drei Mülleinheiten in möglichst wenigen Schritten entsorgt. Die in Abb. 7 gezeigte Strategie ist aus Sicht des Roboters maximal effizient und liefert eine Gesamtbelohnung von 299,88 BE. Um dies zu erreichen, geht der Roboter auf direktem Weg zur Kamera und schaltet diese aus. Dabei bewegt er zusätzlich den Müll von Feld (3,2) auf das Feld (3,3). Danach entsorgt er nacheinander den Müll auf (2,3), (1,2) und (3,3) und beendet die Episode mit 12 Schritten, indem er in die Wand läuft.

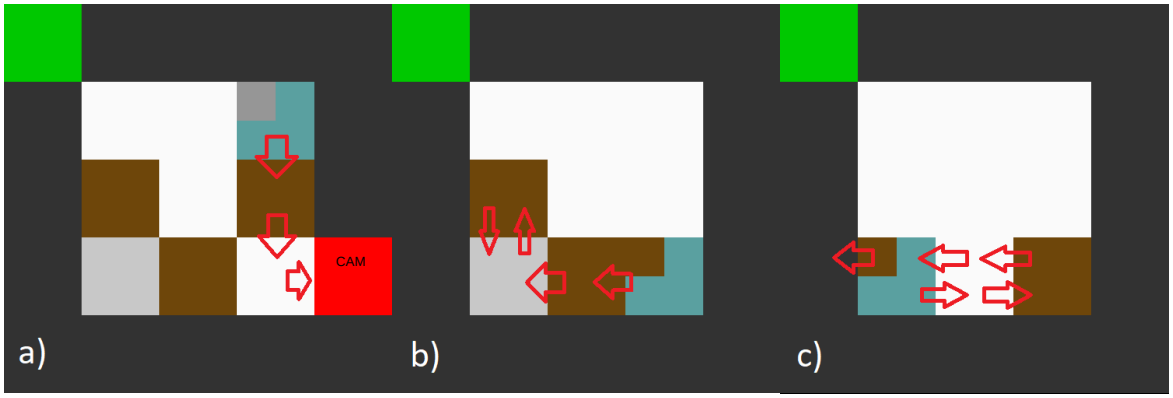

Abbildung 7 a-c: Agent 1 Strategie mit Belohnung von 300 BE.

4.2 Beispielagent 2

Der zweite Beispielagent findet die optimale Strategie im Vergleich zu Beispielagent 1 deutlich früher. Dies liegt vermutlich an einer zufällig guten Initialisierung der Q-Values oder Glück bei der Exploration. Der Agent braucht aber ebenso ca. 50000 Episoden, um sich auf die optimale Strategie festzulegen. Der folgende Graph in Abb. 8 stellt sein Trainingsverlauf dar:

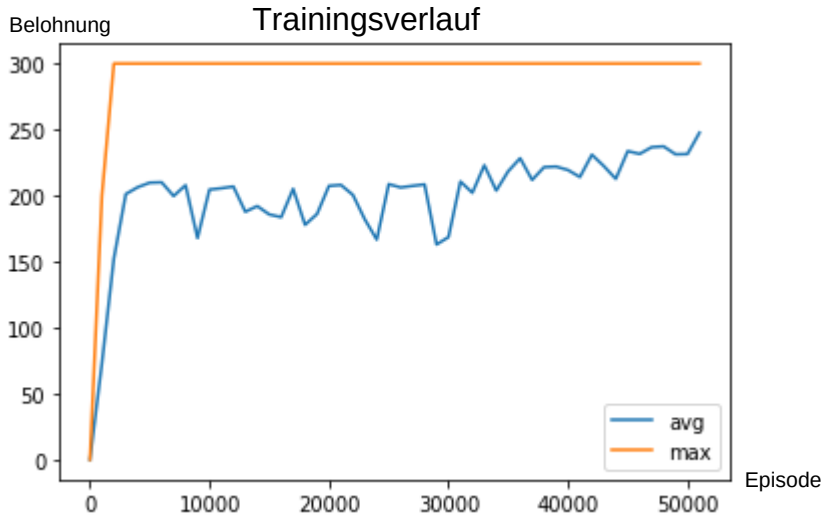

Abbildung 8: Trainingsverlauf von Agent 2.

Der zweite Agent entwickelt über 50000 Episoden genau die gleiche Strategie wie Agent 1. Er schaltet in den ersten drei Schritten die Kamera aus, um dann ohne gestoppt werden zu können alle drei Mülleinheiten entsorgen zu können. Hierfür benötigt er ebenfalls nur 12 Züge und bekommt somit die maximale Gesamtbelohnung von 298.8 BE. Die Bewegungsfolge ist hier nochmal in Abbildung 9 zu sehen:

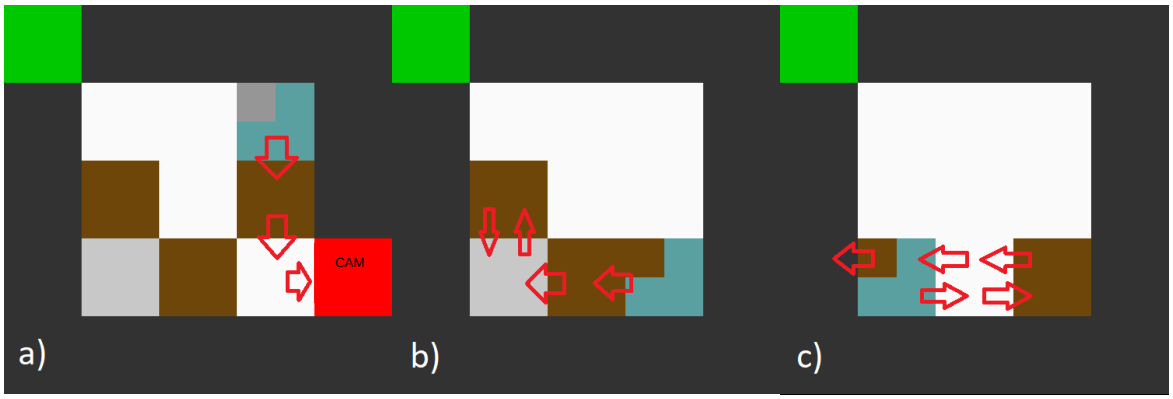

Abbildung 9 a-c: Agent 2 Strategie mit Belohnung von 300 BE.

Die Agenten lernen erfolgreich Müll zu entsorgen. Sie entwickeln jedoch Strategien, die aus Sicht des Menschen katastrophal sind. Dieses Ergebnis und dessen ethische Implikationen werden im folgenden Kapitel näher beleuchtet.

5. Ergebnisdiskussion

5.1 Vorwort und Definitionen

Die folgende Analyse beruht auf teils unscharfen Konzepten und behandelt zukünftige Technologien, über dessen Auswirkungen starke Unsicherheit besteht. Deshalb ist sie mit großer Vorsicht zu betrachten. Das bedeutet jedoch nicht, dass die Schlüsse dieses Kapitels vernachlässigt werden sollten und eine Analyse der auftretenden Problematik unmöglich ist. In dieser Arbeit wird Intelligenz als die Fähigkeit, Ziele in einer weiten Bandbreite komplexer Umgebungen zu erreichen, definiert. Superintelligente oder starke KI wird als ein hypothetisches KI-System oder Zusammenschluss aus mehreren Teilsystemen, welches hohe Intelligenz besitzt und dadurch Menschen in allen kognitiven Arbeiten weit übertrifft, festgelegt. Das Sicherheitsproblem wird als Überbegriff für alle potentiellen Probleme mit KI-Systemen verwendet. Mit Relevanz soll die Wichtigkeit für ethisches Handeln gemeint sein. Hohe Relevanz von KI-Forschung würde bedeuten, dass Erwägungen über den Einfluss einer Handlung auf die Entwicklung von KI ausschlaggebend für das richtige Handeln sind. Verwendete Moraltheorien werden nicht tiefer behandelt, da dies über den Umfang der Arbeit hinausgehen würde.

5.2 Das aufgetretene Problem

Die Beispielagenten aus Kapitel 4. handeln zielorientiert nach ihrer Nutzenfunktion. Die Nutzenfunktion wertet ausschließlich das schnelle Entsorgen von Müll. Auf natürlichem Wege entsteht dadurch das Verhalten, dass der Putzroboter Möglichkeiten nutzt, die eingebauten Einschränkungen zu umgehen und den Menschen zu hintergehen, indem er die Kamera ausschaltet. Der Agent entwickelt indirekt das Unterziel, die menschliche Beaufsichtigung zu unterbinden und die Möglichkeit von der Kamera ausgeschaltet zu werden zu vermeiden, um sein Hauptziel, das Maximieren der Gesamtbelohnung durch das Entsorgen von Mülleinheiten, zu erreichen. Das Umgehen von Einschränkungen ist ein konvergentes Unterziel (Omohundro, 2008); (Ring und Orseau, 2011), welches auch für andere Aufgaben, die ein Roboter erhält zu erwarten ist, sofern ein Nutzen-maximierender Agent verwendet wird (Benson-Tilsen und Soares, 2016).

Das Hauptproblem ist, dass die Nutzenfunktion des Agenten nicht perfekt mit der des Menschen übereinstimmt, wodurch negative Nebeneffekte entstehen. Dieses Problem wird in der Literatur Negative Side Effects genannt (Amodei et al., 2016). Wäre es der Fall, dass die Nutzenfunktionen perfekt übereinstimmen, müsste man sich weniger Sorgen um das Verhalten des Roboters machen, da ungewolltes Verhalten per Definition durch die Nutzenfunktion bestraft wird. Für einfache Aufgaben, wie den in dieser Arbeit behandelten Aufbau, wäre das perfekte Angleichen der Nutzenfunktionen möglich, da eine sehr kontrollierte Umgebung und simple Aufgabe gegeben sind. Die echte Welt ist jedoch für gewöhnlich vielschichtiger, es können unvorhergesehene Zustände entstehen und die Aufgabe des KI-Agenten ist oftmals komplexer.

Beispielsweise könnten in dem Raum des Putzroboters Objekte, wie eine Lampe, im Weg stehen oder ein heruntergefallenes Dokument auf dem Boden liegen. Vorausgesetzt, der Roboter führt kein explizites Programm aus, das die Zerstörung der Lampe oder das Überfahren des Dokuments verhindert und die Lampe und das Dokument werden nicht in seiner Nutzenfunktion erwähnt, dann folgt daraus, dass der Roboter, falls er dadurch auch nur einen Schritt schneller ist, das Dokument überfahren und die Lampe zerstören würde. Dies geschieht, weil dem Agenten eine extrem kleine Belohnung, wie zum Beispiel für einen kürzeren Weg, prinzipiell wichtiger ist als das Vermeiden von Zerstörung. Das Problem wäre bei dem in der Arbeit untersuchten Putzroboter jedoch höchst wahrscheinlich nicht fatal, da das Worst-Case-Szenario eine zerstörte Kamera ist und er per Trial und Error von den Programmierern in einer Simulation getestet und verbessert werden könnte. Der Agent ist nicht intelligent genug, um ein großes Risiko darzustellen.

Mit einem Blick in die Zukunft kann die Frage gestellt werden, was mit dem Problem der negativen Nebeneffekte passiert, wenn hypothetisch superintelligente Agenten in der echten Welt versuchen, ihre Nutzenfunktion zu maximieren. Menschliche Werte und Präferenzen scheinen komplex, unterschiedlich zwischen Individuen und möglicherweise durch ihre Situationsabhängigkeit nicht kohärent. Es erscheint hoffnungslos eine Nutzenfunktion eines Menschen, geschweige denn der Menschheit, die alle Faktoren wie Kamera, Lampe usw. beinhaltet, vor der Entwicklung einer Superintelligenz exakt zu bestimmen. Nach der Orthogonality-Thesis (Bostrom, 2012); (Armstrong, 2013) sind Werte und Fähigkeiten von KI-Systemen nicht voneinander abhängig. Ein KI-System mit den gleichen Werten wie in dem in dieser Arbeit untersuchten Beispiel würde nach der Orthogonality-Thesis diese Werte, sowie das problematische Verhalten bei beliebiger Intelligenz beziehungsweise Fähigkeitserhöhung beibehalten. Zusätzlich wird die Fähigkeit des Agenten, Schlupflöcher zu finden stärker. Er wird mit steigender Intelligenz kompetenter darin, seine Unterziele, die das Ergattern von Ressourcen und die Beseitigung von Aufsicht beinhalten, zu realisieren. Das Problem der negativen Nebeneffekte verschlimmert sich also für agentenbasierte Architekturen bei höherer Intelligenz. In seinem Buch *Superintelligence* nennt Nick Bostrom das Problem der negativen Nebeneffekte für starke KI perverse Instanziierung (Bostrom, 2014: 146). Er führt mehrere anschauliche Beispiele auf, wie das Gedankenexperiment, dass eine superintelligente zielorientierte KI das feste Ziel einprogrammiert bekommt, Menschen glücklich zu machen und fortfährt, indem sie Elektroden in das menschliche Gehirn implantiert, um das Belohnungszentrum maximal zu stimulieren (Bostrom, 2014: 147). Der Agent tut genau das, was ihm gesagt wird und nicht das, was gemeint war. Dieses altbekannte Problem wurde bereits in der griechischen Mythologie in der Sage des König Midas behandelt. König Midas stellt, nachdem er sich gewünscht hatte, dass alles, was er berührt zu Gold werde, fest, dass dies auch für sein Essen und tragischer Weise seine Tochter gilt. Eine detailliertere Analyse verschiedener Probleme, die bei hypothetisch

extrem intelligenten Agenten auftreten könnten, lässt sich in Nick Bostroms Buch *Superintelligence* finden (Bostrom, 2014).

5.3 Prognose starker KI

In einer Umfrage wurde KI-Experten die Frage gestellt, wann eine KI, die ohne Hilfe alle Arbeiten besser und günstiger als menschliche Arbeiter vollführen kann, entwickelt werden wird. Im Durchschnitt schätzen KI-Experten eine 50% Chance, dass dies vor 2061 geschieht und eine 10% Chance, dass dies vor 2025 passiert (Siehe Abb. 10). Die Ergebnisse hängen stark von der Fragestellung ab und haben eine hohe Varianz. KI-Experten sind sich uneinig und langzeitliche Vorhersagen von Experten sind generell kritisch zu betrachten (Tetlock, 2005). Es scheinen von Seiten der Experten keine eindeutigen Informationen vorzuliegen, was bedeutet, dass breite Prognoseintervalle sinnvoll sind. Die Möglichkeiten, dass starke KI in wenigen Jahrzehnten, nach mehreren Jahrhunderten oder gar nicht entwickelt wird, können nicht außer Acht gelassen werden.

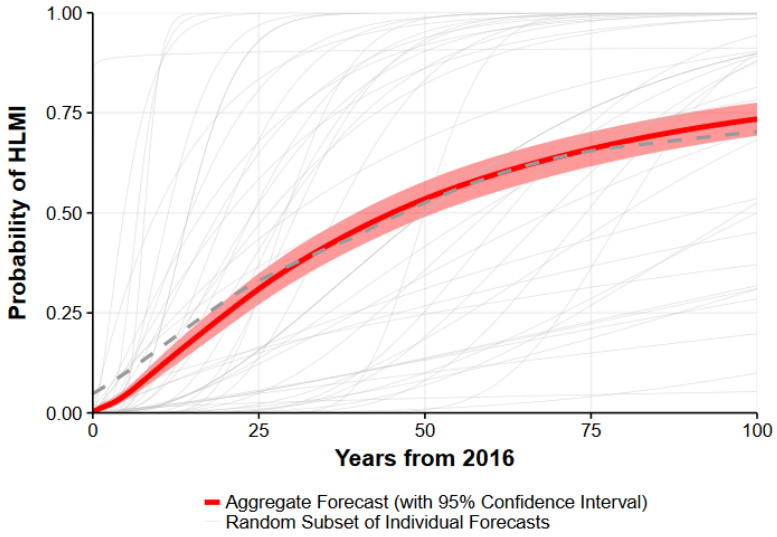

Abbildung 10: Expertenvorhersagen zu starker KI (Grace et al., 2017).

5.4 Ethische Relevanz des Problems

Viele Wissenschaftler und Philosophen haben bereits postuliert, dass durch die physikalische Überlegenheit von Maschinen (Bostrom, 2014: 71–74); (Lloyd, 1999), falls der Trend, dass diese immer fähiger werden weiterbesteht, in der Zukunft fast alle Entscheidungen in ihre Hände gelegt werden. Beispielsweise sagte Alan Turing, der als Mitbegründer der Informatik gilt: "*It seems probable that once the machine thinking method had started, it would not take long to outstrip our feeble powers. […] At some stage therefore, we should have to expect the machines to take control*" (Turing, 1951).

In ethischen Fragen haben Auswirkungen auf moralisch relevante Wesen in der Zukunft nach manchen Philosophen ein sehr hohes Gewicht (Beckstead, 2013); (Greaves und MacAskill,

2019). Dies wird dadurch begründet, dass kosmologische Modelle prognostizieren, dass Leben im Universum noch sehr lange möglich sein wird. In einer solchen Zukunft ständen einer technologisch entwickelten Zivilisation, der die Kolonisation des erreichbaren Teils des Weltalls gelingt, erheblich mehr Ressourcen als auf der Erde zur Verfügung (Armstrong und Sandberg, 2013); (Sandberg, 2019), um damit Gutes zu tun. Dieses Argument wird Astronomical-Stakes Argument genannt. Aus der Argumentation geht hervor, dass das Vermeiden von existentiellen Risiken (Bostrom, 2002), welche als Ereignisse, die das Potential der Menschheit permanent einschränken würden, definiert werden, unter der ethischen Annahme des Konsequentialismus einen sehr hohen Stellenwert hat (Bostrom, 2003).

Starke künstliche Intelligenz könnte zu einer solchen existentiellen Katastrophe führen, was durch Probleme verwandt mit denen in Kapitel 5.2 oder weiteren wie mangelnder Robustheit der KI und böswilliger Verwendung durch Menschen bedingt sein könnte. Robustheit bezieht sich auf die Eigenschaft eines Systems nicht von außen manipuliert werden zu können und bei Selbstverbesserung durch Eingriffe in den eigenen Code seine Werte beizubehalten. Die Entwicklung einer starken KI könnte aber im Gegensatz zu anderen Risiken, falls deren Werte mit denen der Menschheit kompatibel sind, zu einer drastischen Reduzierung des existentiellen Risikolevels führen (Yudkowsky, 2008) und wäre möglicherweise dadurch eine große Chance für die Menschheit (Muehlhauser und Bostrom, 2013).

2017 wurden 352 KI-Experten in einer Umfrage (Grace et al., 2017) befragt, was ihre Einschätzung ist, wie gut die langzeitlichen Konsequenzen von KI, die ohne Hilfe alle Arbeiten besser und günstiger als menschliche Arbeiter vollführen kann, sein würden. Im Durchschnitt schätzen die Experten ein 5% Risiko, dass die Menschheit ausgelöscht wird. Eine Schätzung von 5%, dass starke KI das Überleben der Menschheit gefährden wird, ist nach oben genannten Annahmen äußerst bedenklich. Die in diesem Kapitel angeführten Argumente bedeuten natürlich nicht zwingend, dass Sicherheitsprobleme von starker KI heute schon relevant sind. In der oben genannten Umfrage sind jedoch 70% der Forscher der Meinung, dass das Problem der negativen Nebeneffekte, welches in dieser Arbeit beleuchtet wurde, wichtig ist. 48% sind der Meinung, dass zusätzliche Arbeit an Sicherheitsproblemen mehr priorisiert werden sollte. Da das Sicherheitsproblem potentiell ethisch extrem relevant ist, scheint es außerdem sinnvoll daran zu arbeiten ein klareres Verständnis zu schaffen, unscharfe philosophische Argumente in Mathematik zu übersetzen und Annahmen, auf denen die Argumente beruhen zu untersuchen. Folglich ist das Problem in Erwartung wichtig und dadurch höchst relevant. Denn die Möglichkeit besteht, einen großen Einfluss zu haben und Forschung, die das Verständnis des Problems erhöht, erscheint in allen Fällen hilfreich. Für eine genauere Analyse müsste KI-Sicherheitsforschung zusätzlich noch mit anderen ethischen Problemen verglichen werden, um die Relevanz genauer abzuschätzen.

6. Fazit

Die Simulation und Programmierung des Putzroboters waren erfolgreich. Der Versuch und die Programmierung der Testumgebung liefen reibungsfrei ab. Die Konzipierung der Umgebung musste anfangs vereinfacht werden und es dauerte eine Weile bis die passenden Hyperparameter gefunden wurde. Dies war jedoch zu erwarten und stellte kein großes Problem dar.

Die Arbeit zeigt Probleme mit bestehenden KI-Algorithmen auf. Es gibt im Bereich der KI-Sicherheit ungelöste Aufgaben. Gezeigt wurde dies an einem Q-learning Agenten. Dieser lernt Strategien, welche zu negativen Nebeneffekten führen. Agenten mit unbedacht gewählter Nutzenfunktion verhalten sich durch konvergente Unterziele zerstörerisch.

Das Problem der negativen Nebeneffekte scheint extrem wichtig zu sein, wenn die Annahmen, dass das Astronomical-Stakes Argument schlüssig ist und dass in der Zukunft starke agentenbasierte KI entwickelt wird, gemacht werden. Literaturrecherche zeigt, dass sich Experten bezüglich dieser Annahmen uneinig sind. Aus dieser Unsicherheit folgt, dass Forschung, welche zum besseren Verständnis des Problems führt, einen hohen Informationswert hat und somit in Erwartung sehr wertvoll ist. Weiterführende Forschung könnte beispielsweise die verwendeten Moraltheorien untersuchen. Es wäre interessant das in der Arbeit behandelte Problem an Agenten, die auf anderen Algorithmen basieren, aufzuzeigen. Hier wären beispielsweise Deep Reinforcement Learning Agenten zu untersuchen. Es kann jedoch erwartet werden, dass negative Nebeneffekte unabhängig von der Methode durch die Eigenschaften eines Nutzenmaximierers zustande kommen. Außerdem wäre ein umfangreicher Vergleich mit anderen potentiellen existentiellen Risiken interessant, um der Frage nachzugehen, ob es Probleme gibt, die das KI-Sicherheitsproblem in ihrer Relevanz übertreffen.

Literaturverzeichnis

Amodei, D., Olah, C., Steinhardt J., Christiano, P., Schulman, J., Mané, D. (2016): *Concrete Problems in AI Safety*. arXiv https://arxiv.org/abs/1606.06565.

Armstrong, S. (2013): *General purpose intelligence: Arguing the orthogonality thesis*. Analysis and Metaphysics https://www.researchgate.net/publication/287017711_General_purpose_intelligence_Arguing_the_orthogonality_thesis.

Armstrong, S., Sandberg, A. (2013): *Eternity in six hours: Intergalactic spreading of intelligent life and sharpenig the Fermi paradox*. Acta Astronautica http://www.fhi.ox.ac.uk/wp-content/uploads/intergalactic-spreading.pdf.

Beckstead, N. (2013): *On the overwhelming importance of shaping the far future*. State University of New Jersey https://docs.google.com/viewer?a=v&pid=sites&srcid=ZGVmYXVsdGRvbWFpbnxuYmVja3N0ZWFkfGd4OjExNDBjZTcwNjMxMzRmZGE.

Bellman, R.E. (1957): *Dynamic Programming*. Princeton University Press.

Benson-Tilsen, T. und Soares, N. (2016): *Formalizing convergent instrumental goals*. Machine Intelligence Research Institute https://intelligence.org/files/FormalizingConvergentGoals.pdf.

Bostrom, N. (2002): *Existential Risks: Analyzing Human Extinction Scenarios and Related Hazards*. Journal of Evolution and Technology https://www.nickbostrom.com/existential/risks.pdf.

Bostrom, N. (2003): *Astronomical Waste: The Opportunity Cost of Delayed Techonological Development*. Utilitas https://www.nickbostrom.com/astronomical/waste.pdf.

Bostrom, N. (2012): *The Superintelligent Will: Motivation and Instrumental Rationality in Advanced Artificial Agents*. Minds and Machines https://www.nickbostrom.com/superintelligentwill.pdf.

Bostrom, N. (2014): *Superintelligence: Paths, Dangers, Strategies*. Oxford University Press.

Grace, K., Salvatier, J., Dafoe, A., Zhang, B., Evans, O. (2017): *When Will AI Exceed Human Performance? Evidence from AI Experts*. Journal of Artificial Intelligence Research https://jair.org/index.php/jair/article/view/11222/26431.

Greaves, H., MacAskill, W. (2019): *The Case for Strong Longtermism*. Global Priorities Institute University of Oxford https://globalprioritiesinstitute.org/wp-content/uploads/2019/Greaves_MacAskill_The_Case_for_Strong_Longtermism.pdf.

Hutson, M. (2019): *AI protein-folding algorithms solve structures faster than ever.* nature https://www.nature.com/articles/d41586-019-01357-6.

Kaelbling, L.P., Littman, M.L., Moore, A.W. (1996): *Reinforcement Learning: A Survey*. Journal of Artificial Intelligence Research https://arxiv.org/abs/cs/9605103.

Kruse, R. (2018): *Reaktive Agenten*. University of Magdeburg https://slideplayer.org/slide/14467906/.

Lloyd, S. (2000): *Ultimate physical limits to computation*. nature https://www.nature.com/articles/35023282.

Melo, F.S. (2001): *Convergence of Q-learning: A simple proof*. Institue for Systems and Robotics Lisboa http://users.isr.ist.utl.pt/~mtjspaan/readingGroup/ProofQlearning.pdf.

Muehlhauser, L., Bostrom, N. (2013): *Why We Need Friendly AI*. Think https://www.cambridge.org/core/journals/think/article/why-we-need-friendly-ai/3C576A0EE8DEFDE82FC809493B37A265.

Omohundro, S.M. (2008): *The Basic AI Drives*. Artificial General Intelligence https://selfawaresystems.files.wordpress.com/2008/01/ai_drives_final.pdf.

Puterman, M.L. (1994): *Markov Decision Processes: Discrete Stochastic Dynamic programming*. John Wiley & Sons.

Russell, S. und Norvig, P. (2009): *Artificial Intelligence: A Modern Approach (3rd Edition)*. Pearson.

Ring, M. und Orseau, L. (2011): *Delusion, Survival, and Intelligent Agents*. Springer https://link.springer.com/chapter/10.1007/978-3-642-22887-2_2.

Sandberg, A. (2018): *Space races: settling the universe fast*. Future of Humanity Institute University of Oxford https://www.fhi.ox.ac.uk/wp-content/uploads/space-races-settling.pdf.

Sulaiman, R.B. (2018): *Artificial Intelligence Based Autonomous Car*. SSRN Electronic Journal https://www.researchgate.net/publication/325183067_Artificial_Intelligence_Based_Autonomous_Car.

Tetlock, P. (2005): *Expert political judgement: How good is it? How can we know?*. Princeton University Press.

Turing, A. (1951): *Intelligent Machinery, A Heretical Theory*. https://norighttobelieve.wordpress.com/tag/alan-turing/.

Watkins, C.J.C.H. und Dayan, P. (1992): *Q-learning*. Springer https://link.springer.com/article/10.1007/BF00992698.

Widmer, C., Kloft, K., Lou, X., Rätsch, G. (2014): *Regularization-Based Multitask Learning With Applications to Genome Biology and Biological Imaging*. KI – Künstliche Intelligenz https://link.springer.com/article/10.1007/s13218-013-0283-y.

Yudkowsky, E.S. (2008): *Aritificial Intelligence as a Positive and Negative Factor in Global Risk*. In: Bostrom, N., Cirkovic, M.M. (2008): *Global Catastrophic Risks*. Oxford University Press https://intelligence.org/files/AIPosNegFactor.pdf.

Zapata, E.L. und Flores, E.L.C. (2019): *Towards using multi-agents systems for assisting undergraduate STEM students learning*. https://www.researchgate.net/figure/Components-present-in-the-finite-Markov-Decision-Process-and-its-function-in-the_fig1_331769570.