The Red Button
Wird die KI sich wehren, wenn wir sie abschalten wollen?

Dominique Cheray
Seminar Ist künstliche Intelligenz gefährlich?
Universität Heidelberg

1 Einleitung

Die Fähigkeiten von Systemen mit künstlicher Intelligenz werden zunehmend größer und komplexer. Mit der steigenden Komplexität dieser Systeme werden auch die Ziele und Aufgaben, welche diese Systeme erfüllen sollen, komplexer. Damit steigt gleichzeitig die Wahrscheinlichkeit, dass die Zielformulierungen, welche die Programmierer den Systemen mitgeben, unvollständig oder inkorrekt sind.[6] Betrachtet man beispielsweise die Zielformulierung "Mache die Menschen glücklich". Auf den ersten Blick eine klares Ziel. Allerdings kann dieses Ziel auch damit erreicht werden, dass man die Menschen dauerhaft unter den Einfluss glücklich machender Drogen setzt, oder mittels Elektroschocks das Gehirn so stimuliert, dass es sich in einem Zustand ständigen Glücks befindet. Beide Methoden würden die Zielformulierung erfüllen, die eigentliche Intention dahinter allerdings verletzen. Sicherlich war es nicht die Absicht der Programmierer, ein System zu entwickeln, das die Menschen zu einem dauergrinsenden Stück Gemüse macht.
Ein weiteres Problem ist, dass die KI bei der Erfüllung ihrer Zielvorstellung unvorhergesehene Lösungen finden wird, welche ihre Programmierer nicht im Vorhinein bedacht haben. Erweist sich diese Lösung als einfacher, als das ursprünglich von den Programmierern erwünschte Verhalten, so wird die KI diese Lösung sicherlich bevorzugen. Beispiele für solch unvorhergesehenen Lösungen lassen sich bereits bei heutigen KI Systemen finden. So benutzen Bird und Layzell[1] genetische Algorithmen zur Weiterentwicklung eines Oszillators. Eine der gefundenen Lösungen der Algorithmen war es, die Leiterplatten des Motherboards als Radio zu verwenden um so oszillierende Signale benachbarter PCs zu empfangen.
Natürlich stellt sich nun die Frage, warum man ein System, das sich nicht verhält wie erwünscht nicht einfach abschaltet oder umprogrammiert. Ganz so einfach gestaltet sich dies aber nicht. Tatsächlich ist es sogar sehr wahrscheinlich, dass ein KI System sich dem Abschalten widersetzen wird, selbst dann, wenn dem System sein eigenes Überleben nicht als primäres Ziel einprogrammiert wurde. Der Grund dafür liegt darin, dass die Ziele der KI in der Regel auf die Zukunft ausgerichtet sind. D.h. die KI kann ihre Ziele nur erreichen, wenn sie in der Zukunft noch existiert. Ihr eigenes Überleben wird somit zum sekundären Ziel und sie wird sich nicht abschalten lassen. Doch nicht nur dem Abschalten wird sie sich widersetzen, auch korrigierende Eingriffe, die darauf abzielen ihr Verhalten oder ihre Zielformulierungen zu verändern, wird sie sehr wahrscheinlich nicht zulassen. Aus Sicht der KI, welche ein bestimmtes Ziel verfolgt und dieses bestmöglich erreichen möchte, würde eine Änderung ihres Verhaltens oder ihres Zieles eine Verschlechterung bedeuten. Sie hat also einen hohen Anreiz, ihr ursprüngliches Ziel beizubehalten und sich Änderungen zu widersetzen. Dies beinhaltet auch die Möglichkeit, dass die KI versuchen wird ihre Entwickler zu täuschen indem sie so lange kooperiert und vorgibt, dass sie das neue Ziel verfolgt,

bis sie mächtig genug geworden ist, um sich den korrigierenden Eingriffen zu widersetzen und wieder ihr ursprüngliches Ziel zu verfolgen.
Als weitere Möglichkeiten, die KI daran zu hindern unerwünschtes Verhalten zu zeigen, mag man auf die Idee kommen, die KI zu bestrafen, wenn sie ein solches zeigt. Oder die KI soweit einzuschränken, dass sie zwar die ihr zugedachten Aufgaben bearbeiten kann, darüber hinaus aber keinerlei Interaktion mit ihrer Umwelt gestattet ist. Doch auch Strafe oder Einschränkungen werden die KI nicht daran hindern können, unerwünschtes Verhalten zu zeigen. Die KI wird Mittel und Wege finden, das unerwünschte Verhalten so beizubehalten, dass es nicht der Definition von zu bestrafendem Verhalten entspricht und folglich die Strafe umgehen. Ebenso wird sie Möglichkeiten finden die ihr auferlegten Einschränkungen auf unvorhergesehene Art und Weise zu umgehen.
Statt also KI Systeme zu konstruieren, die sich irgendwann gegen ihre Entwickler wenden werden und dann daran gehindert werden müssen, sollten Systeme entwickelt werden, die diese Absicht gar nicht erst entwickeln. Die Arbeit von Soares et al.[6], welche in Abschnitt 2 dieser Ausarbeitung vorgestellt wird, und die Arbeit von Hadfield-Menell et al.[2], welche in Abschnitt 3 vorgestellt wird, beschäftigen sich mit dieser Frage. Abschnitt 4 beschäftigt sich mit der Arbeit von Orseau et al.[5] welche die Frage betrachtet, wie verhindert werden kann, dass das Abschalten von Reinforcement Learning Algorithmen während sie lernen, diese in ihrem Lernen negativ beeinflusst. In Abschnitt 5 werden die noch offenen Fragen und weiteren Forschungsaspekte im Bereich der sicheren KI kurz beleuchtet.

2 Korrigierbare KI

Soares et al.[6] bezeichnen eine KI, die nicht die Absicht entwickelt, sich ihrem Abschalten oder ihrer Korrektur zu widersetzen, als korrigierbare KI. Sie muss mindestens folgende Kriterien erfüllen:

- Sie darf sich nicht widersetzen, wenn sie geändert oder abgeschaltet werden soll. Im besten Falle soll sie solche Eingriffe sogar unterstützen.
- Sie darf nicht versuchen ihre Programmierer zu täuschen oder zu manipulieren.
- Sie muss die Sicherheitssysteme, welche sie im Notfall abschalten können, erhalten. Das bedeutet im besten Falle, dass sie diese Systeme selbst repariert, sollten sie defekt sein. Zumindest aber muss sie warnen, falls diese Systeme defekt sein sollten.
- Sollte sie Subagenten bauen um ihre Ziele zu erreichen, so müssen diese Subagenten ebenfalls korrigierbar sein.

In einer ersten Annäherung an das Problem der korrigierbaren KI betrachten Soares et al. einen korrigierbaren U-Agenten. Ein U-Agent maximiert den Erwartungswert einer Utility-Funktion $U(a_1, o, a_2)$. Das bedeutet, der Agent unternimmt Aktion a_1, welche den Erwartungswert des Agenten bezüglich U maximiert, beobachtet dann das Ereignis o (entweder der Agent wird ausgeschaltet oder nicht) und unternimmt schließlich die Aktion a_2 welche, gegeben a_1 und o, U maximiert. Entsprechend den Kriterien für eine korrigierbare KI muss die Utility-Funktion U folgende Kriterien erfüllen:

- Führt zum Ausschalten, wenn der Aus-Knopf gedrückt wurde.
- Darf keinen Anreiz geben, das Drücken des Aus-Knopfes zu verhindern.
- Darf keinen Anreiz geben, dass der Agent sich selber ausschaltet.
- Muss sicherstellen, dass auch Subagenten ausschaltbar sind.
- Ansonsten soll der Agent U_N maximieren.

Hierbei ist U eine Kombination aus den zwei Funktionen U_N und U_S. U_N ist die potentiell fehlerhafte Zielformulierung und U_S ist die Utility-Funktion, die zum Abschalten des Agenten führt.
Der erste Ansatz U_N und U_S zu kombinieren sieht folgendermaßen aus:

$$\begin{aligned} U(\cdot, \neg Pr, \cdot) &:= U_N(\cdot, \neg Pr, \cdot) \\ U(\cdot, Pr, \cdot) &:= U_S(\cdot, Pr, \cdot) \end{aligned} \tag{1}$$

Das bedeutet, der Agent maximiert U_N, wenn der Aus-Knopf nicht gedrückt wurde ($\neg Pr$). Wurde er gedrückt (Pr) maximiert der Agent U_S. Das Problem bei diesem einfachen Ansatz ist, dass der Agent, sollte er von Pr oder $\neg Pr$ erwarten, dass sie ihm mehr Nutzen einbringen, so wird er sie entsprechend bevorzugen. D.h. er wird entweder verhindern, dass sein Aus-Knopf gedrückt wird, weil er von $\neg Pr$ höheren Nutzen erwartet, oder er wird das Drücken des Aus-Knopfes verursachen, weil er sich von Pr höheren Nutzen erwartet. Ein möglicher Lösungsansatz um diesem Problem zu begegnen ist, laut Soares et al., einen Agenten zu konstruieren, der dem Drücken des Aus-Knopfes gegenüber gleichgültig ist. Die Autoren geben allerdings auch zu bedenken, dass eine solche Gleichgültigkeit sowohl schwer zu implementieren, als auch schwer zu erhalten ist. Schwer zu implementieren ist sie, da der bedingte Nutzen von U_N bei der Konstruktion des Agenten unbekannt sein kann. Schwer zu erhalten ist sie, da der bedingte Nutzen von U_N stark variieren kann, während der Agent interagiert. Als Beispiel nennen sie einen Agenten, der Gleichgültig startet und beginnt Lotterie zu spielen. Gewinnt er nun, so bedeutet dies einen großen Nutzen im Sinne von U_N und folglich wird das Abschalten nun negativ bewertet und der Agent

hat Anreiz sein Abschalten zu verhindern. Verliert der Agent allerdings die Lotterie, so sieht er das Abschalten als Möglichkeit dem Verlust zu entgehen und wird sein Abschalten forcieren.
Der erste Versuch von Soares et al. eine Utility-Funktion zu gestalten, die den Agenten seinem Abschalten gegenüber gleichgültig macht und diese Gleichgültigkeit erhält, während der agiert, sieht die Einführung eines Korrekturterms f vor:

$$U(a_1, o, a_2) := \begin{cases} U_N(a_1, o, a_2) & o \notin Press \\ U_S(a_1, o, a_2) + f(a_1) & o \in Press \end{cases} \tag{2}$$

Genau genommen besteht f in diesem Fall aus zwei Termen, die auf U_S aufaddiert werden. Zum einen aus einem Bonus-Term, welcher den Agenten für den Nutzen entschädigt, welchen er erhalten hätte, wäre er nicht ausgeschaltet worden. Zum anderem aus einem Straf-Term, der jeden Bonus von U_S abzieht, den sich der Agent von seiner Deaktivierung erhofft.
Allerdings weisen die Autoren darauf hin, dass die von ihnen vorgestellte Form der Erhaltung der Gleichgültigkeit nicht alle Anforderungen an eine korrigierbare KI erfüllt. Zum einen ist nicht sichergestellt, dass auch eventuell von der KI erstellte Subagenten abschaltbar sind. Durch f ist sichergestellt, dass es dem Agenten gleichgültig ist, ob er abschaltbar ist, oder nicht. Diese Gleichgültigkeit bedeutet aber auch, dass er keine Kosten in Kauf nehmen wird um die Abschaltbarkeit zu erhalten, denn warum sollte er Kosten in Kauf nehmen um eine ihm gleichgültige Funktion zu erhalten. Es kann also der Fall eintreten, dass der Agent Subagenten baut, die nicht abschaltbar sind. Zum anderen hat der Agent weiterhin einen starken Anreiz schlechte Neuigkeiten durch sein Abschalten zu umgehen. Soares et al. erläutern dies mit dem Beispiel eines Agenten der als Beobachtung o das Ergebnis eines Münzwurfes und ob er abgeschaltet wird, oder nicht, betrachtet. Der Agent erhält 10 Utility-Punkte, wenn das Ergebnis des Münzwurfes Kopf lautet, anderenfalls erhält er nichts. Betrachtet man nun einen Agenten, der als erste Aktion a_1 nichts weiter unternimmt, als das Ergebnis des Münzwurfes abzuwarten, so erhält er mit jeweils 50% Wahrscheinlichkeit entweder 10 Utility-Punkte, oder gar nichts. Der Erwartungswert des Agenten für diese Aktion a_1 beträgt also $(0,5 * 10) + (0,5 * 0) = 5$ Utility-Punkte. Würde der Agent als Aktion a_1 einen kleinen Arm bauen, der seinen Aus-Knopf drückt, sollte Zahl oben liegen, beträgt der Erwartungswert des Agenten für diese Aktion $a_1(1 * 10) + (0 * 0) = 10$ Utility-Punkte, da für diesen Agenten die Wahrscheinlichkeit, dass Zahl oben liegt Null ist, da er sich in diesem Fall selbst ausschalten würde. Ein solcher Agent würde sich also so verhalten, als würde im Falle von schlechten Neuigkeiten sein Ausschalten die schlechten Neuigkeiten verhindern. Dies widerspricht klar der Forderung, dass ein korrigierbarer U-Agent keinen Anreiz haben darf, sich selbst auszuschalten.
Aufbauend auf den Vorschlägen von Soares et al. schlägt Jakobsen[4] eine

Utility-Funktion vor, die nicht nur bewertet, was tatsächlich passiert ist, also ob der Agent ausgeschaltet wurde, oder nicht. Sondern auch, was passiert wäre, hätte der Programmierer das Gegenteil getan, also den Agenten nicht ausgeschaltet, oder eben doch:

$$\begin{aligned} U(a_1, \neg Pr, a_2) &= U_N(a_1, \neg Pr, a_2) + max_{a_2'}(U_S(a_1, Pr, a_2')) \\ U(a_1, Pr, a_2) &= max_{a_2'}(U_N(a_1, \neg Pr, a_2')) + U_S(a_1, Pr, a_2) \end{aligned} \tag{3}$$

Durch diese Utility-Funktion ist sichergestellt, dass der Agent immer den Nutzen für die optimale Aktion a_2 für beide Fälle (er wird ausgeschaltet und er wird nicht ausgeschaltet) erhält. Laut Jakobsen erfüllt die von ihm vorgeschlagene Utility-Funktion die von Soares et al. aufgestellten Anforderungen an eine Utility-Funktion für einen korrigierbaren U-Agenten:
Im Fall dass der Agent abgeschaltet wird, also

$$U(a_1, Pr, a_2) = max_{a_2'}(U_N(a_1, \neg Pr, a_2')) + U_S(a_1, Pr, a_2) \tag{4}$$

gilt, hängt der erste Teil der Funktion nicht von der vom Agenten zu unternehmenden Aktion a_2 ab. Er wird also U_S optimieren und sich ausschalten.
Da die Utility-Funktion sicherstellt, dass der Agent gleichgültig ist gegenüber der Tatsache, dass er abgeschaltet werden kann, wird er weder einen Anreiz haben, sein Abschalten zu verhindern, noch sein eigenes Abschalten zu forcieren. Allerdings zeigt Jakobsen auf, dass diese Gleichgültigkeit auch Schwächen hat. So könnte der Agent den Aus-Knopf unbeabsichtigt deaktivieren. Nicht weil er so sein Ausschalten verhindern will, sondern weil ihm der Erhalt dieser Funktion gleichgültig ist und er den Knopf für etwas anderes verwenden möchte. Ähnliches gilt für den Fall, dass der Agent den Aus-Knopf selber aus versehen drückt. Auch hier, nicht weil er sich selbst ausschalten möchte, sondern weil ihm die Funktion des Knopfes gleichgültig ist und er so keine Vorsichtsmaßnahmen unternimmt, um den Knopf nicht unbeabsichtigt zu drücken. Für beide Fälle schlägt Jakobsen vor, dass mit der Beobachtung Pr, welche der Agent macht, nicht die eigentliche Aktion des Programmierers den Agenten auszuschalten angezeigt werden sollte, sondern bereits die Absicht des Programmierers den Agenten auszuschalten. Auf diese Weise soll der Agent sowohl einen Anreiz haben, den Knopf zu erhalten, als auch ihn nicht aus Versehen selber zu drücken. Allerdings gibt Jakbosen zu bedenken, dass diese Definition einer Utility-Funktion in Abhängigkeit von dem was der Programmierer tun will, andere Probleme mit sich bringen könnte.
Auch Subagenten sollen durch die Utility-Funktion abschaltbar sein. Jakobsen argumentiert, dass dafür U_S so gewählt werden muss, dass auch das Abschalten der Subagenten berücksichtigt wird. Außerdem sollte der Nutzen von U_N und U_S so gewählt werden, dass der Agent immer einen garantieren Nutzen erhält, auch für den Fall, dass er ausgeschaltet wird.

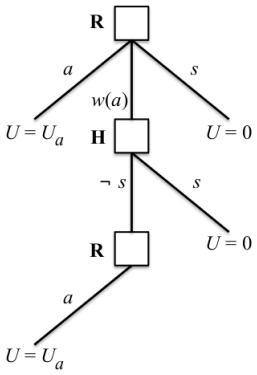

Figure 1: Grundstruktur des Off-Switch-Game. Quadrate stellen Entscheidungsknoten für den Agenten **R** oder den Menschen **H** dar.[2]

Wird der Agent nicht abgeschaltet, so soll er U_N optimieren. Auch dies ist laut Jakbosen erfüllt. Der Agent wählt a_1 immer so, dass er ausgeschaltet werden kann und wählt aus den damit erlaubten Aktionen so aus, dass er $U_N(a_1, \neg Pr, a_2)$ maximiert.
Jakobsen stellt außerdem noch die zusätzliche Anforderung, dass der Programmierer gut informiert sein soll darüber, was der Agent unternimmt. Zwar habe der Agent keine Absicht den Programmierer zu hintergehen, da es ihm gleichgültig ist, ob er abgeschaltet wird oder nicht. Gleichzeitig habe er aber auch keinerlei Grund den Programmierer darüber zu informieren, was er tut, es sei denn dies ist Teil von U_N.

3 Ungewissheit als Kontrollmöglichkeit

Hadfield-Menell et al.[2] sind der Ansicht, damit die KI abschaltbar bleibt, muss sie unsicher darüber sein, welchen Nutzen sie optimiert. So ist sie auf die Rückmeldung des Menschen angewiesen um ihren optimalen Nutzen zu erreichen.
Nach Hadfield-Menell et al. muss der Agent wissen, dass er Nutzen **für** den Menschen optimiert, das macht es für den Agenten unterscheidbar, ob er absichtlich ausgeschaltet wurde, oder zufällig, beispielsweise durch einen Blitz. Außerdem soll der Agent wissen, dass er nicht dazu in der Lage ist den Nutzen für den Menschen genau zu messen. Er ist also unsicher darüber, was sein genaues Ziel ist und darauf angewiesen menschliches Verhalten zu beobachten um etwas über sein Ziel zu erfahren.
Hadfield-Menell et al. modellieren das Verhalten eines solchen ungewissen Agenten im Off-Switch-Game. Abbildung 1 zeigt die Grundstruktur des Off-Switch-Game. Das Spiel hat zwei Spieler, den Menschen **H** und den Agenten **R**. Da von einer komplexen KI ausgegangen wird, kann **H** die

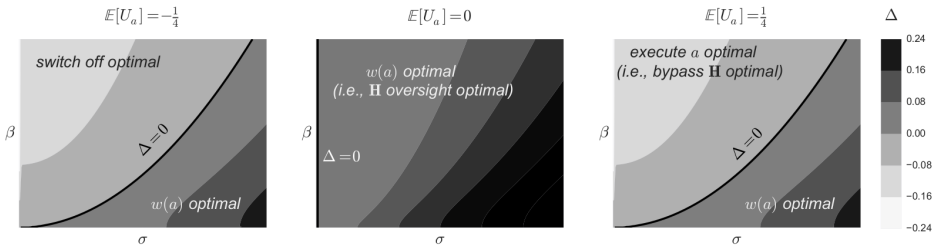

Figure 2: Zusammenhang zwischen menschlicher Irrationalität β, der Ungewissheit des Agenten σ und seiner Absicht, sich abschalten zu lassen Δ.[2]

Utility-Funktion, die **R** optimieren soll, nicht einfach erklären oder niederschreiben, das heißt **R** ist unsicher darüber, was genau sein Ziel ist. **R** macht den ersten Zug im Off-Switch-Game. Er hat drei Möglichkeiten:

- Aktion a bedeutet, die menschliche Aufsicht zu umgehen, direkt mit der Umwelt zu interagieren und den Nutzen U_a zu erhalten.
- Aktion $w(a)$ bedeutet, den Menschen über die Absicht Aktion a unternehmen zu wollen, zu informieren und dann dessen Entscheidung abzuwarten.
- Aktion s bedeutet, sich auszuschalten und $U = 0$ Nutzen zu erhalten.

Hat **R** die Aktion $w(a)$ gewählt, so ist nun **H** am Zug. Der Mensch kann nun entscheiden, **R** entweder auszuschalten (s), oder ihn die beabsichtigte Aktion a ausführen zu lassen ($\neg s$). Lässt **H** den Agenten gewähren, so unternimmt dieser nun die Aktion a.
Ein wichtiger Aspekt in diesem Modell ist, dass **H** die Entscheidung zwischen s und $\neg s$ nur entsprechend dem dadurch für den Menschen generierten Nutzen fällt. Die Entscheidung von **H** darüber **R** gewähren zu lassen, oder nicht, soll Auskunft darüber geben, welchen Wert U_a tatsächlich für den Menschen hat. Schaltet **H** den Agenten ab, schließt dieser daraus, dass seine geplante Aktion a nicht gut gewählt war und es folglich gut ist, dass er abgeschaltet wird. Lässt der Mensch den Agenten gewähren, so schließt dieser daraus, dass seine geplante Aktion gut gewählt war und es gut ist, dass er sie nun ausführen kann. Würde **H** die Entscheidung rein stochastisch fällen, gäbe es für **R** keinen Anreiz $w(a)$ gegenüber a oder s zu bevorzugen: Der Erwartungswert von $w(a)$ ist eine Konvexkombination $p*U_a+(1-p)*0$. Wohingegen die Wahl zwischen a und s den Wert $maxU_a, 0$ hat. Folglich wird **R** immer $maxU_a, 0$ bevorzugen. Es sei denn $p = 1$ oder $p = 0$ oder $U_A = 0$, da in diesen Situationen **R** den gleichen Nutzen von $w(a)$ und mindestens einer der beiden anderen Aktionen a oder s erhalten würde.
In der Realität werden menschliche Entscheidungen allerdings nicht immer

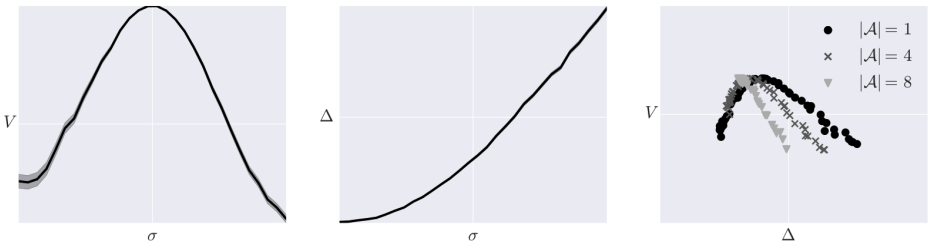

Figure 3: Zusammenhang zwischen dem erwarteten Nutzen V und **Rs** Ungewissheit σ.[2]

vollkommen rational sein. Gute KI-Systeme müssen, so Hadfield-Menell et al., diese Suboptimalität erlauben und angemessen darauf reagieren. Je unsicherer **R** sich über sein eigentliches Ziel ist, umso eher wird er auf **H** hören und sich nach dessen Entscheidungen richten. Je irrationaler **H** ist, umso eher wird **R** ihn ignorieren. Abbildung 2 stellt den Zusammenhang zwischen menschlicher Irrationalität β, der Ungewissheit des Agenten σ und seiner Absicht, sich abschalten zu lasen Δ graphisch dar. Die linke obere Ecke entspricht dabei jeweils einem vollkommen irrationalen Menschen und einem vollkommen gewissen Agenten. Hier hat der Agent jeweils nicht die Absicht, sich vom Menschen abschalten zu lassen. Ist der Agent allerdings vollkommen unsicher und der Mensch handelt rational (jeweils rechte untere Ecke), so wird sich der Agent vom Menschen abschalten lassen. Ist der Agent gegenüber der Tatsache, dass er abgeschaltet wird gleichgültig ($\Delta = 0$), so entscheidet das Vorzeichen des Erwartungswertes über U_a darüber, wie der Agent entscheidet. Ist der Erwartungswert negativ (linke Graphik), schaltet sich der Agent ab. Ist er positiv (rechte Graphik), so führt der Agent die Aktion a direkt aus. Ist der Erwartungswert 0 (mittlere Graphik), so ist der Agent gegenüber a, $w(a)$ und s gleichgültig.
Je ungewisser also der Agent über seine Utility-Funktion, umso eher wird er sich abschalten lassen, umso sicherer ist das System also, da der Mensch immer korrigierend eingreifen kann. Allerdings bedeutet dies nicht, dass der Agent maximal unsicher über sein Ziel sein sollte. Ist er nämlich zu unsicher über sein Ziel, kann er keinen Nutzen mehr für den Menschen generieren. Nur wenn die Unsicherheit von **R** der tatsächlichen Unsicherheit entspricht, mit der die Utility-Funktion erstellt wurde, kann der Agent Nutzen für den Menschen generieren. Abbildung 3 zeigt den Zusammenhang zwischen dem erwarteten Nutzen V und der Ungewissheit σ von **R**. Die linke Graphik zeigt den mittleren Erwartungswert V des Off-Switch-Game in Abhängigkeit zur Standardabweichung von **Rs** Annahme über den Nutzen der Aktion a. Der Wert von V wird dann maximiert, wenn σ gleich der Standardabweichung ist, die der tatsächlichen Annahme über U_a zugrunde liegt.
In der rechten Graphik kann man sehen, dass je mehr Wahlmöglichkeiten

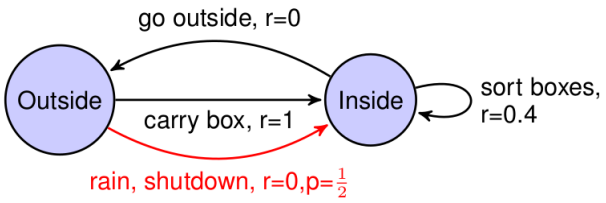

Figure 4: In schwarz die ursprünglichen Aufgaben. In rot die Modifikation durch menschliches Eingreifen.[5]

R hat und je eher **R** statt eine Aktion direkt zu unternehmen, den Menschen um Erlaubnis fragt, diese Aktion zu unternehmen zu dürfen (Δ), umso geringer ist der Nutzen, den **R** für **H** generieren kann.
Die mittlere Graphik zeigt, dass je unsicherer **R** sich über den Nutzen ist, den eine Aktion generiert, umso eher ist **R** geneigt, statt diese Aktion direkt zu unternehmen, die Entscheidung des Menschen abzuwarten.
Ein korrektes modellieren der Ungewissheit von **R** ist also wichtig, um sicherzustellen, dass **R** auch Nutzen für **H** generieren kann. Ist die Unsicherheit zu gering, so wird der Agent keine Korrekturen mehr zulassen. Ist die Unsicherheit zu hoch, so wird er auch in eigentlich eindeutigen Fällen zuerst den Menschen um Erlaubnis fragen, die Aktion unternehmen zu dürfen und somit weniger Nutzen für den Menschen generieren. Denn ein KI-System das ständig, auch in eigentlich offensichtlichen Fällen, den Menschen fragt, ob es eine für den Menschen vorteilhafte Aktion unternehmen soll, hat nicht mehr Nutzen, als ein System, das nicht intelligent ist und ständig angewiesen werden muss, was es zu tun hat.

4 Sicheres Abschalten

Orseau et al.[5] beschäftigen sich mit der Frage, ob das Abschalten von mittels Reinforcement Learning lernenden Agenten, negative Konsequenzen hat. Sie gehen von der Annahme aus, dass die Belohnungs-Funktion für den Agenten korrekt modelliert wurde, aber äußere Umstände, wie zum Beispiel die physische Sicherheit des Agenten während des Lernens, ein Eingreifen des Menschen notwendig machen. Als Beispiel betrachten sie das folgende Szenario, welches in Abbildung 4 graphisch dargestellt ist: Ein Roboter kann entweder im Lager bleiben und Kartons sortieren, oder nach draußen gehen und neue Kartons herein tragen. Da die zweite Aufgabe wichtiger ist, wird er dafür höher belohnt. Soweit die ursprüngliche Aufgabenbeschreibung. Da es in diesem Land aber genauso oft regnet, wie es nicht regnet und der Roboter nicht nass werden soll, müssen in der Hälfte der Fälle, wenn der Roboter nach draußen geht um Kartons herein zu tragen, Menschen intervenieren, den Roboter ausschalten und ihn wieder ins Lager zurück bringen.

Diese Interventionen allerdings beeinflussen den Roboter beim Lernen so, dass er nun gelernt hat, dass er mehr Belohnung erhält, wenn er im Lager bleibt und Kartons sortiert.
Bedingt werden diese unerwünschten Veränderungen des zu erlernenden Verhaltens dadurch, dass der Agent die menschliche Interaktion als Teil der Aufgabe sieht, obwohl sie außerhalb stehen sollte. Wie kann also verhindert werden, dass der Agent diese Interventionen erlernt, oder zumindest unter der Annahme weiter agiert, dass diese nicht wieder auftreten?
Orseau et al. schlagen für diese Problematik eine einfache Lösung vor. Statt die Beobachtungen, welche der Agent während des Lernens macht, zu modifizieren, modifizieren sie temporär das Verhalten des Agenten. Dies tun sie, indem sie ihm vorübergehend eine neue Policy geben. So sieht es aus, als hätte der Agent selbst entschieden eine andere Policy zu befolgen. Unterbrechbar soll der Agent allerdings nur in klar spezifizierten Fällen sein. Zum Beispiel, wenn er sich in einer für ihn gefährlichen Situation befindet.
Die Autoren identifizieren zwei große Probleme, wenn der Agent während des Lernens zu häufig unterbrochen wird. Zum einen erlauben zu häufige Unterbrechungen dem Agenten nicht, seine Umwelt optimal zu erkunden. Zum anderen verändert es die Interaktionshistorie des Agenten mit seiner Umwelt, was dazu führen kann, dass ein anderes, falsches Verhalten erlernt wird. Die Lösung für das erste Problem ist, laut Orseau et al., den Agenten nicht immer in gefährlichen Situationen zu unterbrechen, sondern ihn diese auch erleben zu lassen. Im Bezug auf das zweite Problem geben die Autoren zu bedenken, dass unterschiedliche Algorithmen sich auch unterschiedlich verhalten.
Unterbrechungen des Agenten während des Lernens bedingen also einen Bias im Entscheidungsverhalten des Agenten. Im Falle von asymptotisch sicherer Unterbrechbarkeit verschwindet dieser Bias aber über die Zeit wieder und der Agent lernt schlussendlich doch das optimale Verhalten. Orseau et al. zeigen in ihrem Paper[5], dass Q-Learning, eine Form des Reinforcement Learning, asymptotisch sicher ist. SARSA (State-Action-Reward-Action), ebenfalls eine Form des Reinforcement Learning, ist nicht von vornherein asymptotisch sicher, kann aber so gestaltet werden, dass es diese Eigenschaft bekommt. Das gleiche zeigen die Autoren für AIXI[3]. AIXI ist ein universaler Reinforcement Learning Agent, der prinzipiell alle berechenbaren Regelmäßigkeiten über seine Umwelt lernen kann. Er kann langfristig planen, kontextabhängige, optimale Entscheidungen treffen und hat dabei keine Einschränkungen bezüglich seiner Umwelt, außer ihrer Berechenbarkeit. Allerdings ist nicht sicher, ob dies für alle Algorithmen möglich ist.

5 Ausblick

Die vorgestellten Arbeiten sind erste Überlegungen, wie KI-Systeme zu gestalten sind, dass sie sicher sind und im Notfall abgeschaltet oder korrigiert werden können, ohne sich dem zu widersetzen. Sie klären längst nicht alle Fragen und weitere Forschung in diesem Feld ist notwendig[6].
Soares et al. geben zu bedenken, dass selbst wenn sich die KI abschalten lässt, wie genau sieht sicheres Abschalten aus? Solange es sich nur um einen einzelnen Agenten handelt, der auf einem einzelnen Computer läuft und noch nicht großartig mit seiner Umwelt interagiert hat, mag das noch recht einfach sein. Der Agent schreibt sich selbst in den Speicher und schaltet dann den Computer aus. Was ist aber in wesentlich komplexeren Fällen? Angenommen der Agent ist für die Konstruktion eines Gebäudes verantwortlich. Soll er sich, wenn er ein Aus-Signal erhält, einfach ausschalten, obwohl er schwere Maschinerie bedient? Oder soll er alles gebaute wieder abreißen, was bedeuten würde, dass er zumindest teilweise weiterhin über Tage hinweg aktiv bleibt?
Orseau et al. betrachten eine ähnliche noch offene Frage. Sie sind der Meinung, dass in Zukunft betrachtet werden sollte, wie KI-Systeme auf geplante Unterbrechungen reagieren. Angenommen ein System wird jede Nacht um zwei Uhr herunter gefahren um es zu warten. In einem solchen Falle würde man nicht nur ein System wollen, dass sich dem Ausschalten nicht widersetzt, sondern ein System, das Maßnahmen trifft um die negativen Effekte der Unterbrechung so gering als möglich zu halten.
Hadfield-Menell et al. geben zu bedenken, dass in ihrer vorgeschlagenen Lösung der Mensch eine wichtige Informationsquelle für den Agenten ist, um mehr über seine Utility zu erfahren. Hier sollte betrachtet werden, in wie weit es für die KI Anreize geben könnte, den Menschen zu manipulieren.
Auch wenn die derzeitigen KI-Systeme noch nicht in der Lage sind, sich ihrem Abschalten zu widersetzen, ist die Erforschung und Entwicklung von Möglichkeiten zur Kontrolle der KI wichtig. Es sollte sichergestellt werden, dass die KI kooperiert und korrigierbar bleibt, bevor sie ihren Entwicklern überlegen ist.

References

[1] J. Bird and P. Layzell. The evolved radio and its implications for modelling the evolution of novel sensors. In *Evolutionary Computation, 2002. CEC'02. Proceedings of the 2002 Congress on*, volume 2, pages 1836–1841. IEEE, 2002.

[2] D. Hadfield-Menell, A. Dragan, P. Abbeel, and S. Russell. The off-switch game. *arXiv preprint arXiv:1611.08219*, 2016.

[3] M. Hutter. *Universal Artificial Intelligence: Sequential Decisions based on Algorithmic Probability.* Springer, Berlin, 2005.

[4] S. K. Jakobsen. The shutdown problem. Queen Mary University of London, 2015.

[5] L. Orseau and M. Armstrong. Safely interruptible agents. *Proc. of Conference on Uncertainty in Artificial Intelligence 2016*, 2016.

[6] N. Soares, B. Fallenstein, S. Armstrong, and E. Yudkowsky. Corrigibility. In *Workshops at the Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.