§ 2 Wesen: Willen, Wissen und Verhalten

Da der Begriff der Künstlichen Intelligenz problembehaftet ist und zu Fehlassoziationen verleitet, selbst aber kaum einen Aussagegehalt über die beschriebene Entität hat, ist zunächst das Wesen einer Künstlichen Intelligenz genauer zu untersuchen. Von besonderem Interesse sind Fragen nach einem möglichen Willen, möglichem Wissen und dem Verhalten. Hierbei ist eine gewisse technische Vertiefung unumgänglich.[45] Das Forschungsgebiet in der Informatik, das sich mit dem beschäftigt, was heute allgemein »Künstliche Intelligenz« genannt wird, ist sogar älter als ihr Name selbst; schon seit 1943 und damit seit Aufkommen des programmierbaren Rechners[46] wird in dieser Disziplin geforscht.[47] In diesem Kontext kann zudem davon ausgegangen werden, dass sich Künstliche Intelligenz auf Maschinen,[48] im Speziellen auf programmierbare Maschinen, also Computer bezieht. Um die Grundfrage nach den möglichen Unterschieden zu anderen Computerprogrammen bzw. -software[49] genauer zu untersuchen, ist zunächst weiter auszuholen. Zu Beginn soll daher auf die Problematik des Vergleichs von Computern und Menschen eingegangen werden (A.), sodann werden kurz mögliche Definitionsversuche und deren Schwierigkeiten dargelegt (B.). Nachfolgend wird eine Reihe von gängigen KI-Methoden vorgestellt und am Beispiel des künstlichen neuronalen Netzes vertieft (C.). Danach muss für ein weiteres Verständnis dargelegt werden, was eigentlich ein Algorithmus ist und wie der Weg vom Modell zur Computersoftware aussieht (D.). Sodann wird der Determinismusbegriff in der Philosophie und der Informatik vorgestellt, um die rechtstatsächliche Ausgangsfrage des Determinismus zu klären (E.). Im Folgenden stellt sich dann die Frage, wie ein Fehler einer Künstlichen Intelligenz zu klassifizieren ist, welches Verhalten noch vorhersehbar ist und ob es Ansätze einer (teilweisen) Erklärbarkeit gibt (F.). Abschließend bleibt noch zu klären, was Wissen ist und was eine Künstliche Intelligenz wissen kann (G.).

---

45 Ebenso *Herberger,* NJW 2018, 2825.

46 Durch das sog. von Neumann-Modell, vgl. *v. Neumann,* IEEE Annals of the History of Computing 15 (1993), Issue 4, 27 ff.

47 *McCulloch/Pitts,* Bulletin of Mathematical Biophysics 5 (1943), 115 ff.

48 Also nicht auf künstlich geschaffenes biologisches Material.

49 Im Weiteren wird davon ausgegangen, dass die Begriffe »Computerprogramm«, »Programm« und »Software« – unabhängig von möglichen inhaltlichen Unterschieden – Synonyme sind.

A. Problematik des Vergleichs zum Menschen

Das Phänomen der Künstlichen Intelligenz ist Gegenstand verschiedener Disziplinen. Unter anderem wird es auch philosophisch betrachtet.[50] Hier befasst sich die Teildisziplin der Ethik mit Fragestellungen, wie auf gewisse Phänomene aus moralischer Sicht zu reagieren ist bzw. wäre und wie Entitäten im moralischen Universum einzuordnen sind bzw. wären.[51] Weiterhin wird aber auch ein Vergleich zwischen zwei Architekturen durchgeführt: Dem Menschen und dem Computer bzw. der Maschine.[52] Ein solcher Vergleich kann durchaus sinnvoll sein, allerdings verleitet er auch dazu, die Eigenschaften einer Architektur auf die andere zu übertragen. Einerseits kann damit die Angst verbunden sein, dass Maschinen dem Menschen überlegen sein können; solche Gedanken findet man gewiss schon in *Hobbes'* Leviathan, in der Gestalt des vom Menschen geschaffenen »künstlichen Tiers«.[53] Anderseits neigt man dazu, die Architekturen nur noch im Vergleich zueinander zu sehen, ohne deren spezifische Fähigkeiten zu berücksichtigen. In Bezug auf Künstliche Intelligenz heißt dies, in dieser etwas Intellektuelles oder gar Menschenähnliches zu sehen, dessen Ziel es ist, wie der Mensch zu denken[54] oder diesem sogar darin überlegen zu sein.[55]

Zunächst stellt sich erstens die Frage, ob die Probleme, die der Mensch lösen kann, auch durch einen Computer gelöst werden können. Die Beantwortung dieser Frage beginnt bei der nicht bewiesenen Church-Turing-These. Diese sagt aus, dass die (hypothetische) Turing-Maschine[56] alle intuitiv berechenbaren Funktionen berechnen kann.[57] Vereinfacht wird weiterhin angenommen, dass ein Computer eine reale Umsetzung einer

50 Vgl. *Russell/Norvig,* S. 26 ff.
51 Vgl. *Loh,* InTeR 2017, 220.
52 *Russell/Norvig*, S. 1176.
53 *Hobbes*, S. 6.
54 *Dettling/Krüger,* PharmaR 2018, 513, 514.
55 So aber wohl *Reichwald/Pfisterer,* CR 2016, 208, 211.
56 Eine Turing-Maschine kann an dieser Stelle als hypothetisches und nicht real umsetzbares Modell eines (universellen) Computers gesehen werden, ist aber eigentlich ein formales Modell, welches für exakte, mathematische Aussagen bei der Berechenbarkeit verwendet wird, wobei mehrere Arten der Turing-Maschine mathematisch modelliert werden können. Weder der formale Berechenbarkeitsbegriff noch die formalen Definitionen der möglichen Arten der Turing-Maschine sollen an dieser Stelle weiter erörtert werden, da sie für das Verständnis der Argumentation nicht von Bedeutung sind; vgl. vertiefend *Hoffmann*, Theoretische Informatik, S. 254 ff, 277 ff.
57 *Kleene*, S. 376.

Turing-Maschine sei[58] und dass unter dem (nicht wohldefinierbaren) Begriff »intuitiv berechenbar« alle (mathematischen) Funktionen verstanden werden, die man – also auch der Mensch – überhaupt berechnen kann.[59] Daraus würde folgen, dass ein Computer alles berechnen könnte, was berechenbar ist, also für alle berechenbaren Probleme eine mögliche mächtigste Problemlöse-Instanz sei. Daraus folgt aber nicht, dass auch alle berechenbaren Probleme jemals durch einen Computer berechnet werden. Für jedes Problem ist es notwendig, den Computer durch Computersoftware, die wiederum auf Algorithmen basieren,[60] die Berechnung durchführen zu lassen, sofern dies praktisch umsetzbar ist. Zugleich können (alle) berechenbaren Probleme (theoretisch) auch durch den Menschen gelöst werden; er ist daher auch eine mögliche, mächtigste Problemlöse-Instanz (für berechenbare Probleme). Gewiss lösen auch Menschen diese Probleme algorithmisch, dennoch darf daraus nicht geschlossen werden, dass die Klasse der Probleme, die Mensch und Maschine lösen können, äquivalent sei.[61] Dies ist ein prädikatenlogischer Fehlschluss, da ein Mensch nicht nur berechenbare Probleme lösen bzw. versuchen kann zu lösen, sondern auch gänzlich andere,[62] beispielsweise theologische oder moralische Probleme.[63] Daraus folgt weiter, dass Künstliche Intelligenz als Computersoftware auf berechenbare Probleme beschränkt ist, was der Mensch nicht ist.

Zweitens stellt sich die damit verbundene Frage nach der Bedeutung der Hypothese der »starken Künstlichen Intelligenz«.[64] Während »schwache Künstliche Intelligenz« nur eine einzelne Aufgabe besser lösen soll als der Mensch, soll »starke Künstliche Intelligenz« eine generelle Problemlösungsmaschine für lösbare Probleme darstellen und damit eine Maschine sein, die denkt.[65] Dieser Unterschied wird teilweise sogar als Meinungsstreit zweier Thesen dargestellt.[66] Zumindest in der technischen Betrach-

58 Was genau genommen unzutreffend ist, da eine Turing-Maschine ein unendliches Band (als Speicher) besitzt, was jedoch in der Realität eines begrenzten Universums nicht konstruierbar ist.
59 *Hoffmann*, Theoretische Informatik, S. 308.
60 S. u., Vom Modell über den Algorithmus zur Computersoftware (S. 52).
61 So aber *Reichwald/Pfisterer,* CR 2016, 208, 209.
62 Gilt unter anderem auch für nicht-berechenbare Probleme.
63 Dies gilt nur soweit, solange kein Modell gefunden wird, dass die Moral mathematisch abbildet.
64 Zu finden bei *Keßler,* MMR 2017, 589, 592; *Dettling/Krüger,* PharmaR 2018, 513, 515.
65 *Dettling/Krüger,* PharmaR 2018, 513, 514.
66 *Pieper,* InTeR 2018, 9, 11.

tungsweise spielt die starke Künstliche Intelligenz aber keine ernst zu nehmende Rolle. Diejenigen, die sich mit der technischen Umsetzung von Künstlicher Intelligenz befassen, beachten sie schlichtweg nicht oder nur peripher.[67] Gewiss ist schwer darzulegen, dass etwas nicht existiert.[68] Nach heutigem Kenntnisstand bestehen jedoch noch keine Konzepte, geschweige denn Applikationen einer starken Künstlichen Intelligenz.[69] Zudem reicht ein Blick in die technische Literatur, um zu verstehen, dass kein »künstlicher Wille« im Sinne des menschlichen Willens, »künstliches Bewusstsein« oder »künstliche Emotionen« geschaffen werden sollen,[70] sondern Fragen der Aussagen- und Prädikatenlogik, des stochastischen Schließens und heuristischer Suchverfahren behandelt werden.[71] Es geht also – wie noch dargelegt wird – um viel konkretere Ansätze zur Lösung bestimmter Probleme und nicht um einen künstlichen Menschen (sog. »Homunculus«).[72]

Letztendlich ist es notwendig, zwischen »Verhalten«, »Handeln« und »Denken« zu unterscheiden. »Denken« ist ein innerer Prozess, der von außen nicht betrachtet werden kann.[73] »Verhalten« ist dagegen ein Prozess einer verkörperten Entität über die Zeit verbunden mit einer von außen beobachtbaren Zustandsänderung, ohne dass es auf die Begründung dieser Zustandsänderung ankommt.[74] Bei einer Handlung kommt zum Verhalten noch hinzu, dass sie zielgerichtet ist.[75] Dies ist zwar ebenfalls ein inneres Attribut, jedoch im Hinblick ihres Zwecks zeigt eine Künstliche Intelligenz Verhalten. Durch die Beobachtung von außen kann aber nur auf das allgemeine Verhalten bzw. Handeln, nicht aber auf dessen Begründung – beispielsweise im Sinne von intelligentem Handeln – oder gar auf einen Denkprozess geschlossen werden. Nur weil eine Entität eine (richtige) Lösung (eines berechenbaren Problems) findet, heißt dies nicht, dass sie

67 *Russell/Norvig*, S. 1176.

68 Man müsste für alle (real existierenden) Entitäten zeigen, dass sie etwas nicht sind, was ein klassisches Induktionsproblem ist, vgl. *Mittelstraß/Haas*: Induktion, Bd. 3, S. 594 ff.

69 *Buxmann/Schmidt*, S. 6.

70 Vgl. allgemein *Ertel*; *Jarosch* und *Pérez Castaño*.

71 Vgl. insbesondere *Ertel*, S. 25 ff.

72 *Russell/Norvig,* S. 1176.

73 *Mittelstraß/Mittelstraß/Lorenz*: Denken, Bd. 2, S. 154.

74 *Sandkühler/Regenbogen*: Verhalten, Bd. 3, S. 2882 f.

75 Streitig, da bei einer intentionalistischen Ansicht das Wollen hinzukommt, vgl. *Sandkühler/Lumer*: Verhalten, Bd. 1, S. 2882 f.; im Folgenden wird aber von einer nicht-intentionalistischen Handlung ausgegangen, die sich nur durch die Zielgerichtetheit unterscheidet.

auch sinnvoll gefunden wurde oder gar ein Verständnis über den Lösungsweg vorliegt. Auf die inneren Zustände kann nicht geschlossen werden. Dies zeigt das Gedankenexperiment des sog. »Chinesischen Zimmers«.[76] Hierbei sitzt eine Person, die der chinesischen Sprache nicht mächtig ist, in einem Raum und erhält Karten mit chinesischen Schriftzeichen. Aufgabe der Person ist es, mithilfe eines für sie verständlichen und vollständigen Buches, für alle denkbaren ankommenden Schriftzeichen eine Antwort, die wiederum in Chinesisch zu verfassen ist, aufzuschreiben und herauszugeben. In diesem Sinne erzeugt das chinesische Zimmer immer eine korrekte Antwort. Dabei ist für außenstehende Beobachter – die das Innere des Zimmers nicht kennen – nicht bekannt, ob die Person im Inneren (wirklich) versteht, was sie macht. Damit kann sie nur das Verhalten dieser Entität betrachten.[77] Ein anderer Ansatz besteht darin, einfach alle möglichen Lösungswege durchzuprobieren, bis ein korrektes Ergebnis vorliegt; hierfür ist es nur notwendig zu wissen, wie das korrekte Ergebnis aussieht (sog. »Brute-Force-Ansatz«).[78] In diesem Sinne hat auch der im juristischen Schrifttum gerne zitierte »Turing-Test« nicht das Ziel, zu beweisen, dass ein Computer denkt, sondern nur, dass sich ein Computer wie ein Mensch (nach außen) verhält.[79] Die Frage, ob Computer denken können oder zumindest vorspielen können, zu denken, ist unerheblich. Der niederländische Informatiker *Edsger Dijkstra* hat es mit dem Zitat »Die Frage, ob Maschinen [wie Menschen] denken können, ist genauso interessant wie die Frage, ob U-Boote [wie Menschen] schwimmen können«,[80] polemisch auf den Punkt gebracht. Künstliche Intelligenz hat somit nicht das Ziel, Computer wie Menschen denken zu lassen,[81] sondern nur, sie wie Menschen handeln zu lassen.[82] Darüber hinaus sei erwähnt, dass an Maschinen, welche den gerne zitierten Turing-Test[83] erfüllen sollen, kaum geforscht wird[84] und dieser Test daher keine große Relevanz in der KI-Forschung hat.[85]

---

76 Zu finden auch bei *Schönberger,* ZGE/IPJ 10 (2018), 35, 40.
77 *Searle,* The Behavioral and Brain Science 3 (1980), 417 ff.
78 *Russell/Norvig,* S. 45.
79 *Russell/Norvig,* S. 23 f.
80 Mündliches Zitat, ACM 1984 South Central Regional Conference. Zu beachten ist, dass U-Boote im Englischen nicht schwimmen, sondern sich durch das Wasser bewegen (»move through the water«), vgl. *Russell/Norvig*[2], S. 948.
81 Ebenso *Stiemerling,* CR 2015, 762, 765.
82 *Russell/Norvig*, S. 1178.
83 *Ménière/Pihlajamaa,* GRUR 2019, 332, 333; *Söbbing,* InTeR 2018, 64; *v. Graevenitz,* ZRP 2018, 238, 240.
84 *Russell/Norvig*, S. 24.
85 *Ertel*, S. 4.

Somit ist auch der Rahmen dafür gesteckt, wie das Phänomen der Künstlichen Intelligenz juristisch betrachtet werden muss. Es hilft nicht, sich mit reinen Gedankenexperimenten zu beschäftigen.[86] Um der Steuerungs- und Kontrollfunktion der Rechtswissenschaften gerecht zu werden,[87] sind daher nur rechtstatsächliche Phänomene von Relevanz, die technisch möglich sind oder bei denen es zumindest signifikant wahrscheinlich ist, dass sie in näherer Zukunft umgesetzt werden. Insofern betrachtet die Rechtswissenschaft hier bestehende oder aufkommende Konflikte.[88] Technisch – und damit auch juristisch – relevant ist nur der Bereich, der als »schwache Künstliche Intelligenz« bezeichnet wird.[89] Sich juristisch derzeit mit der »starken Künstlichen Intelligenz « zu beschäftigen, wäre genauso, als würde man sich mit dem Recht der erdähnlichen Exoplaneten beschäftigen. Insofern wird im Folgenden unter »Künstlicher Intelligenz« nur noch die »schwache Künstliche Intelligenz« verstanden.

B. Definitionsversuche

Nach den bisherigen Ausführungen steht bislang nur fest, was Künstliche Intelligenz nicht ist. Zwar existiert aus technischer Sicht keine umfassend anerkannte Definition,[90] allerdings finden sich durchaus allgemeine Beschreibungen, die das Phänomen der Künstlichen Intelligenz erörtern. Wie dargelegt geht es dabei nur um Handlungen, die auch vom Menschen ausgeführt werden können. Ein treffender Definitionsversuch stammt von *Rich*: »Künstliche Intelligenz ist die Forschung, wie man Computer Dinge ausführen lassen kann, die zur Zeit noch vom Menschen besser beherrscht werden«.[91] Vom Wortlaut her problematisch, aber auch von *Rich* erkannt, ist, dass sobald ein Computer eine Tätigkeit besser kann als der Mensch,

86 Die Verf. können natürlich nicht ausschließen, dass derartige Technologien in Zukunft entstehen, halten sie aber derzeit für unwahrscheinlich.

87 Vgl. *Kaufmann/Hassemer/Neumann/Neumann*, S. 396 ff.

88 Vgl. *Steiniger,* NJW 2015, 1072, 1073 f.

89 Anders *Pieper,* InTeR 2018, 9, 14 f.

90 Ebenso im juristischen Schrifttum, *Pieper,* InTeR 2018, 9, 11; *Söbbing,* InTeR 2018, 64; *Hoeren/Niehoff,* RW 2018, 47, 49; im technischen Schrifttum: *Ertel*, S. 1 ff.; *Rich*, S. 1; *Russell/Norvig*, S. 22 ff.

91 *Rich*, S. 1. Dieses Zitat bezieht sich auf die Handlung, nicht auf das Denken, wie von *Reichwald/Pfisterer,* CR 2016, 208, 211 behauptet. Vgl. auch die englische Sprachfassung, welche von »[...] make computer do things [...], people better do« spricht, *Rich/Knight*, S. 3.

diese Lösung keine Künstliche Intelligenz mehr wäre.[92] Damit ist dann einerseits mehr das Ziel der aktuellen KI-Forschung gemeint, andererseits wird klar, dass KI-Methoden irgendwann zur regulären Informatik werden.[93] Zudem konnte ein Computer immer schon gewisse Dinge besser als der Mensch; der Computer wurde ursprünglich als Rechenmaschine geschaffen, sollte also zumindest das Rechnen als Handlung immer schon besser – in diesem Sinne schneller – können als der Mensch.[94] Würde man das Ganze auf die Spitze treiben, würde dies sogar auf jede Maschine zutreffen, die sich der Mensch als Hilfsmittel heranzieht.

Nimmt man neben dem Begriff des Computers den Begriff des Agenten hinzu, wird eine andere Sichtweise eröffnet. Agenten sind Entitäten, die ihre Umgebung wahrnehmen und entsprechend reagieren.[95] Zwar ist auch jeder Computer ein Agent, allerdings wird er nun von seinem Verhalten und seiner Interaktion her betrachtet. Künstliche Intelligenz kann man nun als den Versuch sehen, den besten rational handelnden Agenten auf einer bestimmten Architektur für ein bestimmtes Problem zu finden.[96] Gewiss ist damit nicht definiert, was man unter »rationalem Handeln« versteht, dies kann aber auch offenbleiben. Deutlich wird aber, dass es um die Lösung bestimmter (berechenbarer) Probleme mit einem bestimmten – dem möglichst besten – Ansatz geht, so dass dem Agenten scheinbar eine Form von Intelligenz zukommt. Folglich ist unter Künstlicher Intelligenz Computersoftware zu verstehen, welches zur verbesserten Lösung von bestimmten Problemen eingesetzt wird.[97] Die Fähigkeiten derartiger Computersoftware sind Folge einer fortschreitenden Entwicklung der Leistungsfähigkeit von Computern, mit der nun bestehenden Möglichkeit, schnell auf große Datenmengen zuzugreifen und diese systematisch zu analysieren.[98]

92 In diesem Sinne *Rich*, S. 1.
93 Zitat von *Kohlhase* aus der Vorlesung zur Künstlichen Intelligenz v. 19.10.2017.
94 *Aspray*/*Aspray*, S. 251 ff.
95 Zu eng daher bei *Sester/Nitschke,* CR 2004, 548.
96 *Russell/Norvig*, S. 25 f.
97 So schon bei *Konertz/Schönhof,* ZGE/IPJ 10 (2018), 379, 380 ff.
98 *Russell/Norvig,* S. 36 f.

C. Technologien der Künstlichen Intelligenz

Die bisherigen Beschreibungen waren freilich noch abstrakt. Genau wie es auch bei dem eingangs genannten technischen Phänomen der Eisenbahn notwendig ist, sich mit dessen konkreten technischen Eigenarten zu befassen, um aus diesen rechtswissenschaftlich fundierte Schlussfolgerungen zu ziehen, ist es unumgänglich, sich mit den Funktionsprinzipien realer KI-Methoden auseinanderzusetzen, insbesondere, um deren Wesen näher zu ergründen.

I. Vom einfachen Schachcomputer zu AlphaZero

Bevor jedoch ausgewählte KI-Methoden vorgestellt werden, sei zu Beginn an einem einfachen Beispiel die reale Verwendung von Künstlicher Intelligenz sowie deren Fähigkeiten dargelegt. Als prägendes Beispiel werden auch in der juristischen Literatur gerne die Leistungen bei Brettspielen von Schach oder GO herangezogen.[99] In den letzten Jahren sind Brettspielcomputer mitunter in ihrer Architektur bemerkenswerten Veränderungen unterzogen worden. Das Ziel ist nicht nur, Programme zu erschaffen, die besser als ein Mensch spielen, sondern auch außerordentlich komplexe Probleme effizienter zu bewältigen. Das Brettspiel Schach kann auf viele Weisen eröffnet werden: Insgesamt stehen einem Spieler 20 Möglichkeiten offen, ein Spiel zu eröffnen.[100] Ebenso existieren 20 weitere Möglichkeiten des Gegners, sein Spiel zu eröffnen. So sind nach der ersten Runde bereits 400 verschiedene Stellungen möglich. Die Menge aller möglichen Schachspiele wird mit 10^{120} beziffert.[101] Im Vergleich sei angemerkt, dass nur ca. 10^{79} Teilchen im bekannten Universum vermutet werden.[102] Als Folge würde ein Brute-Force-Ansatz schon an der Komplexität des Problems scheitern. Die ersten Schach-Algorithmen haben versucht, alle möglichen Züge für möglichst viele Runden zu simulieren und deren Effektivität zu bewerten.[103] Selbst mit erheblicher Rechenleistung konnten nur einige wenige Runden vorberechnet werden. Nachfolgesysteme hatten primär das Ziel, diesen Algorithmus effizienter zu betreiben, beispielsweise indem der Anteil an offensichtlich erfolglosen Zügen frühzeitig erkannt und ignoriert

99 *Grapentin,* NJW 2019, 181, 183; *Pieper,* InTeR 2018, 9, 11 f.
100 Die acht Bauern und zwei Springer können jeweils zwei Felder erreichen.
101 *Shannon,* Philosophical Magazine, Ser. 7 (1950), Vol. 41, 256, 259 f.
102 *Hanslmeier*, S. 520.
103 Vgl. *Shannon,* Philosophical Magazine, Ser. 7 (1950), Vol. 41, 256, 262 ff.

wurde.[104] Die gesparte Rechenleistung konnte in die Evaluation vielversprechenderer Züge reinvestiert werden. Dies findet sich auch in modernen Schach-KIs, wie Stockfish, wieder.[105]

Einen Paradigmenwechsel erfuhren die Spielecomputer spätestens[106] mit dem Programm AlphaZero für Brettspiele wie Schach, Shogi und GO, welche die Suchstrategien durch künstliche neuronale Netze[107] und Monte-Carlo-Suchbäume verdrängten.[108] Das Training dieser künstlichen neuronalen Netze erfolgte jedoch ausschließlich durch Partien gegen sich selbst. Vereinfacht lässt es sich wie folgt veranschaulichen: Zu Beginn stehen sich zwei (virtuelle) Teams gegenüber,[109] welche noch nie GO gespielt haben. Sie wissen nichts über das Spiel, nicht einmal, wann ein Sieg vorliegt, wurden aber auf einen Sieg hin optimiert. Ein Teammitglied (das neuronale Netz) übernimmt die Rolle des Strategen und ermittelt, welche Züge infrage kommen, das andere Teammitglied (das Suchprogramm) übernimmt die Rolle des Überwachenden und versucht, zu bestimmen, wie gut diese Züge tatsächlich waren. Abwechselnd generieren die Teams nun ihre Strategien, um die Figuren auf dem Brett zu verschieben. Ein »virtueller Schiedsrichter« achtet auf die Einhaltung der Regeln und bestimmt, wann eine Partie gewonnen wurde oder die zulässige Spielzeit abgelaufen ist. Sobald die Partie entschieden oder beendet wurde, versucht das neuronale Netz seine Taktik anzupassen, indem es das Feedback des Suchalgorithmus und das des Schiedsrichters berücksichtigt. Im Anschluss beginnt die nächste Partie. Am Ende des Trainings können die Strategen die erfolgversprechendsten potentiellen Spielzüge ausfindig machen, sodass der Suchalgorithmus nur die besten Strategien auswerten muss.[110] Alleine dieses Beispiel zeigt bereits, worum es bei Künstlicher Intelligenz geht: Es wurde versucht, den am besten rational handelnden Agenten für die Spiele zu finden, nicht jedoch die Spiele wie ein Mensch »denkend« zu lösen.

104 Vgl. *Nilsson*, S. 89 f.
105 *Czarnul*, in: *Shi et al.,* S. 458.
106 Self-play Neuronale Netze wurden bei simpleren Spielen schon früher in der Literatur vorgeschlagen, vgl. für das Spiel Backgammon *Tesauro*, Artificial Intelligence 134 (2002), 181 ff.
107 S. u., Beispiel: (Künstliches) neuronales Netz (S. 45).
108 *Silver/Hubert/Schrittwieser et al.*, S. 12.
109 In Wirklichkeit spielt der Computer gegen sich selbst.
110 *Silver/Schrittwieser/Simonyan et al.* Nature, Vol. 550 (2017), 354 f.

II. Methoden der Künstlichen Intelligenz im Überblick

Im vorherigen Abschnitt wurden bereits zwei KI-Methoden genannt: »Das künstliche neuronale Netz« und »Suchstrategien«. Schon daran zeigt sich, dass es eine Fehlvorstellung ist, dass es die Künstliche Intelligenz »an sich« gibt. Unter den Begriff »Künstliche Intelligenz« fallen vielmehr eine Fülle von unterschiedlichen Ansätzen und Verfahren, welche in der Lage sind, bestimmte Probleme zu lösen. Nachfolgend sollen eine Reihe von bekannten Methoden der Künstlichen Intelligenz und ihre Anwendungsgebiete in einer kurzen Übersicht charakterisiert werden. Hierbei soll auch die Unterscheidung zwischen Künstlicher Intelligenz und der Untergruppe des sog. »*Machine Learning*«[111] illustriert werden.[112]

1. Suchstrategien

Zu den wohl einfachsten und ältesten Formen der Künstlichen Intelligenz zählen die Suchstrategien. Sie eignen sich zum Lösen einer Vielzahl von Problemen des Alltags. Sie finden bei heuristischen Suchen Anwendung, aber auch – in stark optimierter Form – bei Routenplanern in der Logistik.[113] Anschauliche Beispiele für die Suchstrategien sind die Ermittlung eines Weges durch ein Labyrinth oder die Lösung eines Schachspiels.[114] Obgleich das Durchschreiten eines Labyrinths und das Lösen eines Schachspiels anscheinend wenig gemeinsam haben, können sie dennoch auf ein gemeinsames Problem reduziert werden: Wie soll sich der Algorithmus an einer beliebigen Stelle im Labyrinth bzw. zu einer beliebigen Spielsituation während des Schachspiels entscheiden? Während das Labyrinth-Problem recht übersichtlich ist, da hier immer nur die Wahl zwischen rechts oder links besteht, ist die Wahl beim Schach vielfältiger. Der Algorithmus kann mit jeder Figur eine bestimmte Menge an Spielzügen machen. Eine Suchstrategie würde nun sämtliche Entscheidungsmöglichkeiten ausprobieren;

111 Auf Deutsch »Maschinelles Lernen«.

112 Das sog. »künstliche neuronale Netz« wird im nachfolgenden Abschnitt vertiefend erläutert, s. u., Beispiel: (Künstliches) neuronales Netz (S. 45).

113 Beispielsweise der sog. »Dijkstra Algorithmus«, vgl. *Nha/Djahel/Murphy*, S. 1 ff.; *Cormen/Leiserson/Rivest/Stein*, S. 670 ff.

114 Vgl. *Moore*, S. 1 ff.; *Shannon,* Philosophical Magazine, Ser. 7 (1950), Vol. 41, 256, 264 ff.

in den genannten Beispielen also systematisch das Labyrinth oder die möglichen Schachzüge berechnen.[115] Sobald das Ende des Labyrinths bzw. ein vorteilhafter Zug gefunden wurde, wird dieser ausgeführt. Problematisch ist, dass ein solcher Algorithmus schnell an seine Grenzen gerät, wenn dessen Wachstum schneller als polynomielles Wachstum[116] ist – wie beispielsweise bei Schach – und damit sowohl die nötige Rechenzeit als auch der notwendige Speicherbedarf unpraktikabel werden.[117] Nichtsdestotrotz ist die Suchstrategie ein weitverbreitetes Werkzeug, welches – sofern das Problem nicht zu komplex oder kompliziert wird – stets die beste Lösung findet. Allerdings besitzen Suchstrategien kein Gedächtnis, welches ihnen erlauben würde, den gelernten Weg wiederzuverwenden.

2. Entscheidungsbäume

Bei einem Entscheidungsbaum handelt es sich um eine Kette von Wenn-Dann-Entscheidungen.[118] Vereinfacht ist dies dem Juristen bei der Fallprüfung bekannt. Nimmt man als (vereinfachtes) Beispiel die Prüfung der zivilrechtlichen Stellvertretung, dann muss eine eigene Willenserklärung, die im Namen eines anderen mit Vertretungsmacht abgegeben wurde, geprüft werden (Abbildung 1). Sobald eines der Elemente nicht bejaht werden kann, führt die Prüfung zu dem Ergebnis, dass eine wirksame Stellvertretung nicht vorliegt.[119]

115 *Shannon*, Philosophical Magazine, Ser. 7 (1950), Vol. 41, 256, 259.
116 Beispielsweise exponentielles Wachstum.
117 Vgl. *Cormen/Leiserson/Rivest/Stein*, S. 1059 ff.
118 *Söbbing*, CR 2020, 223, 224 nimmt den Entscheidungsbaum als grundsätzliche Methode an, Algorithmen darzustellen, reduziert ihn allerdings auf seine grafische Darstellung.
119 Das Schema ist hier bewusst vereinfacht worden, um die Grafik nicht zu überfrachten.

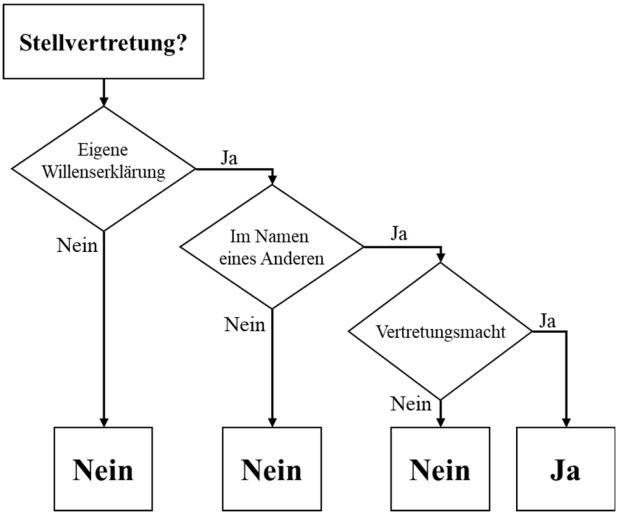

Abbildung 1: Das (vereinfachte) Prüfungsschema als Illustration eines Entscheidungsbaums

Bei den Entscheidungsbäumen wird eine Reihe von numerischen Entscheidungen zur Bestimmung eines Ergebnisses genutzt, was gewiss – genauso wie reale Stellvertretungsprüfung – komplizierter ist. In Abgrenzung zu den Suchstrategien müssen Entscheidungsbäume zum Erreichen des Zieles den Suchraum nicht ständig neu durchlaufen. Ein Entscheidungsbaum speichert die Schwellenwerte und deren Reihenfolgen. Das so ermittelte Regelwerk kann deswegen wiederverwendet werden, weswegen die Methode des Anlernens von Entscheidungsbäumen nicht nur zu der Gruppe der Künstlichen Intelligenz zählt, sondern auch zum *Machine Learning*. Der Entscheidungsbaum soll nachfolgend an einem der bekanntesten Beispielaufgaben des *Machine Learning* illustriert werden, dem sog. »Titanic Datenset« der Universität Stanford.[120]

120 Stanford University, Titanic Dataset, »A Titanic Probability«, verfügbar unter: https://web.stanford.edu/class/archive/cs/cs109/cs109.1166/problem12.html (zuletzt abgerufen am 01.06.2020).

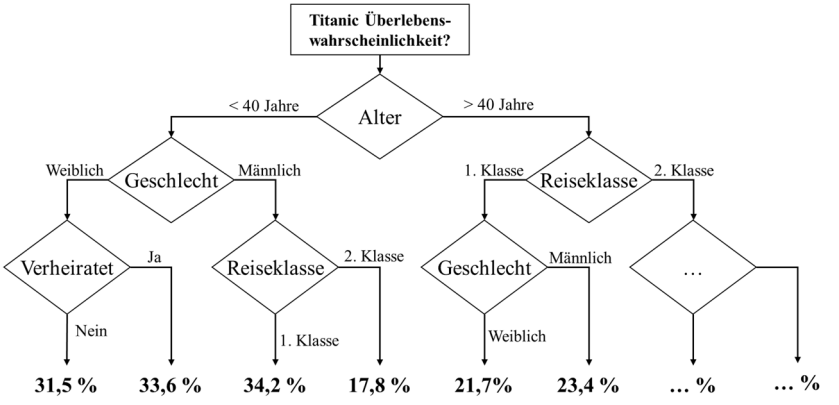

Abbildung 2: Schematisches Beispiel eines Entscheidungsbaums des Titanic Datensets

Der Titanic Datensatz ist im Kern eine Liste von 887 Passagieren, welche sich zum Zeitpunkt der Jungfernfahrt im Jahre 1912 an Bord befanden. Neben dem Namen der Person enthält diese Liste unter anderem auch Informationen über Reiseklasse, das Geschlecht, das Alter sowie die Information, ob die Person das Schiffsunglück überlebt hat. Die Aufgabe ist nun üblicherweise, einen Entscheidungsbaum zu ermitteln, welche die Überlebenswahrscheinlichkeit einer Person prognostizieren kann. Hierzu wird zu jeder Eigenschaft ein Schwellenwert gesucht, bei dessen Anwendung das Datenset möglichst zutreffend aufgeteilt wird. So kann es eine bestimmte Altersgrenze geben, ab der ein Überleben zunehmend (un)wahrscheinlich wird. Gleiches gilt für die Frage des Geschlechts oder der Reiseklasse. Während des Lernprozesses, werden diese Schwellenwerte sowie die Reihenfolge, nach der sie optimalerweise abgefragt werden, sukzessive ermittelt. Mit Abschluss des Lernprozesses werden die Entscheidungsregeln sowie Schwellenwerte abgespeichert und können anschließend zur Ermittlung der Überlebenswahrscheinlichkeit genutzt werden (Abbildung 2).

Obgleich dieses Beispiel sehr realitätsfern wirkt, so illustriert es doch die Funktionsweise von Entscheidungsbäumen. Beispielsweise finden sich bei Versicherungen, Entscheidungsbäume, welche anhand von Versicherungsdaten aus der Vergangenheit erlernt werden und dessen Ergebnisse auf neue Kunden übertragen werden.[121] Auf diese Weise kann bei Lebensver-

121 Vgl. *Frempong/Nicholas/Boateng*, International Journal of Statistics and Applications 2017, 117 ff.

sicherungen das voraussichtliche Alter der Versicherungsnehmer abgeschätzt werden. Im Bankwesen kann mit Entscheidungsbäumen die Kreditwürdigkeit sowie der angemessene Zinssatz eines Kreditinteressenten ermittelt werden.[122] Hierbei ist die Eigenschaft, dass die Entscheidungsprozesse für den Menschen nachvollziehbar sind, ein entscheidender Vorteil im Gegensatz zu vielen anderen Methoden der Künstlichen Intelligenz.[123]

3. Clustering-Algorithmen

Clustering-Algorithmen sind eine sehr alte Form der Künstlichen Intelligenz und zählen – ebenso wie die Entscheidungsbäume – zu den Methoden des *Machine Learning*. Ziel dieser Algorithmen ist es, zusammenhängende Gruppen (sog. »Cluster«) in Datenpunkten zu identifizieren (Abbildung 3).[124]

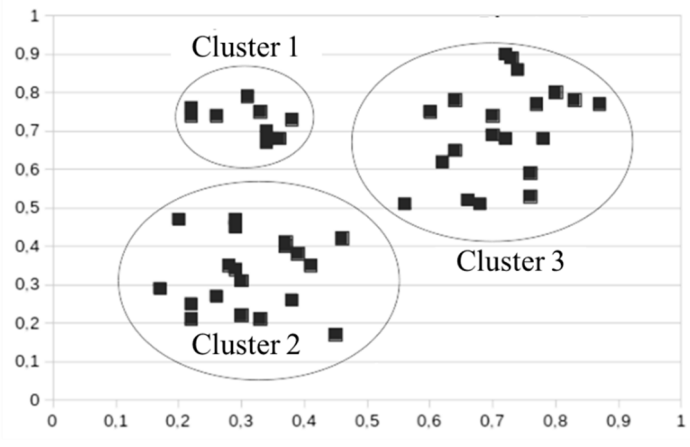

Abbildung 3: Punkte, welche in drei Cluster unterschieden werden können.

Ein geeignetes Beispiel für Clustering-Algorithmen ist die Warenkorbanalyse im Handel.[125] Hierbei werden die Warenkörbe in Bezug auf deren Zusammensetzung in Verbindung mit weiteren verfügbaren Daten –

122 Vgl. *Jiang*, in: WRI, S. 18 ff.
123 Vgl. *Arrieta/Dias-Rodríguez/Del Ser et al.*, S. 14 ff., 28.
124 Vgl. *Omran/Engelbrecht/Salman*, Intelligent Data Analysis 11 (2007), 583.
125 Vgl. *Liu/Lee/Mu*, Knowledge Management Research 19/4 (2018), 59 ff.

beispielsweise Kundendaten – geclustert. Werden beispielsweise Nutzerdaten eines Online-Shops gespeichert, versucht ein Clustering-Algorithmus nun innerhalb dieser Daten Gruppen zu identifizieren, sodass den Nutzern für sie spezifische Waren angeboten werden können mit dem Ziel, die Verkaufswahrscheinlichkeit der angebotenen Artikel kontinuierlich zu erhöhen.

4. Fuzzylogiken

Im Gegensatz zur bekannten klassischen Logik mit ihren beiden (booleschen bzw. binären) Werten »wahr« und »falsch«,[126] ist die Fuzzylogik[127] eine unscharfe Logik, die zwischen den sicheren und den garantiert nicht eintretenden Ereignissen jegliche stetige Zwischenwerte annehmen kann.[128] Im Kern erlauben Fuzzylogiken damit einem Computer, mit unscharfen Informationen umzugehen. Das Anwendungsgebiet von Fuzzylogiken ist schwerpunktmäßig in der Regelungstechnik verortet (sog. Fuzzy-Regler).[129] Grundsätzlich sind solche unscharfen Informationen in digitalen Rechnern allerdings problematisch,[130] da ein Computer für jede Situation einen exakten Zustand einnehmen muss.[131] Dies verhindert, dass unscharfe Regeln, wie Menschen sie täglich anwenden, sich eins zu eins auf einen Computer übertragen lassen. Nimmt man beispielsweise die Steuerung einer Drohne, die in der Luft stehen und das Gleichgewicht halten soll, sind die Regeln hierzu verbal einfach formuliert: Wenn sich die Drohne zu weit nach links bewegt, soll sie mehr nach rechts fliegen. Wenn sie zu hoch ist, soll sie sinken etc. Für den Flugregler der Drohne – der letzten Endes nur darüber entscheiden kann, ob er einen der Motoren stärker oder schwächer antreibt – ergeben sich dadurch eine Reihe von Regeln, welche es zu erfüllen gilt. Problematisch ist nun, dass der Flugregler erkennen muss, wie hoch zu hoch sein soll. An diesem Punkt setzen Fuzzylogiken an, welche anhand von einfachen Funktionen (den sog. »Zugehörigkeitsfunktionen«) diese Zuordnungen ermöglichen. Die Zugehörigkeitsfunktionen weisen den (unscharfen) Zuständen einen Wert zu, sodass mithilfe dieser Werte entsprechend geregelt werden kann. Ist beispielsweise die Drohne zu 70% in

126 Vgl. *Schnapp*, S. 13 ff.
127 Vom englischen Begriff »fuzzy«, zu Deutsch »verwischt«.
128 *Dassow*, S. 123.
129 Vgl. *Thomas*, S. 168.
130 Vgl. *Ansari*, Computer Education 88 (1998), 5.
131 Vgl. *Hoffmann*, Technische Informatik, S. 33 ff.

der richtigen Höhe und zu 30% zu hoch, kann durch diese Aufteilung der Flugregler – bei einer an sich binären Entscheidung – auch auf komplexe Flugsituationen differenziert reagieren, indem er beispielsweise die Motorleistung mit rund einem Drittel seiner möglichen Intensität anpasst. Der Vorteil von Fuzzylogiken ist es, Maschinen auf eine verhältnismäßig einfache und intuitive Weise zu regeln. Schon seit einiger Zeit finden Fuzzylogiken Anwendung in Autopiloten der Luftfahrt[132] oder in Antiblockiersystemen von Kraftfahrzeugen.[133] Allerdings wird diese Technologie zunehmend auch in einfachen Haushaltsgeräten benutzt: Fuzzylogiken werden in modernen Waschmaschinen eingesetzt, um beim Waschen den Verschmutzungsgrad der Wäsche abzuschätzen und Waschvorgänge zu optimieren.[134]

5. Semantic Web Systeme/Expertensysteme

Die bisher dargelegten Methoden zeichnen sich dadurch aus, dass sie zur Lösung von Problemen herangezogen werden, welche kein oder wenig Wissen erfordern. Das stellt sich bei komplizierten Fragestellungen anders dar, wie sie beispielsweise in einer Quizshow gestellt werden. Für die Arbeit mit vernetzten Informationen eignen sich Semantic Web Systeme bzw. Expertensysteme.[135] Kern eines solchen Systems ist ein Wissensgraph,[136] welcher Entitäten und Klassen in einen logischen Zusammenhang bringt und diese durch einen Computer ausgelesen werden können. Ein exemplarischer Ausschnitt eines Wissensgraphen ist in Abbildung 4 skizziert.[137] Er bildet ein paar wenige beispielhafte Informationen über die beiden Physiker *Albert Einstein* und *Vera Kistiakowsky* ab, wie den Geburtsort, den Beruf oder in welchem Staat sich ein für die Personen relevanter Ort befindet.

---

132 Vgl. *Luo/Lan*, in: *Li/Gupta*, S. 90 ff.
133 *Unlusoy/Yazicioglu*, Int. J. Vehicle Design, Vol 48(3), 1 ff.
134 *Agarwal*, S. 2 ff.
135 Vgl. *Davies/Warren*, in *Domingue/Fensel/Hendler*, S. 737 ff.
136 Teils wird in der Literatur auch von einer Ontologie gesprochen.
137 Es handelt sich hier um eine vereinfachte Darstellung. Damit ein Computer einen Wissensgraphen durchsuchen kann, muss dieser eine spezielle Syntax verwenden. Hierbei bedient man sich üblicherweise dem Ressource Description Framework (RDF) oder in komplexen Fällen der Web Ontology Language (OWL), s. u., Wissen und dessen Repräsentation (S. 66).

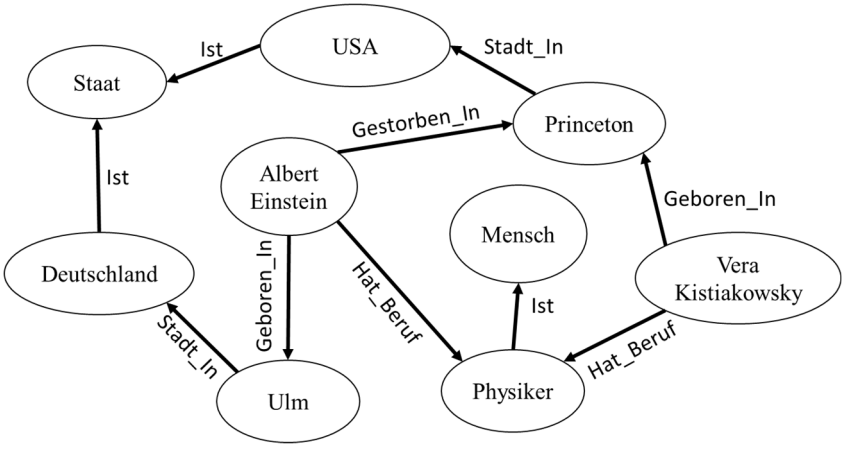

Abbildung 4: Beispiel eines einfachen Wissensgraph

Obgleich dieser Wissensgraph sehr trivial gehalten ist, sind bereits einfache maschinelle Abfragen möglich. Es kann nach dem Geburtsort von *Albert Einstein* oder nach dem Beruf von *Vera Kistiakowsky* gefragt werden. Das KI-System würde in diesem Fall »Ulm« und »Physiker« ausgeben. Gleichzeitig ist der Wissensgraph ausreichend komplex, dass erste Schlussfolgerungen[138] durchgeführt werden können. Möglich sind Abfragen nach der Stadt, in der ein Physiker gestorben und ein weiterer Physiker geboren wurde. Der sog. »Reasoning-Algorithmus« wird den Wissensgraphen nach einer Entität durchsuchen, die der Frage genügt und dabei auch Aussagen des Wissensgraphen kombinieren.[139] Im vorliegenden Beispiel wäre die Ausgabe »Princeton«.

Es ist zu beachten, dass für einen außenstehenden Dritten – der keinen Einblick in den Wissensgraphen hat – ein solches KI-System den Eindruck erweckt, dass es die Aussagen und Konzepte versteht. Es wirkt daher so, als hätte die Künstliche Intelligenz ein Verständnis von dem Konzept eines Physikers, einer Person oder eines Ortes. Das Verhalten des KI-Systems hängt aber nur von dem zugrunde liegenden Wissensgraphen ab. Wird im oben gezeigten Wissensgraphen beispielsweise danach gefragt, was »Deutschland« sei, wird lediglich das ausgegeben, auf was der Zeiger »Ist« zeigt, also »Staat«. Ein Verständnis dieser Begriffe liegt aber nicht vor.

138 Auch »Reasoning« genannt.

139 Diese Algorithmen verwenden meist eine Form der Suchstrategien, s. o., Suchstrategien (S. 34).

Insofern ähnelt diese Form der Künstlichen Intelligenz einer sehr komplexen Datenbankabfrage. Dennoch lassen sich – sofern die Wissensgraphen sehr umfangreich sind – sehr komplexe Schlussfolgerungen ziehen, welche auch jenseits der menschlichen Fähigkeiten liegen können. So kennt der Wissensgraph »YAGO 4« über 10.000 Klassen und über 300 Millionen Fakten.[140] Die einfache Maschinenzugänglichkeit der Information machen sich auch Suchmaschinen wie »Google« und »Yahoo!« mit eigenen Wissensgraphen zunutze, um Informationen bereit zu stellen.[141] KI-Systeme wie diese bilden das Fundament für komplexe Programme wie das IBM Watson System, welches in der Lage ist, sich bei einer Quizshow gegen Menschen zu behaupten.[142] Es ist jedoch zu beachten, dass gegenwärtig Wissensgraphen mit einer hohen Qualität nicht ohne Zutun des Menschen erstellt werden können,[143] was einer weiten Verbreitung dieser Technologie bisher entgegen steht.

6. Evolutionäre Algorithmen

Eine exotischere Form der Künstlichen Intelligenz sind die evolutionären Algorithmen. Wie der Name vermuten lässt, sind evolutionäre Algorithmen von biologischen Prozessen in der Natur inspiriert und versuchen deren Strategien zum Bewältigen von Optimierungsproblemen nachzuahmen. Ein evolutionärer Algorithmus läuft in drei Schritten ab: Initiierung, Evaluation und Optimierung. Das lässt sich am (vereinfachten) Beispiel des Berechnens einer optimalen Antennengeometrie aus einem einfachen Stück Draht veranschaulichen.[144] Das Ergebnis eines solchen Prozesses ist die in Abbildung 5[145] gezeigte Antenne,[146] die im Rahmen einer NASA-Mission entwickelt wurde und sich unter anderem durch eine bessere Sende- und Empfangsleistung bei einem gleichzeitig sehr simplen Design und einer verkürzten Entwicklungszeit auszeichnet.[147] Daran zeigt sich, dass evolutionäre Algorithmen bei solchen Ansätzen problemlos Lösungen finden

140 *Pellissier-Tanon/Weikum/Suchanek*, The Semantic Web 2020, 583, 594.
141 *Paulheim*, The Semantic Web 2017, 589, 493.
142 *Gliozzo/Ackerson/Bhattacharya et al.,* S. 30.
143 *Pellissier-Tanon/Weikum/Suchanek*, The Semantic Web 2020, 583, 586
144 Vgl. *Hornby/Lohn/Linden*, Evolutionary Computation 19 (2011), 1 ff.; Dieses Beispiel findet sich schon bei *Konertz/Schönhof*, ZGE/IPJ 10 (2018), 379, 398 ff.
145 Entnommen aus *Lohn/Hornby/Linden*, AI EDAM 22 (2008) 235, 242.
146 Vgl. U.S. Patent Nr. 5,719,794.
147 *Hornby/Lohn/Linden*, Evolutionary Computation 19 (2011), 1, 20.

können, die unter objektiver Betrachtung als (patentrechtliche) Erfindungen gelten würden.[148]

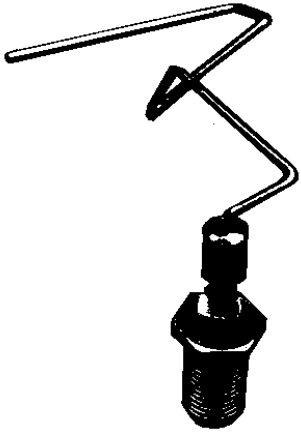

Abbildung 5: Eine evolutionär designte Antenne für einen Satelliten

Damit ein evolutionärer Algorithmus eine Lösung optimieren kann, ist die Grundvoraussetzung, dass ein – üblicherweise ungenügender – Lösungsansatz existiert; im Beispiel der Antenne ein Stück Draht. Dieser Lösungsansatz wird bei der Initiierung zufällig modifiziert; beim Antennenproblem werden also zufällige Antennengeometrien generiert. Diese Antennengeometrien stellen ein Stück Draht mit einem zufälligen Knick dar. Die so genierten Lösungen werden anschließend evaluiert,[149] wobei sie sich nach wie vor als ungenügend erweisen werden. Dennoch unterscheiden sich die Ergebnisse voneinander und manche Antennengeometrien werden weniger schlecht bzw. minimal besser sein als andere.

An diesem Punkt beginnt nun die Optimierung durch den evolutionären Algorithmus. Dabei können vier Phasen unterschieden werden: Selektion, Rekombination, Mutation und Evaluation, die – in Schleifen – immer wieder durchlaufen werden. Zunächst werden die schlechtesten Lösungen entfernt (Selektion). An deren Stelle tritt eine Kreuzung aus den Lösungsvarianten, welche verblieben sind (Rekombination). Dadurch wird die Menge der Lösungen, die durch die Selektion verkleinert wurde, wieder aufgefüllt, sodass die Anzahl der Lösungen mit der Anzahl nach der

148 *Konertz/Schönhof,* ZGE/IPJ 10 (2018), 379, 397 ff.
149 Üblicherweise durch eine Simulation.

Initiierung identisch ist. Damit sich die Lösungen nun über eine simple Kreuzung hinaus weiterentwickeln können, bedarf es im Weiteren einer Änderung der Lösungen. Dafür wird ein Teil der Lösungen (pseudo-)zufällig und geringfügig verändert (Mutation). Dieser neue Satz an Lösungen wird nun erneut evaluiert und der Prozess beginnt so lange von vorne, bis die Evaluation die gewünschte Qualität erreicht oder weitere Durchläufe zu keiner signifikanten Verbesserung führen.[150] Im Antennenbeispiel bedeutet dies, dass die zufällig generierten und anschließend evaluierten Antennen zunächst selektiert werden. Die fehlenden Antennengeometrien können nun entweder durch Kombination erzeugt oder durch Kopieren der besten Antennengeometrien ersetzt werden. Im Anschluss wird ein Teil der Antennengeometrien geringfügig mutiert, beispielsweise durch geringfügige Änderungen des Biegewinkels an einer beliebigen Stelle. Diese neuen Antennengeometrien werden abschließend in einer Simulation evaluiert und bewertet. Mit der anschließenden Selektion beginnt der Durchlauf erneut. Mit jedem Durchlauf führen kleine zufällige Änderungen dazu, dass sich die Leistung der Antennen verändert. Durch den fingierten »evolutionären Druck« wird sich stets die leistungsfähigste Antenne behaupten.

7. Zwischenergebnis

Es lässt sich festhalten, dass es eine Vielzahl von KI-Methoden gibt, die an dieser Stelle nicht abschließend dargelegt wurden. Es kann auch bezweifelt werden, dass aufgrund des schnellen Fortschritts in der Entwicklung der KI-Methoden derzeit überhaupt eine abschließende Darstellung erfolgen kann. Gezeigt wurde auch, dass die verschiedenen Methoden, unterschiedliche Ansätze zur Lösung unterschiedlicher Probleme aufweisen. Hinzu kommt, dass es möglich ist, ein Problem mit unterschiedlichen KI-Methoden zu lösen. Daher lassen sich an dieser Stelle zwei wichtige Aspekte festhalten: Erstens, es gibt nicht »die« Künstliche Intelligenz und zweitens gehören nur manche KI-Methoden zum sog. »*Machine Learning*«. Deshalb ist Künstliche Intelligenz eben nicht in jedem Fall selbstlernend. Streng von den KI-Methoden zu trennen sind »*Big Data*«-Methoden, welche primär darauf abzielen, große und heterogene Datenmengen zu verwalten.[151] KI-Methoden sind zwar nicht selten datengetrieben und müssen daher eine

150 Vgl. *Wong*, S. 2.
151 *Engels/Goecke*, S. 6 ff.

Aufgabe, wie bei Entscheidungsbäumen oder Clusteralgorithmen, erst lernen, sodass große Datenmengen aus Big Data Systemen hierbei hilfreich sind. Es handelt sich aber bei Künstlicher Intelligenz und Big-Data um verschiedene Konzepte.

III. Beispiel: (Künstliches) neuronales Netz

Aufgrund der kontinuierlich anwachsenden Bedeutung des autonomen Fahrens, der Sprachverarbeitung, der Bilderkennung oder des automatischen Handels mit Wertpapieren soll nachfolgend jene KI-Methode ausführlicher dargestellt werden, die diesen Trend maßgebend ermöglicht hat: Die (künstlichen) neuronalen Netze.[152]

1. Modelldarstellung

Vereinfacht kann ein neuronales Netz näherungsweise mit einem Thermostat respektive Drehregler an der heimischen Heizung verglichen werden (Abbildung 6).[153] Je weiter der Regler in eine Richtung gedreht wird, desto wärmer wird ein Raum und je weiter er in die andere Richtung gedreht wird, desto kälter wird der Raum. Jede nennenswerte Abweichung der Zimmertemperatur vom Komfortwert wird der Bediener in Form von Schwitzen bzw. Frösteln erfahren. Bereits ein Kind ist in der Lage, den Zusammenhang zwischen der Stellung des Thermostats und der Zimmertemperatur zu ermitteln. Nach einer kurzen Zeit des Probierens ist die passende Stellung ermittelt worden und wird fortan beibehalten (sog. »Training«).

152 Vgl. *Ehinger/Stiemerling,* CR 2018, 761.
153 Vgl. *Ashby,* Cybernetica 1 (1958), 83, 91.

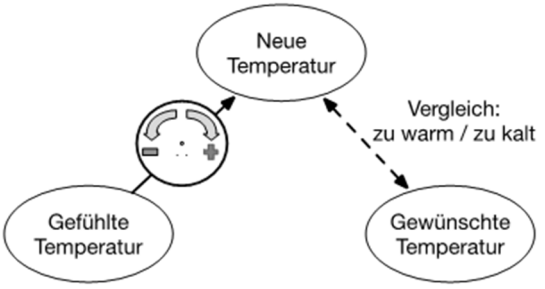

Abbildung 6: Das Einstellen eines Thermostats als Analogie für den Lernprozess (künstlicher) neuronaler Netze

Auf genau diese Weise lernt auch ein neuronales Netz. Statt der gefühlten Temperatur, welche in Form einer einzigen Zahl beschrieben wird, kann ein neuronales Netz viele Werte aufnehmen.[154] Gleichsam kann statt eines einzigen Ergebnisses (die neue Temperatur) auch eine Vielzahl von Werten resultieren.[155] Zudem wird eine enorme Menge an virtuellen Thermostaten verwendet (in der Informatik werden diese als Gewichte bezeichnet), welche den Input mit dem Output über Multiplikationen und Additionen zu einem Netz verflechten. In Abbildung 7 ist ein einfaches neuronales Netz abgebildet, welches aus neun Neuronen bzw. Knoten (N1-N9), davon vier Inputneuronen, zwei Zwischenneuronen und drei Outputneuronen besteht, die über insgesamt 14 gewichtete Kanten (Linien) verbunden sind.

154 Die Gesamtheit aller Zahlenwerte, die in das Netz geleitet werden, werden Inputdaten genannt.

155 Die Gesamtheit aller Zahlenwerte, die aus dem Netz geleitet werden, werden Outputdaten genannt.

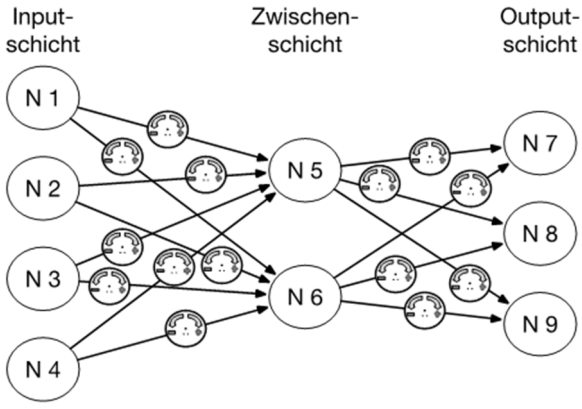

Abbildung 7: Schematischer Aufbau eines einfachen neuronalen Netzes

Sobald die Inputwerte bekannt sind, kann über die Gewichte der Kanten zunächst der Wert der Zwischenneuronen und anschließend der Wert der Outputneuronen berechnet werden. Welche (reale) Information letztlich durch die Neuronen repräsentiert wird, ist nicht relevant. Es kann sich um ein Sensorsignal, eine Tonfolge oder ein Bild[156] handeln. Gleichwohl kann der Output unterschiedliche Bedeutung haben, beispielsweise ein Bild, eine Kategorie[157] oder – wie beim autonomen Fahren – Signale an die Lenkung, den Motor und die Bremse. Mit steigender Anzahl an Neuronen und Schichten ist das Netz in der Lage, immer kompliziertere Zusammenhänge zwischen dem Input und dem Output abzubilden.

2. Trainingsphase

Wird ein neuronales Netz erstellt, sind die Gewichte der Knoten zu Beginn nicht bekannt und werden daher mit zufälligen Werten[158] belegt; es handelt sich um ein untrainiertes, neuronales Netz. Bezugnehmend auf die Analogie

156 Ein Netz, welches HD-Bilder in Farbe verarbeiten kann, benötigt beispielsweise drei Neuronen für jedes Pixel und somit schon 1920*1080*3 = 6.220.800 Inputneuronen.

157 Beim Klassifizieren wird üblicherweise jedem Neuron eine Klasse zugeordnet. In einem neuronalen Netz zur Bilderkennung ist beispielsweise je ein Neuron einer Tierart zugeordnet.

158 Zu der Problematik von Zufall in Computersystemen, s. u., Randomisierte Algorithmen und Pseudozufall (S. 54).

des Heizungsthermostats sind zu Beginn sämtliche Reglerstellungen komplett zufällig gewählt worden. Daraus folgt, dass mit einem untrainierten Netz noch keine Aufgaben erfüllt werden können. Damit ein neuronales Netz eine Aufgabe ausführen kann, muss es diese zunächst erlernen (sog. »Trainingsphase«). Ziel ist es, eine geeignete Stellung für jeden Regler in dem neuronalen Netz zu finden, der bei einem gegebenen Inputwert zu dem gewünschten Outputwert führt. Dies erfolgt im einfachsten Fall mit gelabelten Trainingsdaten.[159] Solche Trainingsdaten bestehen aus einem vordefinierten Input und einem Output. Bei sehr einfachen selbstfahrenden Autos kann der Input beispielsweise aus den Kamerabildern, der Geschwindigkeit, der Beschleunigung und den GPS-Koordinaten bestehen. Als Output dient der Lenkradeinschlag, das Brems- und das Motorsignal. Der Optimierungsalgorithmus versucht nun die Gewichte in dem neuronalen Netz so anzupassen, dass der aus dem Input berechnete Output möglichst dem tatsächlich gemessenen Output entspricht. Übertragen auf das Thermostatbeispiel bedeutet dies, dass der Optimierungsalgorithmus anhand von bekannten Sollausgabewerten (im Beispiel die Zieltemperaturen) neue Gewichte berechnet und entsprechend verändert, damit das Resultat ein bisschen besser ausfällt. Falls sich bei nachfolgenden Daten herausstellt, dass sich der Unterschied verringert hat, wird der Optimierungsalgorithmus versuchen, die Gewichte weiter zu verstärken. Dieser Prozess wiederholt sich so lange, bis der Optimierungsalgorithmus keine weitere Verbesserung erreichen kann. Die so gefundenen Gewichtswerte werden nach dem Training nicht mehr verändert und anschließend abgespeichert.[160]

Nachdem ein Netz trainiert wurde, ist es in der Lage, einem verhältnismäßig hohen Anteil an Inputwerten aus den Trainingsdaten die richtigen Outputwerte zuzuordnen. Die Leistungsfähigkeit wird anschließend im Rahmen der Evaluation ermittelt. Hierbei wird geprüft, ob auch Inputwerte, mit denen das Netz nicht trainieren konnte, zu korrekten Outputwerten führen. Dieser Schritt ist notwendig, da immer die Gefahr besteht, dass das Netz nicht die Zusammenhänge, sondern lediglich das Trainingsset »auswendig« lernt (sog. »Overfitting«).[161] In diesem Fall könnte es zwar die Trainingsdaten korrekt klassifizieren, es würde allerdings an neuen, unbekannten Inputs scheitern. Um dies zu verhindern, wird üblicherweise nicht mit dem gesamten Trainingsset trainiert, sondern es

159 Alternativ kann in manchen Anwendungsfällen auch aus ungelabelten Daten gelernt werden. Dies wird grundsätzlich angestrebt, da das Labeln nur manuell durchgeführt werden kann und verhältnismäßig zeitintensiv ist.

160 Vgl. *Ertel*, S. 291 ff.

161 *Hagan/Demuth/Beale/De Jesús*, Kap. 13, S. 3.

wird eine kleine Menge von Daten zurückbehalten, um anschließend damit das KI-System zu evaluieren. Erst wenn das neuronale Netz auch mit diesen Daten einen ausreichend hohen Anteil korrekt berechnet, kann es genutzt werden.[162]

3. Anwendungsphase

Bei der Nutzung des neuronalen Netzes werden die jeweiligen durch das KI-System erfassten Inputdaten an das Netz geleitet, welches die Outputwerte berechnet. Diese werden anschließend genutzt, um entsprechende Steuersignale zu generieren (sog. »Anwendungsphase« oder »Inferencing«). Im Gegensatz zur Trainingsphase finden hier keine zufälligen Werte mehr Verwendung, sodass Zufall keinen Einfluss auf das Verhalten des KI-Systems hat.[163] Die für den Outputwert relevanten Gewichte werden nicht mehr geändert. Daraus folgt, dass ein identischer Inputwert stets zu denselben Outputwerten führt.

4. Sonderfall: Weiterlernende Systeme

Einen gewissen Sonderfall stellen weiterlernende Systeme dar. Vorab ist anzumerken, dass die soeben genannten Phasen, bestehend aus Trainingsphase und Anwendungsphase auf einem neuronalen Netz nicht zeitgleich ablaufen können. Entweder werden die gefundenen Gewichte zur Berechnung eines gegebenen Inputs (Anwendungsphase) genutzt oder es werden bessere Gewichte gesucht (Trainingsphase). Das heißt aber nicht, dass weiterlernende Systeme nicht möglich sind. Bei einem nicht weiterlernenden, starren System (sog. »Batch Learning«) läuft die Trainingsphase nur einmal ab. Die mit dem dabei verwendeten Datensatz (sog. »Batch«) ermittelten Gewichte werden anschließend gespeichert und im Rahmen der Anwendungsphase genutzt. Eine spätere Änderung der Gewichte nach Beginn der Anwendungsphase findet nicht mehr statt.[164] Der Vorteil eines solchen Vorgehens liegt in einem hohen Maß an Kontrolle für den

162 Vgl. *Hagan/Demuth/Beale/De Jesús*, Kap. 22, S. 27 ff.

163 Als Ausnahme von diesem Grundsatz sind hier allenfalls die sog. »Generative Adversarial Networks« (GAN) zu nennen, wobei auch bei diesen nicht die Gewichte selbst verändert werden, sondern aus einem pseudozufälligen Input-Seed ein Output generiert wird.

164 *Kriesel*, S. 54.

Hersteller oder Entwickler über sein neuronales Netz. Ein einmalig angelerntes neuronales Netz wird sein Verhalten in der gleichen Situation – also bei der gleichen Eingabe – während der späteren Anwendung nicht ändern.

Bei einem weiterlernenden System (sog. »Online Learning«) lernt das KI-System während der späteren Anwendung weiter, womit sich die Kantengewichte auch nach der ersten Trainingsphase ändern (können). Dennoch findet dies nicht während der Anwendungsphase statt, vielmehr sind auch hier – wie zu Beginn genannt – die Lern- und Anwendungsphasen zu trennen. Entweder wechseln sie sich (in einer Endlosschleife) ab oder sie werden parallel, aber mit zunächst getrennten künstlich neuronalen Netzen (in sog. »Threads«) ausgeführt.[165] Die Inputs (Dateneingaben) des neuronalen Netzes in der Anwendungsphase werden protokolliert und für zukünftige Trainingsphasen verwendet. Sobald die Trainingsphase zu einer besseren Gewichtskombination geführt hat, werden diese neuen berechneten Gewichte des neuronalen Netzes nach der (vorherigen) Trainingsphase zwischengespeichert und regelmäßig in das neuronale Netz der Anwendungsphase übernommen. Damit ist das KI-System in der Lage, sich an neue Situationen anzupassen und sich kontinuierlich zu verbessern.[166] Problematisch ist jedoch, dass der Hersteller bzw. Entwickler weniger Kontrolle über das neuronale Netz hat, da sich das Verhalten auch ändern kann, wenn das neuronale Netz den Einflussbereich des Herstellers oder Entwicklers verlassen hat. Denn während ein Batch Learning System nach der Trainingsphase bis zu einem gewissen Grad getestet und verifiziert werden kann, ist dies beim Online Learning schwer bis unmöglich.[167]

Als anschauliches Beispiel dient der von Microsoft entwickelte Chatbot *Tay*. Dieser sollte von der Interaktion mit seinen Nutzern lernen und so seine Fähigkeit, zu kommunizieren, kontinuierlich verbessern. Ein solches lernendes System kann aber das Erlernte nicht selbstständig evaluieren und ist daher auf das Feedback seiner Umgebung angewiesen. Im Falle von *Tay* bestand dieses Umfeld jedoch aus seinen Nutzern, welche dem System allerdings rassistische und antisemitische Aussagen vorgaben. Daher wandelte sich *Tay* von einem einfachen Chatbot in einen nationalsozialistischen Parolen-Generator.[168]

165 Vgl. *Trobec/Slivnik/Bulic/Robic*, S. 50; *Ben-Nun/Hoefler*, S. 1:8 ff., 1:22 ff.
166 *Sahoo/Pham/Lu/Hoi*, S. 1 ff.
167 *Sahoo/Pham/Lu/Hoi*, S. 1.
168 *Wolf/Miller/Grodzinsky,* ACM SIGCAS Computers & Society, Volume 47 No. 3 (2017), 54 ff.

Während das Problem des Online Learnings im Falle eines Chatbots noch als einfaches Marketingdebakel verbucht werden kann, stellt es bei sicherheitskritischen Systemen ein fundamentales Dilemma dar. Soll beispielsweise ein autonom fahrendes Fahrzeug kontinuierlich lernen, sicherer zu navigieren, indem es aus vergangenen Fahrsituationen lernt, wäre ein systematischer Fehler, wie bei *Tay*, nicht akzeptabel. Eine Lösung zwischen Batch Learning und Online Learning stellen wiederkehrende Updates dar. Hierbei werden während der Laufzeit Daten erfasst und das Trainingsset kontinuierlich erweitert, was wiederum die Qualität des Netzes begünstigt. Nach einer umfassenden Evaluation durch den Entwickler bzw. Hersteller des neuen neuronalen Netzes bzw. einer Freigabe eines Qualitätssicherungsprozesses kann dies – als Update – an die Endgeräte bereitgestellt werden (sog. »Rollout«).[169]

5. Black-Box Konzept

Neuronale Netze werden oft als »Black-Box« bezeichnet.[170] Wie bereits deutlich wurde, sind die Rechenschritte, mit denen ein neuronales Netz einen Output aus einem Input berechnet, nicht unbekannt. Ein neuronales Netz ist somit keine »Black-Box« bezogen auf die informationsverarbeitenden Prozesse, welche im neuronalen Netz ablaufen.[171] Der Black-Box Gedanke bezieht sich nur auf die semantische Beziehung zwischen den Informationen, die in das Netz eingebracht werden und der Antwort, die man erhält. Analog zu einem neuronalen Netz verhält es sich auch mit einem Kind, das einen Heizkörper zu bedienen gelernt hat. So hat es zwar die Beziehung verinnerlicht und kann diesen Zusammenhang im Alltag nutzen, um für jeden Raum die gewünschte Temperatur zu finden. Es bedarf aber keiner Kenntnis über die Funktionsweise einer Zentralheizung, über Thermodynamik, Mechanik oder Strömungslehre. Der tatsächliche Zusammenhang bleibt dem Kind nach wie vor als »Black-Box« verborgen. Das Problem der Black-Box betrifft allerdings nicht nur neuronale Netze, sondern kann prinzipiell auch jede andere KI-Methode, eigentlich gar jede

169 Freilich kann dies mit datenschutzrechtlichen Problemen verbunden sein, wenn während der Laufzeit beim Anwender erhobene Daten an den Hersteller zwecks Verbesserung übertragen werden, vgl. Problematik bei Kraftfahrzeugen, *Lüdemann*, ZD 2015, 247 ff.

170 Vgl. *Hagan/Demuth/Beale/De Jesús*, Kap. 22, 3.; *Hoeren/Niehoff,* RW 2018, 47, 50.

171 Ebenso *Pasluosta/Chiu*, in: *Cartwright*, S. 183 ff.

Computersoftware, betreffen. Ein Entscheidungsbaum kann, wie der im vorherigen Abschnitt vorgestellte,[172] grundsätzlich durch den Menschen nachverfolgt werden. Wird dieser Entscheidungsbaum allerdings zu komplex – mit Tausenden von Parametern – so beginnt die (menschliche) Nachvollziehbarkeit der Ergebnisse ebenfalls schnell zu schwinden.[173]

IV. Multi-Agenten Systeme

Obgleich jede der zuvor gezeigten KI-Methoden in bestimmten Problembereichen ein hohes Leistungsniveau erreichen kann, so ist jede dieser KI-Methoden nach wie vor auf bestimmte Problemtypen spezialisiert. Beispielsweise lässt sich mit einem neuronalen Netz ein Programm zur Objekterkennung gestalten, für das Beantworten von Quizfragen ist diese KI-Methode jedoch ungeeignet. Umgekehrt kann eine Semantic Web Applikation komplexe logische Fragestellungen beantworten, um diese in menschlicher Sprache formulierten Fragen jedoch für die Maschine zu übersetzen, bedarf es wiederum einer anderen Künstlichen Intelligenz wie ein neuronales Netz. Das zeigt, dass zur Lösung einer komplexen Aufgabe – wie das autonome Steuern eines Kraftfahrzeugs[174] oder wie Unterstützungstätigkeiten im Gesundheitswesen[175] – zahlreiche Einzelprobleme angegangen werden müssen, für welche wiederum unterschiedliche KI-Methoden geeignet sind. Hierbei ist jede implementierte KI-Methode als ein eigenständiges Element und damit als eigenständiger Agent zu betrachten. In der Gesamtheit bilden diese Agenten sog. »Multi-Agenten Systeme«, die Vorteile unterschiedlicher Methoden kombinieren.[176]

D. Vom Modell über den Algorithmus zur Computersoftware

Genauso wenig wie sich die Skizze eines Eisenbahnfahrzeugs als Fortbewegungsmittel eignet, ist das Modell eines neuronalen Netzes konkrete Computersoftware. Allen KI-Methoden ist gemeinsam, dass sie zunächst nur ein Modell beschreiben, mit welchem sich Informationen in einer bestimmten Weise abstrakt verarbeiten lassen. Aus diesem Modell wird ein

172 S. o., Entscheidungsbäume (S. 35).
173 Vgl. *Arrieta/Dias-Rodríguez/Del Ser et al.*, S. 15.
174 *Dafflon/Vilca/Gechter/Adouane*, Procedia Computer Science, 51 (2015), 423 ff.
175 Vgl. *Isern/Moreno,* J Med Syst (2016), 40:43, 6.
176 Vgl. *Alkhateeb/Al Maghayreh/Aljawarneh*, in: ICISMS, S. 75.

Algorithmus abgeleitet, der schließlich in einer gewählten, mehr oder weniger hohen Programmiersprache umgesetzt wird, um – nach evtl. Kompilierung – ein auf einem Computer lauffähiges Programm zu erhalten.[177] Insofern ist der Begriff des Algorithmus sowie dessen Beschränkungen für die Untersuchung des Wesens der Künstlichen Intelligenz von besonderer Bedeutung.[178] Es muss geklärt werden, was ein Algorithmus überhaupt ist und was ihn von Computersoftware unterscheidet. Zudem spielt in vielen Bereichen der Künstlichen Intelligenz der Zufall eine Rolle, sodass eine besondere Art von Algorithmen, die so genannten »randomisierten Algorithmen«, genauer betrachtet werden müssen.

I. Allgemeiner Algorithmus-Begriff

Der Begriff »Algorithmus« wird – wie auch der Begriff »Algebra« – ursprünglich vom Namen des Werks »*dixit algrismi*«, zu Deutsch etwa »So sprach Khowarizmi«, abgeleitet.[179] »*Khowarizmi*« ist der Name des ursprünglichen arabischen Autors, womit »Algorithmus« ein Eigenname ohne eigenständige Bedeutung ist. Er wird seit jeher zur Beschreibung von Vorschriften verwendet, die sich nach vorgegebenen Schritten abarbeiten lassen. Der Begriff Algorithmus wird in diesem Sinne nicht nur für Rechenvorschriften für Computer angewendet,[180] sondern auch für Vorschriften zur Befolgung allgemeiner vorgegebener Regeln, beispielsweise in der Notfallmedizin;[181] auch ein Kochrezept kann als Algorithmus (im weiteren Sinne) verstanden werden.[182] Problematisch ist, dass der Begriff keine einheitliche Definition, auch nicht in der Informatik, gefunden hat.[183] Ein Standardwerk der Informatik beschreibt den Algorithmus »grob gesprochen [als] wohldefinierte Rechenvorschrift, die eine […] Eingabe verwendet und eine […] Ausgabe erzeugt«.[184] Das heißt, für Algorithmen im Bereich von Computern sind drei wesentliche Elemente notwendig: Die Eingabe, die Ausgabe und eine wohldefinierte Rechenvorschrift.[185] Die Eingabe ist eine

177 Vgl. *Aho/Lam/Sethi/Ullman*, S. 3 ff.
178 *Hoeren/Niehoff,* RW 2018, 47, 49.
179 *Klaeren/Sperber*, S. 246.
180 Vgl. *Klaeren/Sperber*, S. 246 ff.; *Broy*, S. 31; so wohl auch *Söbbing*, CR 2020, 223, 224.
181 Vgl. *Peters/Runggaldier*, S. 1 ff.
182 *Goos*, S. 22 ff.
183 *Güting*, S. 1.
184 *Cormen/Leiserson/Rivest/Stein*, S. 5.
185 Im Ergebnis auch bei *Marly*, Rn. 29.

Menge von Daten, mit welcher die Rechenvorschrift vorher definierte Schritte abarbeitet und eine Ausgabe produziert, die selbst wieder eine Menge von Daten ist.[186] Weitere Voraussetzungen an den Algorithmus, die teilweise diskutiert werden, bestehen – wie im Folgenden noch dargelegt wird – nicht.[187]

Bei den oben als Beispiel genannten neuronalen Netzen ist zwischen dem Anlern- bzw. Trainingsalgorithmus und der konkreten Anwendung des künstlichen neuronalen Netzes selbst zu unterscheiden. Beim Anlern- bzw. Trainingsalgorithmus stellen die zufällig gewählten Kantengewichte die Eingaben dar und die späteren (endgültigen) Kantengewichte sind dessen Ausgabe; der Algorithmus wird somit für die Trainingsphase verwendet. Während der Nutzung des trainierten neuronalen Netzes in der Anwendungsphase bestimmt sich die Ausgabe nach den während des Trainings ermittelten Kantengewichten. Diese sind folglich nach dem Training nur noch Konstanten des späteren Programms und damit unveränderlicher Bestandteil. Der Sonderfall des Online Learnings ändert diesen Prozess nicht, denn beim Lernen während des Einsatzes werden lediglich der Trainingsalgorithmus und der Anwendungsalgorithmus abwechselnd ausgeführt.[188]

II. Randomisierte Algorithmen und Pseudozufall

Eine besondere Art der Algorithmen sind sog. »randomisierte Algorithmen«.[189] Hierbei werden in der Rechenvorschrift Zufallszahlen verwendet, um eine zufällige Entscheidung zu treffen (z.B. in der Trainingsphase von künstlichen neuronalen Netzen). Der Vorteil gegenüber nicht randomisierten Algorithmen ist, dass für bestimmte Probleme eine mögliche Lösung schneller (und damit effizienter) gefunden werden kann. Dies ist aber mit dem Nachteil verbunden, dass das Ergebnis evtl. nur näherungsweise korrekt ist (sog. »Monte-Carlo-Algorithmen«) oder nicht bekannt ist, ob der Algorithmus terminiert[190] (sog. »Las-Vegas-Algorithmen«).[191] Die

186 *Cormen/Leiserson/Rivest/Stein*, S. 5 ff.
187 *Güting*, S. 28; so aber *Söbbing*/*Söbbing*, S. 8.
188 S. o., Sonderfall: Weiterlernende Systeme (S. 49).
189 Vgl. *Motwani,* ACM Computer Surveys 8 (1996), 33 ff.
190 S. u. zum Begriff »Terminierung«, Determinismus, Determiniertheit und Terminierung in der Informatik (S. 59).
191 Vgl. *Karp*, Discrete Applied Mathematics 34 (1991), 165 ff.

theoretische Informatik nimmt bei der Betrachtung randomisierter Algorithmen den Zufall als gegeben an. Allerdings kann ein Computer von selbst keinen echten Zufall erzeugen. Entweder ist er Bestandteil der Eingabe des Algorithmus, also von einer externen Quelle,[192] oder nur sogenannter »Pseudozufall«. Dieser wird von Pseudozufallsgeneratoren erzeugt, die selbst deterministisch,[193] also vorhersehbar sind.[194] Als Eingabe erhalten diese Generatoren einen sogenannten »*seed*«, zu welchem sie (vorhersehbare) Zahlen ausgeben, die aber aus stochastischer Sicht den Anforderungen an Zufall genügen.[195] Dies gilt natürlich nur dann, wenn der *seed* bekannt ist. In der Praxis kann man neben der oben genannten externen Zuführung von echt zufälligen Werten auch Werte nehmen, die einer ständigen Änderung unterworfen sind (z.B. die aktuelle Uhrzeit). Diese ändern sich permanent und sind nicht wiederkehrend. Das bedeutet, dass nach wie vor vorhergesagt werden könnte, welche Zahlen ausgegeben werden, wenn der *seed* bekannt ist. Dennoch erzeugt jede (zeitlich unterschiedliche) Ausführung neue – stochastisch hinreichend zufällige – (Pseudo)Zufallszahlen.[196]

III. Vom Algorithmus zur Computersoftware

Ist der Algorithmus bekannt, bleibt die Frage, was ihn von einer Computersoftware bzw. einem Computerprogramm unterscheidet. Allgemein umschrieben ist ein Computerprogramm eine Folge von Anweisungen, die von einem Computer verstanden wird und – bei vorhandener Eingabe – automatisiert abgearbeitet werden kann.[197] Es ist somit die konkrete Implementierung eines Algorithmus oder einer Folge von Algorithmen, die miteinander interagieren[198] und selbst wieder einen (komplexeren) Algorithmus darstellen.[199] Einen bestehenden Algorithmus kann man aber in

192 Beispielsweise atmosphärisches Rauschen oder zufällige Nutzereingaben.
193 S. u. zum Begriff »Determinismus« in der Informatik, Determinismus, Determiniertheit und Terminierung in der Informatik (S. 59).
194 *Karpfinger/Kiechle*, S. 25.
195 Vgl. *Karloff,* Journal of the Association for Computing Machinery 40 (1993), 454 ff.
196 *Kocarev/Jakimoski/Tasev,* LNCIS 292 (2003), 247 ff.
197 Ähnlich auch § 1 (i) der WIPO Mustervorschriften für den Schutz von Computersoftware = GRUR Int. 1978, 286, 290.
198 Vgl. *Kindermann,* ZUM 1985, 2, 5 f.
199 *Haberstumpf*, in: *Lehmann*, Rn. 25.

verschiedener Weise implementieren,[200] der Algorithmus gibt die Art nicht vor.[201] Daraus folgt aber nicht, dass jedes Computerprogramm ein Algorithmus ist,[202] sondern nur, dass die Gesamtheit von Anweisungen als Algorithmus gesehen werden kann. Zusätzlich enthalten Computerprogramme aber noch die Definition von Datenstrukturen sowie weitere statische Daten, die zum Ablauf notwendig sind (z.B. Konstanten).[203] Auch der Umkehrschluss, dass jeder Algorithmus ein Computerprogramm ist, gilt nicht, da Algorithmen eben nicht immer automatisiert ausgeführt werden können. Einerseits sind – wie dargelegt – nicht alle Algorithmen für Computer bzw. Maschinen entworfen, andererseits können Anweisungen teilweise noch so abstrakt sein, dass eine Konkretisierung im Rahmen der konkreten Implementierung notwendig ist oder gar gewisse Anweisungen mit spezifischeren Algorithmen ersetzt werden müssen.[204] Beispielsweise kann in einem Algorithmus festgelegt werden, dass eine Liste sortiert werden muss, der konkrete Sortier-Algorithmus bleibt offen und der Wahl des Entwicklers bzw. Programmierers überlassen.[205] Freilich können in einem Algorithmus, der ein berechenbares Problem löst, alle abstrakt gehaltenen Stellen durch weitere (konkretere) Algorithmen oder sonstige relevante Daten ersetzt werden und zwar so lange, bis der Algorithmus konkret genug ist, um auf einer Maschine ausgeführt werden zu können. Dieser Prozess ist die sog. Implementierung, bei welcher Auswahlmöglichkeiten bestehen. Dann hat man keinen abstrakten Algorithmus

200 Hiermit ist nicht die Wahl einer Programmiersprache gemeint, selbst in Maschinensprache – die praktisch von niemandem als Implementierungslösung herangezogen wird – sind mehrere Möglichkeiten der Implementierung denkbar.

201 Insbesondere ist auch der sog. »Pseudocode«, mit welchem Algorithmen dargestellt werden können, nicht maschinenausführbar, *Cormen/Leiserson/Rivest/Stein*, S. 17 f. Daher ist er weniger nah am maschinenausführbaren Programmcode als die Darstellung zunächst vermuten lässt, so wohl *Söbbing*, CR 2020, 223, 226.

202 *Schricker/Loewenheim/Spindler*, § 69a UrhG Rn. 12; *Dreier/Schulze/Dreier*, § 69a UrhG Rn. 22; *Marly*, Rn. 31.

203 *Güting*, S. 1 f.; so wohl auch *Kindermann,* ZUM 1985, 2, 6.

204 Vgl. *Sommerville*, S. 106 ff.

205 Bei Sortieralgorithmen – die ein typisches Problem in der theoretischen Informatik darstellen – bestehen erhebliche Unterschiede in der Laufzeit. vgl. *Cormen/Leiserson/Rivest/Stein*, S. 147 ff.

mehr, sondern ein (konkretes) Computerprogramm.[206] Somit sind Algorithmen nur das abstrakte – aber hinreichend genaue – Konzept hinter einem Programm.[207]

IV. Folgen für das Wesen der Künstlichen Intelligenz

Ein Modell an sich ist nicht in der Lage, ein konkretes Problem zu lösen, sondern ist nur ein Ansatz für eine Problemlösung. Insoweit sind auch die bisher vorgestellten KI-Methoden nur Modelle zur Lösung von abstrakten Problemen, reichen aber in ihrer Beschreibung nicht aus, um ein konkretes Problem zu lösen. Genauer sind hier Algorithmen, die – auf Computer bezogen – als Rechenvorschriften aus einer Eingabe eine bestimmte Ausgabe erzeugen. In den verschiedenen KI-Methoden kommen bestimmte Algorithmen vor, die allerdings für spezifische Probleme entworfen werden müssen, beispielsweise Trainingsalgorithmen für neuronale Netze. Für die reale Anwendung müssen die KI-Methoden und damit auch die Algorithmen allerdings in Computersoftware implementiert werden. Hierbei gibt das System »Computer« die Restriktionen vor. Daher ist jede real existierende[208] und damit implementierte Künstliche Intelligenz den gleichen Restriktionen unterworfen wie jeder (andere) Algorithmus und damit jedes andere Computerprogramm. Dies gilt insbesondere bei der Verwendung von Zufall, der entweder Eingabe sein muss oder nur Pseudozufall ist.

206 In diesem Sinne sind wohl auch die Definitionen von *Loewenheim/Spindler*, *Dreier* und *Marly* (alle Fn. 202) zu verstehen, die damit einen zu engen Begriff zugrunde legen.

207 *Güting,* S. 28; *Manber*, S. 1; *Wirth*, S. 7; so wohl auch *Haberstumpf*, in: *Lehmann*, Rn. 23 ff.

208 Da auch Konzepte und Methoden Entitäten sind, heißt »real existierend« an dieser Stelle, dass die Künstliche Intelligenz in der Lage ist, durch Verhalten eine Zustandsänderung in der realen bzw. physischen Welt zu bewirken, ergo so konkret ist, dass sie als Computersoftware auf irgendeinem Computer ablaufen kann.

E. Determinismus

Wie festgestellt, müssen KI-Methoden für eine reale Anwendung in Computerprogramme implementiert werden, die auf Algorithmen basieren. Somit kommt die Frage auf, ob Computerprogramme, damit auch Algorithmen – und letztendlich auch Künstliche Intelligenzen – (immer) deterministisch sind. »Determinismus«, vom lateinischen »*determinare*«, zu Deutsch »abgrenzen«,[209] ist ein Homonym, welches einerseits, philosophisch benutzt wird, andererseits auch in der Informatik angewendet wird. Bei der juristischen Wertung von KI-Systemen treffen beide Konzepte unter dem gleichen Begriff aufeinander, sodass differenziert werden muss.

I. Determinismus und Willensfreiheit in der Philosophie und im Recht

Im philosophischen Sinne bedeutet Determinismus, dass zukünftige Ereignisse nicht zufällig geschehen, sondern vorherbestimmt sind. Entweder durch einen Gott (sog. »theologischer Determinismus«) oder als Kausalfolge des bereits Geschehenen (sog. »kausaler Determinismus«).[210] Sofern dies für alle zukünftigen Ereignisse gilt, heißt dies, dass es keinen Zufall gibt, daher auch keine freien Entscheidungen von Subjekten. In einer deterministischen Welt wäre der Wille nicht frei, sondern die zukünftigen Handlungen stünden bereits fest. Somit ist die Frage nach der Willensfreiheit mit der des Determinismus verbunden.[211] Das Resultat dieses Gedankengangs wurde – als Folge der bisherigen Ergebnisse der Neurowissenschaften – vor allem im Strafrecht diskutiert. Sofern der Wille nicht frei ist, kann es auch keine Schuld geben.[212] Im zivilrechtlichen Schrifttum wurde diese Problematik seltener diskutiert, spielt aber bei Fragen der Haftung und der Privatautonomie eine Rolle,[213] worauf noch zurückzukommen ist.

209 *Mittelstraß/Mainzer*: Determinismus, Bd. 2, S. 167.
210 *Sandkühler/Stekeler-Weithofer*: Determinismus/Indeterminismus, Bd. 1, S. 382.
211 *Mittelstraß/Mainzer*: Determinismus, Bd. 2, S. 168.
212 Vgl. *Czerner,* ArchKrim 218 (2006), 65 ff. m.w.N.; kritisch *Stübinger*, S. 341 ff.
213 *Laufs,* MedR 2011, 1, 4 ff.; vgl. auch *Waldstein*, in: FS Schwind, 1978, 329 ff.

II. Determinismus, Determiniertheit und Terminierung in der Informatik

Auch in der Informatik findet sich der Begriff »Determinismus«; zusätzlich bestehen zwei weitere verwandte Begriffe, die für das Verständnis von Relevanz sind: »Determiniertheit« und »Terminierung«. Der Begriff »Determinismus« kommt vor allem bei zwei Teildisziplinen vor: Der Automatentheorie und der Kryptologie. In beiden Disziplinen wird der Begriff mit unterschiedlichem Hintergrund verwendet, beschreibt aber dasselbe Phänomen. Grundsätzlich bedeutet Determinismus, dass ein (implementierter) Algorithmus[214] stets dieselben Schritte und Zwischenergebnisse durchläuft, egal wann und wie oft er ausgeführt wird.[215] Analog einem Autofahrer, der sein Ziel immer über dieselbe Strecke erreicht. Umleitungen sind ihm ebenso fremd wie alternative Routen. Im Umkehrschluss wäre ein nicht-deterministischer Algorithmus, ein solcher, der eine Vielzahl von unterschiedlichen Schrittfolgen durchläuft. Der konkrete Zustand und damit auch Zwischenergebnisse hängen dabei nicht nur von der Eingabe, sondern vom Zufall ab. Entsprechend würde einem nicht-deterministischen Autofahrer das gesamte Straßennetz zur Verfügung stehen, um zu irgendeinem Ziel zu gelangen, sodass unterschiedliche Strecken denkbar sind. Allerdings handelt es sich bei den nicht-deterministischen Algorithmen um eine theoretische Betrachtung, bei welcher Zufall in Computern als gegeben angenommen wird. Tatsächlich handelt es sich bei diesen Algorithmen um die bereits erwähnten »randomisierten Algorithmen«, die entweder keinen echten Zufall, sondern – deterministischen – Pseudozufall verwenden oder echten Zufall als Eingabe. Gleiches gilt für stochastische Entscheidungen in Algorithmen, da diese bei stets gleicher Eingabe die gleiche Ausgabe erzeugen.

Neben dem Determinismus-Begriff existiert in der Informatik auch das Konzept der Determiniertheit. Ein Algorithmus ist determiniert, wenn er bei einem definierten Eingabewert stets zum selben Ergebnis führt.[216] Bezogen auf den Autofahrer bedeutet das, dass er stets zu demselben Ort fährt, aber die Route keine Rolle spielt. Bei einem nicht-determinierten Algorithmus sind hingegen der Weg und das Ergebnis ungewiss. Daraus folgt, dass jeder Algorithmus und damit auch jedes Programm bzw. jede Software, die deterministisch ist, auch determiniert ist.

214 Genau genommen ein Automat, der immer dieselben Zustandsübergänge durchläuft.

215 *Hromkovič*, S. 108 ff.

216 *Priese/Erk*, S. 68.

Der letzte Begriff ist die Terminierung eines Algorithmus, also dass dieser zu einem Ende kommt.[217] Es existieren zahlreiche Algorithmen, welche ohne einen Eingriff des Benutzers nie beendet werden. Es ist relativ einfach, ein Programm zu entwerfen, welches in eine Endlosschleife gerät, sich also in einem sich ständig wiederholenden – niemals endenden – Zustandszyklus befindet und somit auf einem nicht-terminierten Algorithmus basiert.[218] Es gibt aber keinen Algorithmus, der für alle (formalen) Algorithmen und damit für alle Programme erkennen kann, ob sie terminieren. Dies wird auch als »Halteproblem« bezeichnet.[219] Sofern ein reales Programm aber terminiert, ist es den vorherigen Ausführungen nach auch determiniert und deterministisch, sowohl im informatischen als auch im philosophischen Sinne. Aus diesem Grund kann es keine realen, durch Programme und damit in Software implementierten, nicht-deterministischen Algorithmen geben, die terminieren. Da den Darlegungen nach auch jede implementierte Künstliche Intelligenz Computersoftware ist, gilt die hier beschriebene Folgerung uneingeschränkt auch für jede implementierte Künstliche Intelligenz,[220] konkreter: Jede Künstliche Intelligenz ist deterministisch.[221]

III. Konvergierender bzw. stochastischer Determinismus

Alle real existierenden Programme müssen auf realen Maschinen – also auf einem Computer – ausgeführt werden und unterliegen damit den Restriktionen dieser Maschinen. Hier kann die zur Verfügung stehende Rechenkapazität nicht ausreichen, um für alle möglichen Eingaben sämtliche möglichen Folgezustände zu berechnen. Entweder aufgrund der begrenzten Speicherkapazität oder durch zu lange Rechenzeiten, wodurch das Problem zu komplex wird. Diese Problematik betrifft jegliche Art von Software, die hinreichend komplex ist, und ist daher unabhängig vom Phänomen der Künstlichen Intelligenz.[222] Allerdings weisen implementierte KI-Methoden eben oft eine entsprechend hohe Komplexität auf, sodass das Problem hier

217 *Hoffmann*, Grenzen der Mathematik, S. 274.
218 Kurzes Beispiel in Python: »while True: print('BGB')«.
219 *Hoffmann*, Grenzen der Mathematik, 319 ff.
220 Ebenso wohl auch *Hoeren/Niehoff,* RW 2018, 47, 49.
221 Ebenso wohl auch *Käde/v. Maltzan,* CR 2020, 66, 67.
222 Auch sind wohl gängige Programme (z.B. Betriebssysteme) für Endbenutzer so komplex, dass auch nicht mehr alle möglichen Eingaben vorherberechnet werden können.

in besonderer Weise zutage tritt. Das betrifft auch die Algorithmen, die beim *Maschine Learning* im Lern- bzw. Trainingsprogramm genutzt werden. Um ein gegebenes Problem ressourceneffizient zu lösen, kann der erwähnte Pseudozufall, aber auch ein extern zugeführter, echter Zufall hilfreich sein.[223] Hierbei wird mit einer zufälligen (schlechten) Lösung begonnen und durch sukzessive kleine Veränderungen das KI-System immer weiter verbessert. Diese KI-Systeme konvergieren üblicherweise gegen eine globale Ideallösung. Der Lern- bzw. Trainingsprozess beginnt dabei aber mit zufällig gewählten Parametern. Nähme man echte Zufallszahlen, wäre der zugrunde liegende Algorithmus zwar gem. der Automatentheorie ein nicht-deterministischer, aber eben nur, weil die Zufallszahlen nicht als Eingabe gewertet werden, was sie in der realen Umsetzung jedoch sind. Verfehlt wäre es deshalb, hierin die Unberechenbarkeit einer Maschine oder gar das Potenzial eines »freien Willen« zu interpretieren. Komplexe, sich selbst optimierende Algorithmen gleichen mehr Bergsteigern, die denselben Gipfel über unterschiedliche Routen erklimmen. Es wird im Rahmen der Optimierung stets gegen das Optimum – den Gipfel – gestrebt. Möglich ist es mitunter, statistische Aussagen über das Verhalten komplexer Programme in vertretbarer Zeit zu erhalten, die entsprechende Tendenzen erkennen lassen.[224] Dies würde man dann stochastischen Determinismus nennen.[225] Für neuronale Netze bedeutet das, dass während der Anwendung von diesen, deren Ergebnis in keiner Weise weniger vorhersehbar ist, als das anderer (komplexer) Programme oder gar eines Taschenrechners, der zwei Zahlen addiert. Trainingssysteme beruhen zwar auf Zufallszahlen, was aber nichts am deterministischen Charakter ändert. Potential für eine freie Entscheidung besteht auch hier in letzter Konsequenz nicht.

223 Vgl. *Henderson*, S. 8.
224 Vgl. *Montavon/Samek/Müller*, Digital Signal Processing 73 (2018), 1 ff.
225 *Russell/Norvig*, S. 70, nennt dieses Verhalten einfach »stochastisch«.

F. Fehlerhaftigkeit, Erklärbarkeit und Verhaltensvorhersehbarkeit

Für einfache Computerprogramme gilt bereits, dass diese nicht fehlerfrei sind.[226] Gerade im Leistungsstörungsrecht bei Computersoftware wurde dieses Problem ausgiebig diskutiert.[227] Wenn sich (implementierte) Künstliche Intelligenz aufgrund ihrer deterministischen Eigenschaft nicht von anderer Computersoftware unterscheidet, gilt die Annahme der Fehlerhaftigkeit erst recht für Künstliche Intelligenz. Allerdings können aufgrund der dargelegten steigenden Komplexität, dem Einfluss des Pseudozufalls und der scheinbar stärker werdenden Autonomie mancher Künstlicher Intelligenzen einerseits weitreichende Folgen resultieren und andererseits mag es wirken, als würde sich Künstliche Intelligenz beliebig verhalten.

Im Rahmen der unterschiedlichen Aufgaben, die mit Künstlicher Intelligenz gelöst werden können, ist ein neuronales Netz zur Klassifikation (sog. »Klassifikator«) am anschaulichsten. Hierbei handelt es sich um eine Künstliche Intelligenz, welche eine Eingabe einer bestimmten Klasse zuordnet, beispielsweise um Objekte auf Bildern zu bestimmen (z.B. Fußgänger oder Objekte beim autonomen Fahren). Die Menge an Variablen innerhalb eines solchen neuronalen Netzes ist für einen Menschen allerdings schwer erfassbar. Beispielsweise hat das Bilderkennungsnetz »NASNET« bereits knapp 90 Millionen Variablen.[228] Diese Netze werden grundsätzlich mit Trainingsdaten trainiert. Diese enthalten eine große Zahl von Beispieldaten, welche den später zu verarbeitenden Daten möglichst nahekommen. Nun werden im Rahmen des Trainings die Gewichte des Netzes unter Einfluss von Pseudozufall so optimiert, dass für möglichst alle Bilder der korrekte Output generiert wird. Ziel ist es, dass die aus dem Training resultierenden Gewichte auch für neue, unbekannte Bilder funktionieren. Diese Annahme kann aber nicht verifiziert werden, da die Menge der möglichen Bilder praktisch unbegrenzt ist.[229]

Es besteht aber die Möglichkeit, einen Teil der Trainingsdaten zurückzuhalten und das Netz mit diesen unbekannten Daten zu erproben (sog. »Evaluation«). Diese Evaluation gibt stichprobenartig Aufschluss, ob das Netz Muster erkennen und zuordnen kann oder ob das Netz die Daten nur »auswendig gelernt« hat.[230] Eine solche Evaluation ist im übertragenen

226 *Heussen,* CR 2004, 1, 1 ff.
227 Vgl. *Marly,* Rn. 1437 ff.
228 *Zoph/Vasudevan/Shlens/Le*, S. 1 ff.
229 Vgl. *Sun/Huang/Kroening et al.*, S. 2.
230 In diesem Fall wird dann von »Overfitting« gesprochen, s. o., Trainingsphase (S. 47).

Sinne jedoch stets nur ein Indiz und kein Beweis.[231] Zudem haben Untersuchungen mehrfach gezeigt, dass es recht einfach möglich ist, bei neuronalen Netzen zur Bilderkennung eine falsche Ausgabe zu erzeugen. Die Verfahren für solche sog. »*Adverserial Attacks*« reichen von dem Generieren von non-sense Bildern[232] über das Nutzen von gezieltem Rauschen[233] bis hin zu simplen Lageänderungen eines Objekts auf einem Bild.[234] Auch kann sich aufgrund der Mächtigkeit und Komplexität der Netze unbemerkt eine ungewollte Verhaltensweise in das KI-System eintrainieren.

Ein prägnantes Beispiel ist der sog. »Wolf-Husky Klassifikator«.[235] Dieser Klassifikator wurde – zu Anschauungszwecken der Problematik – anhand eines unausgewogenen Datensets trainiert.[236] Das Trainingsdatenset enthielt Bilder sowohl von Huskys als auch von Wölfen, wobei die Huskys auf den Trainingsbildern stets in einer grünen Umgebung abgebildet waren, während die Wölfe dagegen stets mit Schnee im Hintergrund. Der so trainierte Klassifikator war im Rahmen der Evaluation – welche ebenfalls mit einem Teil desselben unausgewogenen Datensets durchgeführt wurde – in der Lage, die Bilder korrekt zuzuordnen. Allerdings unterschied der Klassifikator nicht zwischen Wölfen und Huskys, sondern orientierte sich unerkannt an Schnee und Gras, womit er eigentlich ein »Schnee-Gras Klassifikator« ist.[237] Wäre dieser Klassifikator von einem durchschnittlichen Softwareentwickler entwickelt worden, wäre dieser auch mit der üblichen Evaluation nicht in der Lage gewesen, zu erkennen, dass das neuronale Netz nicht in gewollter Weise agiert und damit die eigentliche Aufgabe – das Unterscheiden von Wölfen und Huskys – nicht erfüllt.

Hier wird deutlich, dass auch ein deterministisch ablaufender Trainingsprozess, zumindest aus Sicht eines Dritten, zu subjektiv unvorhergesehenen Ergebnissen führen kann. Das trainierte Netz hat während des Trainingsprozesses letzten Endes lediglich Muster und Strukturen gefunden, die das falsche Problem optimal lösen. Beispiele wie der Wolf-Husky Klassifikator zeigen, dass eine Aussage über die Funktionstüchtigkeit einer Künstlichen

231 Vgl. *Hagan/Demuth/Beale/De Jesús,* Kap. 22, 27.
232 *Nguyen/Yosinki/Clune*, S. 3 ff.
233 *Karmon/Zoran/Goldberg*, S. 4 ff.
234 *Alcorn/Li/Gong et al.,* S. 1.
235 *Ribeiro/Singh/Guestrin*, S. 8 f.
236 Das Wort »unausgewogen« trifft den englischen Begriff »baised« nur zum Teil. Grundsätzlich sind dies Datensets, welche durch ihre besondere Beschaffenheit fehlerhafte Schlüsse zulassen.
237 Ein anderes Beispiel ist ein System, welches Schiffe erkennen sollte, allerdings eigentlich Wasser erkannt hat; vgl. *Käde/v. Maltzan*, CR 2020, 66.

Intelligenz durch eine klassische Evaluation nicht möglich ist. Dieser Umstand erstreckt sich nicht nur auf neuronale Netze, sondern grundsätzlich auf jede komplexere Methode der Künstlichen Intelligenz. Daher stellt sich zunächst die Frage, ob jedes Mal, wenn eine Künstliche Intelligenz eine falsche Ausgabe erzeugt, also ein falsches Verhalten zeigt, dies als Fehler zu bewerten ist und weiterhin ob und wie (falsches) Verhalten im Vorhinein erkannt werden kann.

I. Vom Sein und Sollen

Aus der deterministischen Eigenschaft von Künstlicher Intelligenz folgt zwangsläufig, dass eine falsche Ausgabe – sofern sie existiert – bei gleichbleibender Eingabe immer auftritt. Gleichermaßen muss – wie dargelegt – die Existenz einer solchen falschen Ausgabe dem Nutzer nicht zwangsweise (negativ) auffallen. Auch kann die passende Eingabe nie auftreten. Zudem kann – falls Zufallszahlen eine Rolle spielen – der eigentlich deterministische Pseudozufall von der Zeit abhängig sein, sodass zu unterschiedlichen Zeiten unterschiedliche Ausgaben entstehen können.[238] Insoweit ist es notwendig, beim Verhalten einer Künstlichen Intelligenz zwischen dem Sein und dem Sollen zu unterscheiden. Das Sein ist das tatsächliche Verhalten einer Künstlichen Intelligenz, also die Reaktion auf alle (kombinatorisch) möglichen Eingaben. Das Sollen dagegen ist das Verhalten einer Künstlichen Intelligenz, welches sie zeigen soll. Für die Evaluation des Sollens kann nur ein sehr kleiner Bruchteil aller kombinatorisch möglichen Eingaben betrachtet werden, das heißt, das evaluierte Sollen kann nur einen kleinen Teil des Seins abdecken. Bei den dabei verwendeten Eingaben handelt es sich um bekannte und für möglich erachtete Eingaben. Eine vollständige Evaluation aller kombinatorischen Eingaben ist in vertretbarer Zeit nicht möglich. Folglich deckt sich das bekannte Soll-Verhalten nur in einem sehr kleinen Teil mit dem Sein-Verhalten. Welche Eingaben das Soll-Verhalten bestimmen und in welchem Umfang diese zu prüfen sind, kann sich nur im Einzelfall und am Stand der Technik bestimmen und wird im Regelfall danach zu beurteilen sein, welche Eingaben wahrscheinlich sind. Aus den Ausgaben der wahrscheinlichen Eingaben können sodann statistische Aussagen über das Verhalten der Künstlichen Intelligenz getroffen werden. Daraus resultierend, muss – auch für die weitere juristische Betrachtung – zwischen einem Fehler und dem weiteren unvorhersehbaren Verhalten

238 S. o., Randomisierte Algorithmen und Pseudozufall (S. 54).

differenziert werden: Weicht das Sein-Verhalten vom Soll-Verhalten ab, liegt ein Fehler vor. Tritt allerdings ein Verhalten auf der Seite des Seins auf, das auf der Seite des Sollens vorher unbekannt war, so handelt es sich um unvorhersehbares Verhalten. Gewiss ist ein unvorhersehbares Verhalten nach dem ersten (wahrnehmbaren) Auftreten bekannt und muss daher *ex post* dem Sollen zugeordnet werden, sodass dessen Vermeidung bzw. die Evaluation dieses Verhaltens zukünftig zum Stand der Technik gehört und jedes weitere Auftreten als Fehler zu klassifizieren wäre.

II. Explainable AI Ansätze

Der Umstand, dass durch die Evaluation weitreichendes Verhalten nicht erkannt werden kann, kann als ungenügend empfunden werden.[239] Allgemein gilt, dass ein Produkt – insbesondere, wenn es sicherheitskritisch ist – bei Inverkehrbringen hinreichend getestet und verstanden worden sein sollte. Bei einer Künstlichen Intelligenz, deren Antworten zwar richtig, aber für niemanden vollends nachvollziehbar sind, ist diese Grundanforderung zunächst nicht erfüllt. Es ist auch problematisch, dass der Umstand der fehlenden Erklärbarkeit kaum Rückschlüsse auf die Qualität des KI-Systems erlaubt. In sicherheitskritischen Anwendungen kann dies zu einem erheblichen Hemmnis bei der Anwendung von Künstlicher Intelligenz führen.

Um diesem Problem zu begegnen, hat sich im Rahmender KI-Forschung das Forschungsfeld der »Explainable AI« oder kurz xAI gebildet. Zu den wichtigsten Methoden dieses Forschungsgebiets,[240] insbesondere im Bereich der neuronalen Netze, zählen die Sensitivitätsanalyse (SA),[241] die *Layer-wise Relevance Propagation* (LRP)[242] sowie das *Class Activation Mapping* (CAM).[243] Zudem existieren Methoden, um die kaum interpretierbaren, neuronalen Netze in leichter verständliche Entscheidungsbäume zu überführen.[244] Die detaillierte Funktionsweise dieser Verfahren sollen nachfolgend allerdings nicht näher betrachtet werden. Gemeinsam haben

239 *Käde/v. Maltzan*, CR 2020, 66 ff.
240 Vgl. *Käde/v. Maltzan*, CR 2020, 66, 68 ff.
241 Vgl. *Baehrens/Schroeter/Harmeling et al.*, Journal of Machine Learning Research 11 (2010), 1803 ff.
242 Vgl. *Bach/Binder/Montavon et al.*, PLoS ONE 10(7):e0130140 (2015), 1 ff.
243 *Zhou/Khosla/Lapedriza et al.*, S. 1 ff.
244 Vgl. *Schaaf/Huber/Maucher*, S. 1 ff.

diese Methoden jedoch, dass sie deutlich tiefere Erkenntnisse in die komplexen Prozesse eines neuronalen Netzes zulassen und so das fehlerhafte Verhalten, wie das des Wolf-Husky-Klassifikators, erkennbar machen. Damit wird die dortige Klassifikation von Gras und Schnee zum Fehler, was bei klassischer Evaluation nach der im vorherigen Abschnitt vorgestellten Unterscheidung noch als unvorhersehbares Verhalten gegolten hätte.

Als Einschränkung ist aber zu berücksichtigen, dass jegliche Erklärungsansätze immer lediglich eine Unterstützung bei der Interpretation der Entscheidungen einer Künstlichen Intelligenz sind. Die Methoden erreichen ihre Grenzen, sobald die Erklärung an sich zu kompliziert für einen Entwickler wird, sodass auch hier ein Komplexitätsproblem entstehen kann. Dennoch können nun zahlreiche Fehlerklassen, die bis vor einigen Jahren unerkannt geblieben wären, mit xAI jetzt aufgedeckt werden, sodass Verhalten in diesen Klassen als Fehler und nicht mehr als unvorhersehbares Verhalten betrachtet werden muss. Hierbei ist allerdings zu beachten, dass die hier vorgestellten Methoden den Stand der Forschung beschreiben. Entwicklertools existieren nur vereinzelt[245] und entsprechende Normen befinden sich derzeit in der Entstehungsphase.[246] Ungeachtet dessen sind Prinzipen der xAI bereits bekannt und können von einem Entwickler innerhalb weniger Monate implementiert werden. Insofern kann insbesondere bei sicherheitskritischen Anwendungen eine Berücksichtigung dieser Methoden durchaus gefordert werden.[247]

G. Wissen und dessen Repräsentation

Nach der ausführlichen Darlegung über das Verhalten von Künstlicher Intelligenz steht nun fest, dass eine Künstliche Intelligenz keinen Willen hat, aber handeln und sich verhalten kann. Offen bleibt aber noch, ob eine Künstliche Intelligenz etwas wissen kann und wenn ja, welche Eigenschaften dieses Wissen hat. Gleich den vorangegangenen Begriffen ist auch der Begriff des Wissens domänenabhängig und wird teilweise unterschiedlich betrachtet. In der Informatik wird »Wissen« durch das DIKW-

245 Beispieltools: *Bellamy/Dey/Hind et al.*, S. 1 ff.; Fraunhofer Institut für Nachrichtentechnik, Erklärbarkeit durch LRP, Tool verfügbar unter: https://lrpserver.hhi.fraunhofer.de/image-classification (zuletzt abgerufen am 01.06.2020); Google, »What if tool« verfügbar unter: https://pair-code.github.io/what-if-tool/fat2020.html (zuletzt abgerufen am 01.06.2020).

246 Vgl. DIN, Künstliche Intelligenz, S. 3.

247 S. u., Fahrlässigkeit (S. 120).

Modell[248] beschrieben und von »Zeichen«, »Daten« und »Information«[249] hierarchisch abgegrenzt.[250] Das Zeichen[251] bildet den elementaren Baustein einer formalen Sprache und somit den Anfang der Hierarchie. Werden Zeichen kombiniert, entstehen Daten bzw. Wörter, die auch Entitäten benennen können. Für sich genommen liefern Daten keine Information; um sie zu bilden, müssen sich Daten erst gegenseitig referenzieren. Das erfolgt üblicherweise über einen Subjekt-Prädikat-Objekt-Tripel. Hierbei wird einem Subjekt – sprich einem Konzept, über das eine Aussage getroffen werden soll – mit einem Objekt – also dessen konkreter Ausprägung – verknüpft. Um welche Eigenschaft es sich handelt, wird durch das Prädikat beschrieben. Hierauf aufbauend stellt das Wissen sodann die Vernetzung von Informationen dar.[252] Die Repräsentation von Wissen in der Informatik, insbesondere in der Lehre der Expertensysteme,[253] erfolgt üblicherweise durch ein semantisches Netz respektive eine Ressource Description Framework (RDF) Ontologie.[254] Hierbei können beliebig viele Subjekt-Prädikat-Objekt-Tripel verknüpft werden, welche so das semantische Netz bilden.[255] Inwieweit auch neuronale Netze unter diesen Wissensbegriff fallen, ist zum heutigen Zeitpunkt ungeklärt und Gegenstand der Forschung.[256]

Eine andere Betrachtung von Wissen findet im sog. Wissensmanagement statt. Hier wird Wissen als ein zentraler Wettbewerbsvorteil verstanden, welcher den Erfolg eines Unternehmens maßgeblich mitbestimmt.[257] Diese Ressource zu verwalten und zu erhalten ist das Ziel des Wissensmanagements.[258] Weite Teile der Begriffswelt der Informatik, wie Daten und Information, finden auch im Rahmen des Wissensmanagements Anwendung. Der Wissensbegriff beschreibt hier alle (Denk-)Modelle über Objekte und Sachverhalte.[259] Eine Besonderheit im Rahmen des Wissensmanagements ist die Unterscheidung zwischen dem sog. »expliziten

248 »Data, information, knowledge and wisdom«.
249 Gelegentlich wird zudem Weisheit als höchste Stufe der Hierarchie genannt, wobei dies einen theoretischer Ansatz darstellt, der aus Sicht der praktischen Informatik nicht von Relevanz ist.
250 Der Autor hiervon ist unbekannt, vgl. *Wallace*, S. 13.
251 Beispielsweise A-Z, 0-9 oder Sonderzeichen (+-# etc.).
252 Vgl. *Zech*, S. 28 f.
253 S. o., Semantic Web Systeme/Expertensysteme (S. 40).
254 Vgl. *Krempel*, S. 154.
255 *Fischer/Hofer*, S. 684 (»RDF«).
256 Vgl. *Binder/Bach/Montavon et al.*, Information Science and Applications 2016, 913 ff.; *Shu/Zhu*, S. 1 ff.; *Yosinski/Clune/Nguyen et al.*, S. 1 ff.
257 *Gronau/Bahrs/Vladova et al.,* S. 6.
258 Vgl. *Wiendahl*, S. 328.
259 *Wilkesmann/Rascher*, S. 19 f.

Wissen« und dem sog. »impliziten Wissen«. Das explizite Wissen stellt hierbei jenen Teil des Wissens dar, welches sich verhältnismäßig leicht auf ein anderes Individuum übertragen lässt. Klassischerweise ist das in Büchern enthaltene Wissen explizites Wissen. Implizites Wissen dagegen ist der Verbalisierung nicht zugänglich[260] und muss so durch den Lernenden erarbeitet werden. Als Beispiele werden Erfahrungswissen,[261] motorische Fertigkeiten oder soziales Gespür angeführt.[262] Der Wissensbegriff im Wissensmanagement dient somit vorrangig dazu, eine Ressource mit ihren Eigenarten zu beschreiben. Im Vergleich ist festzustellen, dass der Wissensbegriff des Wissensmanagements weiter ist als der Wissensbegriff der Informatik. Insbesondere das Konzept des impliziten Wissens unterscheidet sich fundamental von dem Verständnis der Informatik, deren Wissen stets – wenn auch nur maschinell – artikulierbar ist. Auch kann Wissen in Maschinen – mangels Bewusstseins – weder unbewusst noch bewusst vorliegen. Wird also Wissen in Bezug auf Künstliche Intelligenz diskutiert, ist darauf zu achten, die technischen Eigenarten zu berücksichtigen.

H. Zwischenergebnis

Aus technischer Sicht lässt sich festhalten, dass Künstliche Intelligenz ein Eigenbegriff ist, der zunächst Oberbegriff für diverse (abstrakte) Ansätze (KI-Methoden) darstellt, wie beispielsweise (künstliche) neuronale Netze, Suchbäume, Fuzzylogiken, etc.[263] Die Methoden an sich sind allerdings nur Konzepte, die in der realen Welt keine Wirkung entfalten. Hierfür bedarf es ihrer Implementierung in Computerprogramme. Daher ist die für die Rechtswissenschaft und damit auch die im weiteren Verlauf relevante Entität die reale Implementierung von Künstlicher Intelligenz in Computerprogrammen. Das Ziel von Künstlicher Intelligenz ist es, Handlungen vorzunehmen, die eigentlich (derzeit) menschliche Handlungen sind. Dieser Zweck wird zwar grundsätzlich mit jedem Computer oder sogar mit jeder Maschine verfolgt, allerdings sucht der Forschungsbereich der Künstlichen Intelligenz nach besonders geeigneten Ansätzen hierfür.

Das Phänomen der Künstlichen Intelligenz ist dabei so alt wie der programmierbare Rechner bzw. der moderne Computer selbst. Schon seit Be-

260 *Loenhoff*, S. 20.
261 *Wiater*, S. 22.
262 *Loenhoff*, S. 4.
263 S. o., Methoden der Künstlichen Intelligenz im Überblick (S. 34).

stehen solcher Computersysteme wird an derartiger Computersoftware geforscht, was sich gut am Beispiel des Schachcomputers zeigt. Allgemein zeigt Künstliche Intelligenz durch die Handlungen, zu deren Zweck sie geschaffen wurde, ein von außen beobachtbares Verhalten. Aus diesem Verhalten darf aber nicht fehlerhaft geschlussfolgert werden, es würde sich um Maschinen handeln, die denken, also in dem Sinne intelligent sind wie Menschen. Aus den Restriktionen, denen ein Computer unterliegt, folgt deshalb auch, dass die diskutierten Ansätze der sog. »starken Künstlichen Intelligenz« keine Rolle spielen.

Zudem gibt es nicht »die« Künstliche Intelligenz, sondern es handelt sich vielmehr um verschiedene Ansätze zur Lösung bestimmter Probleme mit dafür gezielt geschaffenen, rational und möglichst effizient handelnden Programmen. KI-Methoden basieren auf Algorithmen und sind daher zwangsläufig deren Restriktionen unterworfen. Wie aufgezeigt, sind alle real implementierten Algorithmen, die terminieren, auch deterministisch, sowohl im philosophischen als auch im informatischen Sinne. Für die in der theoretischen Informatik als nicht-deterministisch bezeichneten randomisierten Algorithmen gilt dies ebenfalls, da die für diese Algorithmen notwendigen Zufallszahlen in der Praxis entweder auch deterministisch erzeugt werden oder von der jeweiligen Eingabe abhängen. Aufgrund der steigenden Komplexität von Computerprogrammen im Allgemeinen ist es schon problematisch, dass zwar theoretisch jede Ausgabe zu jeder Eingabe vorherberechnet werden kann, dies aber in der realen Umsetzung viel zu lange dauern würde. Gerade dieses Problem wird durch Künstliche Intelligenz noch einmal verstärkt. Daraus folgt, dass es nicht mehr praktikabel ist – in realistischer Zeit sogar praktisch unmöglich – alle möglichen Handlungsmöglichkeiten vorherzubestimmen, das heißt alle möglichen Eingaben durchzurechnen, um die jeweiligen Ausgaben zu bestimmen. Möglich sind daher nur noch statistische Aussagen. Aus dieser Komplexität kann – jedenfalls nach außen – eine Autonomie von Computerprogrammen bzw. Künstlicher Intelligenz folgen, die sich von rein vorprogrammierten Entscheidungen unterscheidet, ohne aber einen freien Willen der Maschine zu sein. Diese Autonomie ist vielmehr Folge des Anlernens oder einer stochastischen Berechnung der Eingabe. Zumindest theoretisch kann jede Handlung – als Folge des Determinismus – vorherberechnet werden. Dennoch müssen als Konsequenz des zwar deterministischen aber komplex wie komplizierten Phänomens der Künstlichen Intelligenz gewisse rechtliche Ansätze möglicherweise in einem neuen Licht betrachtet werden. Fraglich bleibt daher, ob der Satz von *Köhler*, »da[ss] auch das komplizierteste Datenverarbeitungssystem letztlich keine autonomen Entscheidungen,

sondern nur logische Operationen nach einem vorgegebenen Programm durchführen kann«,[264] letztendlich noch zutrifft.

264 *Köhler,* AcP 182 (1982), 126, 133.