Erschienen in: Hirsch-Kreinsen, H. & Karačić, A. (Hg.): Autonome Systeme und Arbeit. Perspektiven, Herausforderungen und Grenzen der Künstlichen Intelligenz in der Arbeitswelt, Bielefeld: transcript 2019, 69-97

Peter Brödner

Grenzen und Widersprüche der Entwicklung und Anwendung ›Autonomer Systeme‹

1 Einführung: Überzogene Erwartungen

Derzeit überrollt uns nach drei Jahrzehnten relativer Stille erneut eine Sturzflut von Meldungen über Projekte vermeintlicher ›künstlicher Intelligenz‹ (KI, engl. AI für ›artificial intelligence‹). Die Spannweite der Meldungen reicht von Heilsversprechen über Erfolgsgeschichten bis zu Szenarien der Apokalypse. Als kleine Auswahl aus dem gesamten Spektrum vermitteln die nachstehenden Äußerungen einen Eindruck:

- Unter der Überschrift »The Future AI Company« wird die künftige Automobilfabrik so beschrieben: »Superschlaue Computer, die ständig lernen, werden vieles übernehmen, was bisher Menschen erledigen: Sie antworten, wenn Kunden oder Lieferanten fragen, automatisch […] [S]ie entwerfen sogar Autos und rechnen aus, wie sich die Entwürfe in der Fabrik umsetzen lassen.« (Hofmann, VW-Vorstand, laut Schäfer 2016),
- »Lernende Maschinen erkennen Gesichter und bringen sich selbst das Schachspielen bei.« (Dworschak 2018),
- »Invasion der Roboter: Künstliche Intelligenz ist bald so normal wie Strom.« (Recke 2016),
- Laut Google-CEO Pichai ist künstliche Intelligenz für die Menschheit bedeutender als die Entdeckung des Feuers oder die Entwicklung der Elektrizität (Bastian 2018),
- Hawking fears »AI may replace humans altogether« as a »new form of life that will outperform humans.« (Sulleyman 2017).

Gegenüber solchen offenkundig sensationslüsternen, auf Verblüffung der Öffentlichkeit angelegten Äußerungen gilt es zunächst einmal den gesunden Menschenverstand zu bewahren und Aufklärung in wissenschaftlicher Analyse zu suchen. Auffällig an diesen und ähnlichen Äußerungen ist zunächst eine Gemeinsamkeit: Heilsverkünder/-innen wie Apokalyptiker/-innen sind gleichermaßen gegen Tatsachen immun. Gegen darin zum Ausdruck kommende ›KI‹-Wahnvorstellungen hilft daher nur, die Wahrheit in relevanten Tatsachen zu suchen. Die Probleme beginnen freilich schon damit, dass es bis heute nicht gelingt, ›KI‹-Systeme logisch zufriedenstellend von ›gewöhnlichen‹ Computersystemen zu unterscheiden. Zwecks »Maschinisierung von Kopfarbeit« (Nake 1992) in Zeichenprozessen sozialer Praxis werden beide, auch gewöhnliche Computersysteme, immer schon zur Bewältigung von Aufgaben geschaffen »commonly thought to require intelligence« (so eine übliche ›KI‹-Definition, vgl. Autorengruppe 2018). Dazu führen beide Systemarten gleichermaßen berechenbare Funktionen zur planvollen automatischen Verarbeitung zugehöriger Daten aus (vgl. Brödner 2008).

Bemerkenswert an Äußerungen wie den zitierten ist ferner, dass sie allesamt bestimmte Computerartefakte als solche in den Blick nehmen und diesen die Verblüffung hervorrufenden Eigenschaften zuschreiben. Dieser Fokussierung auf technische Artefakte liegt aber ein ebenso verbreitetes wie tief gehendes Missverständnis von Technik im Allgemeinen und der Computertechnik im Besonderen zugrunde. Einem Bonmot des Philosophen Ortega y Gasset (1949) zufolge ist Technik »die Anstrengung, Anstrengungen zu ersparen«. Damit trifft er den Kern der Sache: Genauere Analyse zeigt nämlich, dass Technik, verstanden als bloße Ansammlung

technischer Artefakte, viel zu kurz greift. Artefakte fallen nicht vom Himmel, sondern müssen für bestimmte Zwecke mühsam konzeptionell entwickelt und materiell hergestellt werden. Als solche sind sie aber bloß tote, nutzlose Gegenstände, solange sie nicht für bestimmte Aufgaben zweckgemäß eingesetzt, mithin dafür angeeignet und praktisch wirksam verwendet werden. Das alles geschieht im Spannungsfeld des technisch Machbaren, der Formbarkeit von Natur, und des sozial Wünschenswerten, abhängig von jeweils herrschenden Interessen.

Nach allgemeinem professionellem Verständnis wird Technik daher definiert als die Gesamtheit von Maßnahmen zur Herstellung und zum Gebrauch künstlicher Mittel für gesellschaftliche Zwecke. Ihr werden damit nicht nur die Artefakte und Sachsysteme selbst zugerechnet, sondern gerade auch deren sozial konstruierte und kulturell vermittelte Herstellung und Anwendung (vgl. Ropohl 1991, VDI 1991). Als geronnene Erfahrung verkörpern sie ein Stück sozialer Praxis und als Arbeitsmittel stellen sie Handlungsanforderungen an ihren Gebrauch, durch den Artefakte erst ihren Sinn erhalten und in ihrer Qualität zu beurteilen sind. Eben in den Prozessen der Entwicklung und Herstellung technischer Artefakte sowie ihrer Aneignung zu praktisch wirksamer Verwendung liegen gerade die eigentlichen Probleme der »Anstrengung, Anstrengungen zu ersparen«; eben hierin liegen auch die Wurzeln missbräuchlichen Umgangs.

Sich in dieser Perspektive einigen der Probleme fortgeschrittener Computerentwicklung zu stellen, ist das Anliegen des vorliegenden Beitrags. Dazu werden im Folgenden zunächst am Beispiel sogenannter ›künstlicher neuronaler Netze‹ (KNN) und Verfahren des ›Deep Learning‹, die derzeit als vermeintliche Schlüsseltechnik der ›KI‹ besonders hoch im Kurs stehen, deren Entwicklungsprobleme und Funktionsweisen aufgezeigt und hinsichtlich ihrer Tragweite beurteilt. Sodann werden die besonderen Einsatz- und Anwendungsprobleme beleuchtet, die mit diesen Systemen im Vergleich zu herkömmlichen Computersystemen verbunden sind. Vor dem Hintergrund eines kurzen historischen Exkurses über Meilensteine der Entwicklung von Computertechnik und Computing Science wird anschließend über verbreitete, tief sitzende Missverständnisse von Funktionsweise und Leistungsgrenzen von Computern aufgeklärt. Darauf fußend wird eine abschließende Bewertung vorgenommen.

2 ›Deep Learning‹ und seine unterschätzten Entwicklungsprobleme

Neben Verfahren zur Analyse von ›Big Data‹ auf Basis von Methoden schließender Statistik bieten sogenannte ›künstliche neuronale Netze‹ (KNN) einen zweiten grundlegenden Ansatz der Verwirklichung sog. ›maschinellen Lernens‹. Infolge exponentiell gesteigerter Leistung der Computer-Hardware sind sie in letzter Zeit zu einem bevorzugten Gegenstand der ›KI‹-Forschung geworden. In Form vielschichtig strukturierter adaptiver Netzwerke bieten sie Verfahren zum sog. ›Deep Learning‹ (LeCun et al. 2015, Schmidhuber 2015) als einer Art neuer ›Wunderwaffe‹.

Im Unterschied zu früheren ›KI‹-Ansätzen der ›allgemeinen Problemlösung‹ (Simon & Newell 1972) oder der ›wissensbasierten Systeme‹ der ›symbolischen KI‹ bzw. des ›Cognitive Computing‹ (IBM), wie sie u.a. beim ›Computer Integrated Manufacturing‹ der 1980er Jahre verfolgt wurden, setzt die heute dominante neue Welle der ›KI‹-Forschung nach einer langen Periode stark gebremster Aktivität nun vornehmlich auf die Entwicklung und den Einsatz von KNN als einem Grundmodell ›lernfähiger‹ Systeme. Dieser Ansatz ist freilich keineswegs neu, sondern geht in seinen Anfängen zurück auf das »Perceptron« (McCulloch/Pitts 1943) als einer biologisch inspirierten Nachahmung der logischen Funktionsweise von Nervenzellen bereits in der Zeit der allerersten materiellen Realisierungen von Computern. In Gestalt der »Lernmatrix« von Steinbuch 1961 wieder aufgegriffen und zu größeren Netzwerken verknüpft (vgl. Hilberg 1995), fiel dieser Ansatz nach einem Verdikt durch Minsky und Papert (1969) erneut in einen Dornröschenschlaf,

aus dem er in den 1990er Jahren langsam wieder erwachte. Wachgeküsst wurde er von der Aussicht auf ›maschinelles Lernen‹ im Zusammenhang mit der Einsicht, dass die sich auf die logische Verarbeitung von durch Daten repräsentiertem explizitem Wissen stützende ›symbolische KI‹ ihrerseits an den Grenzen der Explizierbarkeit von Können bzw. implizitem Wissen gescheitert war (vgl. Brödner 1997).

Versuche, kognitive Leistungen des Menschen, insbesondere sein intelligentes Handeln nachzuahmen, legen die Grundidee und den Ansatz nahe, das Gehirn mit seinen vielschichtig vernetzten Neuronen als Vorbild zu nehmen und einige seiner Strukturmerkmale und Funktionen möglichst direkt in Computersystemen nachzubilden. Freilich dürfen solche konnektionistischen Modelle nicht missverstanden werden als ›naturgetreue‹ Nachbildungen des Gehirns oder des Zentralnervensystems, allenfalls gibt es gewisse strukturelle und funktionale Ähnlichkeiten, die durch den Aufbau des Gehirns inspiriert werden.

Ein KNN besteht im Wesentlichen aus einer Menge miteinander verbundener *Knoten*, die in Abhängigkeit von ihrem aktuellen Aktivierungszustand und der momentanen Eingabe ihren neuen Zustand bestimmen und eine Ausgabe produzieren. Diese Elemente sind gemäß der *Netzwerkstruktur* miteinander verknüpft. Diese kann als gerichteter gewichteter Graph oder durch eine *Konnektionsmatrix* dargestellt werden. Die Dynamik eines KNN wird beschrieben durch (vgl. Abb. 1):

- eine *Propagierungs-* bzw. *Übertragungsfunktion* net_j, die aus den Ausgaben der vorgeschalteten Elemente sowie der Gewichtung der Verbindungen die aktuellen Eingaben in interne Netzwerkelemente berechnet,
- eine *Aktivierungsfunktion* σ, die für jedes Element dessen Aktivierung o_j als Ausgabe bestimmt abhängig davon, ob ein Schwellwert von der Netzeingabe net_j überschritten wird oder nicht.

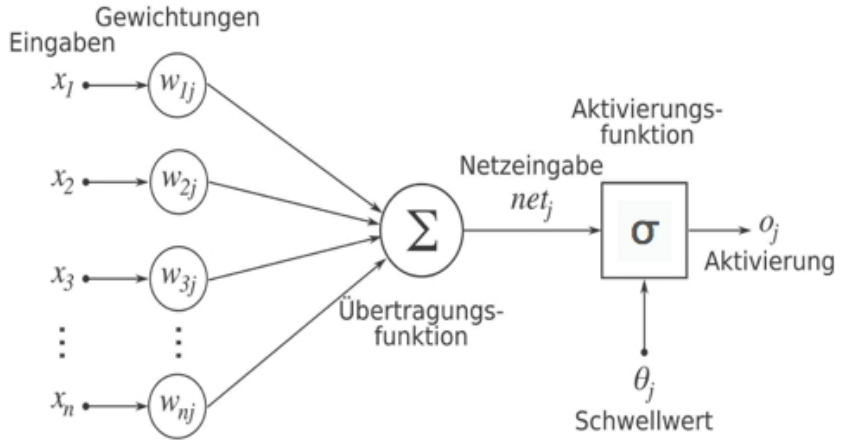

Abb. 1: Berechnungsfunktionen an einem Netzknoten (Quelle: Wikipedia CC BY-SA 3.0)

In der Regel besteht die Propagierungsfunktion aus einer einfachen Summenbildung der gewichteten Verbindungseinflüsse und die Aktivierungsfunktion wird meist für alle Elemente des Netzwerks einheitlich festgelegt. Innerhalb eines Netzwerks wird noch zwischen Eingabe-, Ausgabe- und internen Elementen unterschieden (›hidden units‹, die zu tief gestaffelten ›hidden layers‹ zusammengefasst werden, daher die Bezeichnung ›Deep Learning‹). Mittels verschiedener Typen von Ausgabe-, Propagierungs- und Aktivierungsfunktionen können Klassen von konnektionistischen Modellen gebildet werden.

Um ihre Aufgaben zu lösen, müssen KNN während des Entwurfs passend strukturiert und ihre Prozessoren mit einem bestimmten ›Lern‹-Algorithmus gesteuert werden. Das Netz als Ganzes wird über diese Festlegungen hinaus nicht programmiert, sondern passt sich durch Veränderung der Gewichte nach Maßgabe eines den Nutzen maximierenden ›Lern‹-Algorithmus an die spezielle Aufgabenstellung an (daher die Benennung *adaptiv*). Beispielsweise werden bei Problemen der Musteridentifikation oder der Klassifikation – ein Aufgabentyp, bei dem künstliche neuronale Netze, vor allem solche vom Typ der ›faltenden‹ oder ›Convolutional Neural Networks (CNN)‹, besonders leistungsfähig sind – einer langen Reihe von Eingabemustern jeweils die zugehörigen Klassen als Ausgänge zugeordnet; aus diesen Zuordnungen vermag dann der Algorithmus mittels einer Nutzenfunktion automatisch passende Verbindungsgewichte zu bestimmen. Dies funktioniert auch bei Mustern, die durch explizite Merkmalsbeschreibungen schwierig oder gar nicht zu fassen sind (etwa bei der Identifikation handgeschriebener Buchstaben) – freilich mit Unsicherheiten. Meist ist dafür allerdings eine sehr große Zahl von Trainingsbeispielen (in der Größenordnung von 10^6) erforderlich (vgl. LeCun et al. 1998).

Bei der Anpassung ist für die Bestimmung der Gewichte $W_j := W_j - \eta \nabla_W L$ eines KNN die Berechnung des Gradienten $\nabla_W L(N(x))$ einer Nutzen- oder Verlustfunktion $L(N(x))$ erforderlich (mit $N(x)$ als Netzwerksausgabe, $L(N)$ als etwa über alle Trainingsbeispiele summierter euklidischer Distanz sowie der »Lernrate« η). Mit der Aktivierungsfunktion $\sigma(x)$ lässt sich – am einfachen Beispiel eines dreischichtigen KNN – die Funktionsweise des Backpropagation-Algorithmus und das häufig auftretende Problem des »schwindenden Gradienten« aufzeigen:

$$N(x) = W_1 \cdot \sigma(\overbrace{W_2 \cdot \sigma(\underbrace{W_3 \cdot x}_{N_3})}^{N_2})$$

Die Bestimmung des Gradienten von L erfordert die Bildung der Ableitung von $N(x)$ als Verkettung mehrerer Funktionen nach der Kettenregel:

$$\begin{aligned}
\frac{\mathrm{d}L}{\mathrm{d}W_1} &= \sigma(N_2) \cdot \frac{\mathrm{d}L}{\mathrm{d}N} \\
\frac{\mathrm{d}L}{\mathrm{d}W_2} &= \sigma(N_3) \frac{\mathrm{d}\sigma(N_2)}{\mathrm{d}N_2} \cdot W_1 \cdot \frac{\mathrm{d}L}{\mathrm{d}N} \\
\frac{\mathrm{d}L}{\mathrm{d}W_3} &= x \cdot \frac{\mathrm{d}\sigma(N_3)}{\mathrm{d}N_3} \cdot W_2 \cdot \frac{\mathrm{d}\sigma(N_2)}{\mathrm{d}N_2} \cdot W_1 \cdot \frac{\mathrm{d}L}{\mathrm{d}N}
\end{aligned}$$

Dabei wiederholt auftretende Faktoren sind während des Trainings oft < 1, sodass das Ergebnis infolge ihrer Multiplikation insgesamt gegen Null tendiert – daher der »schwindende Gradient« und die Schwierigkeit, vielschichtige KNN zu trainieren. Zudem ist es schwierig, bei der Suche nach den Extremwerten der Nutzenfunktion eine variable, situativ passende Schrittweitensteuerung zu realisieren (vgl. Schmidhuber 2015, Wick 2017). Diese praktischen Schwierigkeiten bei der Strukturierung wie beim Training der Netzwerke sind allesamt nur durch jeweils fallspezifische Kunstgriffe zu überwinden, die das Können ihrer Entwickler/-innen herausfordern.

Für die Gestaltung der KNN gibt es keine theoretisch fundierten Erkenntnisse. Folglich müssen für jede Aufgabe passende Netzwerkstrukturen und Nutzenfunktionen mühsam mit großem Trainingsaufwand und ohne Erfolgsgarantie ausprobiert werden. Die Performanz der Netzwerke verdankt sich daher allein der Erfahrung, dem Können und der Kreativität ihrer Entwickler/-innen,

darüber hinaus auch der in den letzten Jahren gemäß dem ›Mooreschen Gesetz‹ enorm gesteigerten Leistungsfähigkeit von Computer-Hardware.

Zudem sind KNN in hohem Maße störanfällig. Schon durch geringfügige Veränderung der eingegebenen Bilddaten können sie in ihrer Funktionstüchtigkeit stark beeinträchtigt werden. So gibt es denn auch vielfältige Fehlleistungen von ansonsten erfolgreichen KNN und zahlreiche Beispiele sind belegt etwa bei der Bild-Klassifikation (vgl. z.B. Szegedy et al. 2014, Nguyen et al. 2015, Sharif et al. 2016, Sitawarin et al. 2016, Metzen et al. 2017).

In Anbetracht der immer häufiger eingesetzten Systeme zur automatischen ›Gesichtserkennung‹ ist das Beispiel eines mit einfachen Mitteln ausgetricksten Standardsystems zur Bildklassifikation nach neustem Stand der Technik besonders interessant und eindrücklich. Diese Systeme nutzen ebenfalls KNN, um im Falle von Gesichtsbildern anhand körperlicher Eigenheiten wie Position und Form der Nase oder Augenbrauen – mit Millionen Bildern trainiert – Personen voneinander zu unterscheiden. Werden diese Bereiche von einer Brille überdeckt, lässt das bunte Muster das KNN zur Gesichtsklassifikation Eigenheiten ausmachen, die fälschlicherweise als Gesichtsdetails ausgewertet werden. Ein männlicher Proband wurde so als die Schauspielerin Milla Jovovich erkannt (b), mit einer Genauigkeit von 87,9 %, eine Asiatin hielt die Software mittels Brille für einen Mann aus dem arabischen Raum (c) etc. (vgl. Abb. 2).

(a) (b) (c) (d)

Examples of successful impersonation and dodging attacks. Fig. (a) shows S_A (top) and S_B (bottom) dodging against DNN_B. Fig. (b)–(d) show impersonations. Impersonators carrying out the attack are shown in the top row and corresponding impersonation targets in the bottom row. Fig. (b) shows S_A impersonating Milla Jovovich (by Georges Biard / CC BY-SA / cropped from https://goo.gl/GlsWlC); (c) S_B impersonating S_C ; and (d) S_C impersonating Carson Daly (by Anthony Quintano / CC BY / cropped from https://goo.gl/VfnDct).

Abb. 2: Ausgetrickste automatische »Gesichtserkennung« (Quelle: Sharif et al. 2016: o.S.)

In letzter Zeit hat das System AlphaGo von Alphabet viel Aufsehen erregt, das in Turnieren die weltbesten Go-Spieler zu schlagen vermochte. Es wird häufig und zur Überraschung vieler als der ultimative Nachweis von dem Menschen überlegener ›künstlicher Intelligenz‹ und ›maschinellem Lernen‹ präsentiert. Bei genauerem Hinsehen zeigt sich aber auch hier, dass diese Behauptungen auf dem propagandistischen Treibsand falscher Zuschreibungen gebaut sind.

Zunächst ist festzustellen, dass das Go-Spiel ein mathematisches Objekt ist, das durch seine Regeln vollständig definiert ist. Infolgedessen lässt sich jederzeit in jeder Stellung eines beliebigen Spielverlaufs eindeutig entscheiden, ob ein Spielzug erlaubt ist oder nicht. Im Prinzip ließe sich daher auch der Baum aller möglichen erlaubten Spielzüge und -verläufe darstellen (was freilich wegen sog. ›kombinatorischer Explosion‹, hier der gigantischen Zahl von geschätzt rd. 200^{150} Zweigen, physisch unmöglich ist; es handelt sich um ein NP-vollständiges Problem).

Mittels heuristischer Verfahren muss daher die Suche nach erfolgreichen Spielzügen und einem möglichst optimalen Spielverlauf auf aussichtsreiche Teilbäume beschränkt werden. Als Methode bewährt hat sich bei vergleichbaren Aufgabenstellungen die Monte Carlo Tree Search (MCTS; vgl. Browne et al. 2012), die auch hier als ein heuristischer Verfahrens-Baustein zum Tragen kommt. Dabei werden für Folgezüge einer betrachteten Stellung – dargestellt als Wurzel des Teilbaums aussichtsreicher Züge – Erfolgshäufigkeiten (Verhältnis von Anzahl gewonnener zur Anzahl der insgesamt über diesen Zweig vollzogenen Spiele) mit Hilfe von parallel durchgängig simulierten Zufallspartien ermittelt, die laufend fortgeschrieben werden (vgl. Abb. 3). Im Falle von AlphaGo wurden dazu auf 40 parallel arbeitenden Prozessoren jeweils 10^3 Simulationen pro Sekunde durchgeführt (was bei einer Denkzeit von einer Minute zwischen den Zügen maximal 2,4 Mio. Simulationen pro Zug ermöglicht).

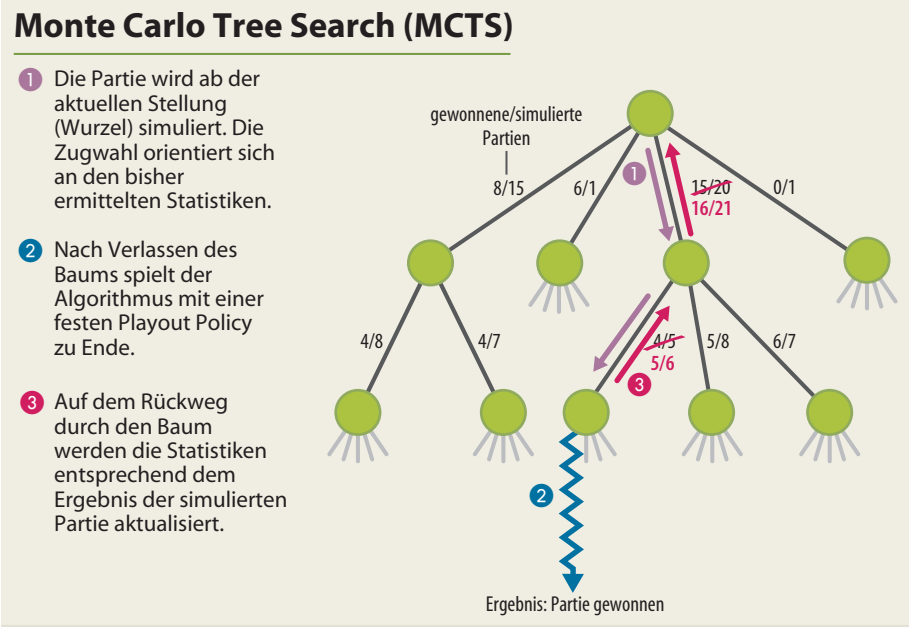

Abb. 3: Funktionsweise der Methode Monte Carlo Tree Search (Quelle: Bögeholz 2016:151)

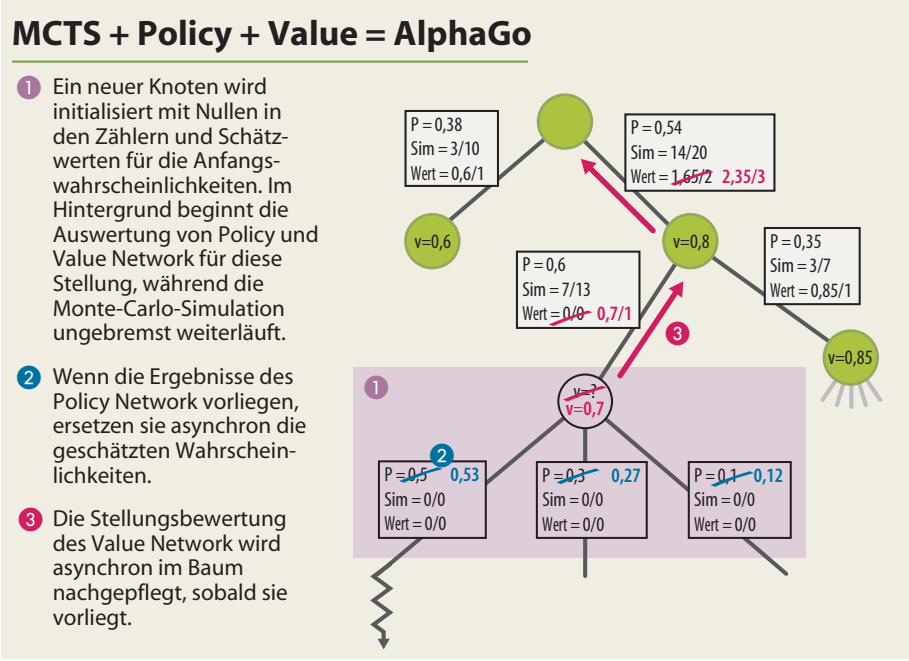

Abb. 4: Kombination der MTCS mit KNN-basierten Bewertungen (Quelle: Bögeholz 2016: 151)

In ihrer Leistung gesteigert wird die MCTS noch durch Kombination mit zwei im Spiel gegen sich selbst trainierten neuronalen Netzen, die im Spielverlauf asynchron zusätzliche Bewertungen zur Zugwahl (»policy«, in Form einer über die Zweige verteilten Erfolgswahrscheinlichkeit) und zur Stellung (»value«, als relativem Wert des zugehörigen Teilbaums) ermitteln. Dabei wächst der betrachtete Teilbaum aussichtsreicher Spielzüge durch Einführung neuer Knoten in besonders erfolgversprechenden Zweigen mit anfangs geschätzten Bewertungsgrößen. Diese werden im Zuge der parallel und asynchron durchgeführten Läufe der MCTS-Simulationen und der Wertbestimmung für Zugwahl und Stellungen durch die neuronalen Netze fortgeschrieben, sobald sie verfügbar sind (vgl. Abb. 4; weitergehende Einzelheiten der Verfahren und ihres dynamischen Zusammenspiels finden sich bei Silver et al. 2016 sowie Yuandong & Yan 2016).

Als mathematische, durch Regeln vollständig definierte Objekte sind Spiele entgegen landläufiger Auffassung geradezu prädestiniert für Modellierung und Formalisierung ihrer Spielverläufe. Es ist daher immer schon *a priori* sicher, dass es Algorithmen geben muss, die menschlichen Spieler/-innen überlegen sind. Erklärungsbedürftig ist folglich nur, warum sie erst jetzt gefunden wurden; Gründe dafür sind:

- es bedarf der Entwicklung hinreichender methodischer Erfahrung und des notwendigen mathematischen Könnens, um leistungsfähige Heuristiken (z.B. MCTS) zur algorithmischen Bewältigung kombinatorischer Optimierung (oft NP-vollständiger Probleme) zu finden,
- es muss hinreichende Rechenleistung – etwa für das Training komplexer KNN – verfügbar sein.

Allgemein gilt für ›KI‹-Systeme weiterhin: Grob irreführend als ›künstlich intelligent‹ bezeichnete, *de facto* nur adaptive Computersysteme sind stets und ausschließlich das Ergebnis methodischer Kompetenz menschlicher Expert/-innen, deren Können und natürliche Intelligenz sie in Gestalt theoretischer Einsichten in zugrunde liegende Prozesse und raffiniert ausgeklügelter heuristischer Verfahren vergegenständlichen. Das gilt freilich für technische Artefakte, gleich welcher Komplexität, schon immer und manifestiert sich im Übrigen auch in der Berufsbezeichnung ›Ingenieur‹ wie im Aristotelischen *téchne* als der Kunst, etwas beruhend auf Fachwissen, Übung und Erfahrung herstellen zu können. Zudem ist, wie gezeigt, das Verhalten von KNN als adaptiven Automaten durch deren Struktur und die Vorschriften des ›Lern‹-Algorithmus zur passenden Veränderung der Gewichte vollständig determiniert (obgleich analytisch nicht mehr zu durchschauen).

3 Epistemische und ethische Anwendungsprobleme von ›KI‹-Algorithmen

Neben Herausforderungen und Schwierigkeiten der *Entwicklung* fortgeschrittener adaptiver Computersysteme bestehen aber auch auf Seiten der *Anwendung* große, wegen ihrer Besonderheit auch über den Einsatz herkömmlicher Systeme hinausweisende Probleme. Ihr Einsatz wirft gravierende ungelöste epistemische und ethische Fragen auf. Sich diesen Fragen zu stellen, ist umso dringlicher, je komplizierter und intransparenter Methoden der Modellierung von Zeichenprozessen und zugehörige algorithmische Verfahren werden und je leichter sie infolgedessen auch missbräuchlich verwendet oder manipuliert werden können (zu Details und Beispielen vgl. Autorengruppe 2018).

Infolge analytischer Intransparenz sind die Probleme adaptiver Systeme sowohl epistemischer wie ethischer Natur: Das Verhalten von ›KI‹-Algorithmen (hier v.a. KNN und Verfahren schließender Statistik) ist selbst für Entwickler/-innen aktuell weder im einzelnen durchschaubar (»inconclusive evidence«) noch im Nachhinein erklärbar (»inscrutable evidence«). Sie produzieren nur wahrscheinliche, daher stets unsichere Ergebnisse, deren Korrektheit und Validität nur schwer zu

beurteilen sind. KNN sind zudem, wie oben beispielhaft gezeigt, sehr störanfällig und leicht auszutricksen. Die Ergebnisse, die sie liefern, sind in hohem Maße von der Qualität der Eingabedaten abhängig, die aber meist ebenfalls unbekannt oder nur schwer einschätzbar ist (»misguided evidence«; vgl. Mittelstadt et al. 2016: 4f).

Nutzer/-innen können dem Verhalten und seinen Ergebnissen daher nur blind vertrauen – trotz der nicht aufhebbaren Unsicherheit. Das stellt sie in der »Koaktion« (vgl. Hubig in diesem Band) mit solchen Systemen vor beträchtliche Belastungen: Wie sollen sie sich solche adaptiven Systeme überhaupt aneignen, wie mit ihnen zweckmäßig und zielgerichtet koagieren, wenn diese sich in vergleichbaren Situationen jeweils anders und unerwartet verhalten? Das wäre ein eklatanter Verstoß gegen eine der Grundregeln der Mensch-Maschine-Interaktion, gegen die Forderung nach erwartungskonformem Verhalten (vgl. EN ISO 9241-11 Anforderungen an die Gebrauchstauglichkeit). Zugleich würden auf Seiten der Nutzer/-innen stets aufs Neue überzogene Erwartungen an die vermeintliche Leistungsfähigkeit der Systeme geschürt, mithin gar ihre Wahrnehmung der Wirklichkeit verändert (»transformative effects«, Mittelstadt et al. 2016: 5). Konfrontiert mit diesen Widersprüchen, unter dem Erwartungsdruck erfolgreicher Bewältigung ihrer Aufgaben einerseits und angesichts des Verlusts der Kontrolle über Arbeitsmittel mit undurchschaubarem Verhalten andererseits, würden sie unter dauerhaften psychischen Belastungen zu leiden haben (so bereits Norman 1994 im Hinblick auf ›Agenten‹ als frühen adaptiven Systemen).

Darüber hinaus stellen sich mit diesen Systemen auch Fragen nach der ethischen Verantwortbarkeit: Dürfen derart undurchschaubare und störanfällige Artefakte überhaupt in die Welt gesetzt werden, da ihr künftiges Verhalten nicht sicher vorhersehbar ist? Wer ist gegebenenfalls für eingetretene Schäden verantwortlich – Entwickler/-innen?, Betreiber/-innen? oder Nutzer/-innen? – und wie werden daraus entstehende Haftungsansprüche geregelt? Bisher getroffene oder sich abzeichnende Regelungen sind unbefriedigend und unzureichend.

Auch hier gelten seit langem diagnostizierte »Ironien der Automatisierung« (Bainbridge 1983) zugespitzt weiter: Von besonderer Bedeutung ist dabei die Ironie, dass mangels hinreichender Übung und Erfahrung bei automatischem Normalbetrieb ausgerechnet die im Stör- oder Versagensfall wiederum benötigte menschliche Handlungskompetenz schwindet. Es fehlen Ansätze, wie dieser Art ›erlernter Inkompetenz‹ entgegengewirkt werden kann. In Schadensfällen wird die Ursache meist dichotomisch in ›menschlichem Versagen‹ oder einer technischen Störung gesucht und dabei die wahre, in fehlgeleiteter soziotechnischer Systemgestaltung liegende Ursache ignoriert. Die derart systemisch bedingte mangelnde Kompetenz, in Verbindung mit der durch Untätigkeit oder Ablenkung geschwächten Vigilanz, führt dann bei plötzlich notwendigen Eingriffen zu beträchtlichen Problemen der Bewältigung:

- unzureichende Erfahrungen und verlernte Fähigkeiten können zu fehlerhaften Diagnosen und falschen oder riskanten Aktionen führen,
- es entstehen, wie die Empirie ergibt, lange Verzögerungszeiten von 7 bis 10 Sekunden bis zum Zurechtfinden in der unerwarteten und ungewohnten Situation und zur Rückgewinnung von Handlungsfähigkeit (ein 150 km/h schnelles Auto fährt in dieser Zeit rd. 400 m weit).

Beispiele und empirische Erkenntnisse zu diesen Herausforderungen gibt es aus der Forschung über Leitwartentätigkeiten, Flugführung oder automatisiertes Fahren zuhauf, gleichwohl wurden bislang wenig weiterführende Konsequenzen gezogen. Daher ist infolge dieser noch weitgehend ungelösten Schwierigkeiten im praktischen Einsatz und Gebrauch adaptiver Systeme in naher Zukunft mit beträchtlichen Verzögerungen der Entwicklung zu rechnen (vgl. Bainbridge 1983, Baxter et al. 2012, Casner/Hutchins/Norman 2016, Casner et al. 2016, DIVSI 2016, Weyer 2007).

4 Historischer Exkurs: Meilensteine der Entwicklung von Computertechnik und Computing Science

Die hier angesprochenen Schwierigkeiten und Herausforderungen von Entwicklung und Anwendung fortgeschrittener adaptiver Computersysteme werden im landläufigen Verständnis von Computertechnik weitgehend ignoriert. Dadurch werden Fallstricke der Realisierung verkannt, Potenziale maßlos übertrieben und gesellschaftliche Folgen falsch eingeschätzt. Dies ist aber beileibe keine neue Erscheinung, sondern begleitet die Computertechnik seit dem Beginn ihrer materiellen Manifestation, die gemeinhin mit rüstungstechnischen Entwicklungen im und kurz nach dem 2. Weltkrieg angesetzt werden. Seither beherrschen prinzipiell irreführende, aber euphorisierende Metaphern wie ›Elektronengehirn‹, ›künstliche Intelligenz‹, ›maschinelles Lernen‹ oder ›autonome Systeme‹ bis hin zu den eingangs zitierten Äußerungen den gesellschaftlichen, oft aber auch den wissenschaftlichen Diskurs. Nur gelegentlich, in Zeiten großer Ernüchterung angesichts wirklicher Probleme der Modellierung und Algorithmisierung von Zeichenprozessen sozialer Praxis, ist weit zutreffender etwa von ›elektronischer Datenverarbeitung‹ die Rede.

In der ansonsten irreführenden, durchweg anthropomorphisierenden Metaphorik kommt aber ein grundsätzlich fehlgeleitetes Verständnis von Computertechnik zum Ausdruck. Computersysteme führen, wie die Theorien der Computing Science lehren, mittels Daten als auf Syntax reduzierten Zeichen berechenbare Funktionen aus und sonst nichts. Ihr Verhalten ist durch die implementierten Algorithmen und eingegebene Daten vollständig determiniert (nach dem Modell der Turingmaschine). Buchstäblich ›wissen‹ sie nicht, was sie tun. Intelligent sind daher nicht die Computersysteme, sondern ausschließlich ihre Entwickler/-innen, die zuvor in mühevoller Arbeit mit ihrer Kreativität und methodischen Kompetenz Zeichenprozesse sozialer Praxis nach gewünschten Anforderungen modelliert, formalisiert und in berechenbare Funktionen überführt haben, oder ggf. auch deren Nutzer/-innen, die durch Aneignung ihrer Funktionen damit etwas Sinnvolles anzustellen vermögen (nähere Einzelheiten hierzu finden sich, gestützt auf den triadischen Zeichenbegriff von Peirce (1983), bei Brödner 2008, 2018 sowie Nake 2001). Diese professionelle Sicht wurzelt in der geschichtlichen Entwicklung von Computertechnik und Computing Science (die Bezeichnung ›Informatik‹ ist ebenfalls eine irreführende Fehlbenennung: es geht um Verarbeitung von Daten, nicht von ›Information‹; vgl. Brödner 2014), wie sie in ihren logischen und konzeptionellen historischen Meilensteinen zum Ausdruck kommt (vgl. Übersicht 1).

Tatsächlich beginnt die Entwicklung nicht erst im 2. Weltkrieg, sondern bereits in der Frühphase der industriellen Revolution mit der hochgradig arbeitsteiligen Organisation von Kopfarbeit, mittels derer anspruchsvolle kognitive Aufgaben (etwa die Neuberechnung umfangreicher mathematischer Tafeln im Dezimalsystem, z.B. Logarithmen, nautische Almanache, Artillerie-Schusstafeln) mittels Formularen in eine geplante Abfolge einfachster Rechenoperationen (Addition und Subtraktion reeller Zahlen mit Hilfe mechanischer Rechenmaschinen) aufgelöst und zur Ausführung vorgeschrieben werden. Dies geschieht in enger ideeller Verzahnung mit der arbeitsteiligen Organisation von Handarbeit in kleinste wiederkehrende Verrichtungen und ihrer anschließenden Mechanisierung mittels Arbeitsmaschinen (z.B. Textilmaschinen, Werkzeugmaschinen). Im Laufe der Entwicklung werden Funktionen des *Antriebs* (›Kraftmaschinen‹), der *Werkzeugführung* und der *Steuerung* zunehmend voneinander getrennt; dadurch werden für Antriebe fossile Energie (Dampfmaschinen mit Transmission, später Einzelantrieb mit Elektromotoren) nutzbar und Steuerungen realisiert, die statt Kräften Signale (Daten) über maschinelle Bewegungen verarbeiten, um deren gewünschten Ablauf zu gewährleisten. In dieser Perspektive lassen sich Computer auch als *universelles Steuerungspotenzial* begreifen, das per Programm in eine spezifische Steuerung verwandelt wird (wie es heute bei digitaler Prozesssteuerung üblich ist).

Die arbeitsteilige Organisation spezialisierter Verrichtungen erfordert zunehmend aufwendigere Planung und sachliche wie zeitliche Koordination der Einzelarbeiten durch ›Manager/-innen‹. Wachsender Aufwand für Koordination und Sicherung der Herrschaft über immer kompliziertere Prozesse der (Massen-)Produktion erfordern zudem Maßnahmen vereinfachender Standardisierung sowie wissensbasierter Planung, Anweisung und Kontrolle (Taylors Prinzipien des »Scientific Management«). Diese vertikale Arbeitsteilung der Trennung von Planung und Ausführung beruht auf expliziten Beschreibungen von Produkten und Prozessen, führt mithin zu einer ›Verdoppelung‹ der Produktion in Zeichen (in Form von Zeichnungen, Stücklisten, Arbeitsplänen etc.). Das resultiert insgesamt in fortschreitender Verwissenschaftlichung von Produktion: Mit der Analyse, Planung und Kontrolle von Produktionsprozessen wird laufend erweitertes explizites Wissen über sie gewonnen und mit anderen wissenschaftlichen Erkenntnissen kombiniert. Dieses Wissen wächst wie ein Baum durch Verzweigung und wird durch Zeichen repräsentiert. So entstehen mit der ›Verdoppelung‹ der Arbeitswelt in Zeichen auch zunehmend durch Zeichen repräsentierte Arbeitsgegenstände und ebensolche Methoden ihrer Verarbeitung. Die Anwendung dieses expliziten propositionalen Wissens zur Lösung wirklicher praktischer Probleme erfordert allerdings in wachsendem Maße wiederum Können, Wissensteilung und Kooperation von spezialisierten Wissensarbeiter/-innen.

Übersicht 1: Meilensteine der Entwicklung von Computertechnik und Computing Science

1792-1801:	*Gaspard de Prony* entwickelt ein formularbasiertes Verfahren zur extrem arbeitsteiligen Neuberechnung mathematischer Tafeln im Dezimalsystem; das Formular-Schema der Abfolge einfacher Rechenoperationen bildet die Urform eines Algorithmus (noch im 2. Weltkrieg wurden V2-Flugbahnen so berechnet).
1805:	*Jacquard-Webstuhl*, erste digital mittels ›Lochbrettern‹ gesteuerte Arbeitsmaschine.
1812:	*Charles Babbag*e konzipiert die *Difference Engine* zur Berechnung der Funktionswerte von Polynomen: $f(x) = a_n x^n + \ldots + a_1 x + a_0$ (teils öffentlich gefördert, 1822 realisiert).
Um 1830:	*Charles Babbage* entwirft und programmiert die allgemeiner verwendbare *Analytical Engine.* Sie nimmt die *von-Neumann-Architektur* programmierbarer Universalrechner (Rechenwerk, Speicher, Steuerung, Datenein- & -ausgabe) vorweg, scheitert aber an der mechanischen Realisierung.
1847:	*George Boole* publiziert einen Logikkalkül (um 1888 von *Peano* als *Boolesche Algebra* axiomatisiert); er bildet das logisch-funktionale Fundament für *binäre Schaltsysteme* (als Kern heutiger Computer-Hardware).
1860-1880:	*Charles S. Peirce* entwickelt erstmals einen Prädikatenkalkül 1. Stufe, arbeitet an ›logischen Maschinen‹ und entwickelt die bislang elaborierteste *Theory of Signs* (triadische Zeichentheorie, ohne die computerisierte Kopfarbeit gar nicht zu verstehen wäre).
1931:	*Kurt Gödel* beweist die *Unvollständigkeit* formaler Systeme wie das der *principia mathematica* von A.N. Whitehead & B. Russell.
1936:	*Alan Turing* publiziert die Idee der *Turingmaschine* und definiert damit formal die Begriffe *Algorithmus* und *berechenbare Funktion* (äquivalent: *Lambda-Kalkül* von *Church* & *Kleene* 1936).
1945:	*Konrad Zuse* entwickelt den *Plankalkül* als erste algorithmische Programmiersprache (in Anlehnung an den *Lambda-Kalkül*).

Erst auf der Grundlage dieser Entwicklungsgeschichte der Computertechnik lässt sich ermessen, wie und warum Computer ihren Siegeszug durch die Arbeitswelt antreten konnten, sobald erst einmal mit elektro-mechanisch, später elektronisch realisierten binären Schaltsystemen die passende Hardware zur Verarbeitung binär codierter Daten als auf Syntax reduzierten Zeichen gefunden war. Im Zuge der ganzen Entwicklung bilden bis heute die logisch-konzeptionellen Ideen der Software mit den auf Erfahrung, Kreativität und Können ihrer Entwickler/-innen, auf deren »lebendiges

Arbeitsvermögen« (Pfeiffer 2004) angewiesenen Modellierungsmethoden und algorithmischen Verfahren den führenden Faktor und die Leistung der Hardware das limitierende Nadelöhr.

5 Prinzipielle Grenzen und Missverständnisse der Computertechnik

In diesem Zusammenhang ist zunächst an prinzipielle Grenzen der Formalisierung von Zeichenprozessen zu erinnern. Selbst die äußerst formalisierte Mathematik widersetzt sich ihrer vollständigen Algorithmisierung. Ausgerechnet im Zusammenhang mit den zu Beginn des 20. Jahrhunderts bestehenden großen Hoffnungen auf eine vollständige Formalisierung der bekannten Mathematik hat sich herausgestellt, dass es erwiesenermaßen unmöglich ist,

- einen Algorithmus anzugeben, der alle Sätze eines formalen Systems abzuleiten und deren Widerspruchsfreiheit zu zeigen imstande ist (vgl. Gödel 1931);
- einen Algorithmus anzugeben, der von jeder Formel eines formalen Systems entscheiden kann, ob diese Formel ein wahrer Satz des Systems ist (vgl. Turing 1936).

Bezeichnenderweise beruht der Beweis von Gödel im Kern darauf, dass er als erfahrener und kompetenter Mathematiker eine Formel im System so zu konstruieren vermag, dass sie über einen durch ihn als wahr erkannten Satz aussagt, nicht beweisbar (ableitbar) zu sein. Zur mathematischen Fähigkeit von Menschen gehört eben auch, dass sie über alles, was sie mit deren Hilfe formalisieren können, durch Nachdenken über die Formalisierung mittels abduktiven Schließens über sie hinaus zu gelangen vermögen (vgl. Brödner 1997, 2018).

Als rein formales Verfahren gibt die Abfolge von Operationen eines Algorithmus zwar Auskunft auf die Frage, was genau operativ abläuft; sie beantwortet aber nicht die Frage nach deren Sinn oder Bedeutung, warum sie so abläuft – eben deshalb gehört zur Software auch die Dokumentation mit derartigen Meta-Aussagen. Mit rein formalen Mitteln gelingt es eben nicht, auf der Metaebene Aussagen über Terme auf der operativen Ebene zu machen. Operationsfolgen sagen nichts über sich selbst aus, etwa ob sie korrekt oder gebrauchstauglich sind. So ist etwa auch die Frage, ob ein Algorithmus terminiert, formal nicht entscheidbar.

Das aus dem geschichtlichen Werdegang von Computertechnik und Computing Science gewonnene Verständnis der Funktionsweise von Computern als semiotischen Maschinen erlaubt zudem, die fundamentalen Unterschiede zu Menschen als lebendigen Organismen aufzuzeigen (vgl. Übersicht 2). Die ständige Rede von ›künstlich intelligenten‹ oder gar ›autonomen‹ Computersystemen entpuppt sich dabei als folgenreicher Etikettenschwindel. Wieder einmal bedarf es der Philosophie als »Kampf gegen die Verhexung unseres Verstandes durch die Mittel unserer Sprache« (Wittgenstein 1984: PU 109).

Die irreführende Metaphorik über Computer, als ob diese ›wie Menschen‹ intentional eingestellt und handlungsfähig wären – ›autonom‹, ›selbstorganisiert‹, ›intelligent‹, ›smart‹, ›selbstlernend‹, ›selbstheilend‹ etc. –, ignoriert in reduktionistischer Weise nach Denkmustern des Funktionalismus die fundamentalen Unterschiede. Dadurch wird kompetentes Handeln von Menschen auf algorithmisch gesteuertes Verhalten von Maschinen reduziert; zugleich entstehen eben dadurch Illusionen über deren Zustandekommen und tatsächliche Leistungsfähigkeit. Das führt im Ergebnis zu einer verbreiteten Selbsttäuschung, wie sie in Diskursen um ›künstliche Intelligenz‹ zum Ausdruck kommt (Brödner 2018). In den Wahnvorstellungen vom vermeintlichen Eigenleben der Maschinen äußert sich deren Fetischcharakter, die »Macht der Machwerke über die Machenden« (Haug 2005: 162).

Übersicht 2: Ontologische Differenz zwischen Mensch und Computer

Mensch (lebendiger Organismus)	**Computer** (semiotische Maschine)
Sich durch *Autopoiese* in Stoffwechsel und Kommunikation *selber machend*.	Wissensbasiert für bestimmte Zwecke *gemacht (konstruiert)*.
Autonom (selbstbestimmte Regeln).	*Automatisch* (auto-operational, selbsttätig).
Handelt intentional (kontingent).	Verhält sich *kausal determiniert*;
Ist sprachbegabt, *reflektiv lernfähig*.	ggf. algorithmisch gesteuert *Umwelt-adaptiv*.
Lebendiges *Arbeitsvermögen*: *Können* (implizites Wissen, Erfahrung, situierte Urteilskraft & Handlungskompetenz), *verausgabt* & *reproduziert* sich im Gebrauch.	Algorithmisch determiniertes *Verhalten*: Setzt *Formalisierung* von Zeichenprozessen voraus, muss für Praxis *angeeignet* & *organisatorisch eingebettet* werden.

6 Fazit: Grenzen und Widersprüche adaptiver Systeme

Aus diesen Ausführungen können mit Blick auf Grenzen und Widersprüche der Entwicklung und Anwendung fortgeschrittener adaptiver Computersysteme unmittelbar verschiedene Schlussfolgerungen gezogen werden:

Erstens bestimmen sog. ›autonome Systeme‹ die Regeln ihres Verhaltens nicht selbst, folglich sind sie tatsächlich nicht autonom, sondern als adaptive Automaten konstruiert.

Zweitens liegt es in der Regel-Natur von Spielen, ihrer Natur als mathematischem Objekt, dass Algorithmen – bei hinreichender Leistung der Hardware – menschlichen Spieler/-innen überlegen sind. Aus dieser Besonderheit kann aber nicht allgemein auf die Überlegenheit maschineller Verfahren über menschliche kognitive Kompetenz geschlossen werden.

Drittens gelingt es menschlicher Kreativität immer wieder, für spezielle, auch schwierige Aufgaben der Zeichenverarbeitung algorithmische Lösungsverfahren zu finden und als adaptive Automaten zu realisieren, die aber nur sehr begrenzt auf andere Aufgaben übertragbar sind. Meist ist die aufwendige Entwicklung jeweils eigener Methoden erforderlich, oft mit mehr Aufwand als Nutzen. Sog. ›KI‹-Verfahren sind daher keine »General Purpose Technology«, wie das Brynjolfsson et al. (2017: 19f) leichtfertig behaupten.

Viertens beruhen ironischerweise die Strukturen und Algorithmen erfolgreich eingesetzter adaptiver Systeme – mangels theoretischer Einsichten in Ursache-Wirkungs-Ketten – ausschließlich auf der Erfahrung und Kreativität, mithin dem Können bzw. dem Arbeitsvermögen ihrer Entwickler/-innen.

Fünftens ist das Verhalten adaptiver Systeme aus gleichem Grund nur schwer oder gar nicht zu durchschauen oder zu erklären, zudem äußerst störanfällig. In Form von KNN oder Verfahren schließender Statistik liefern sie stets nur wahrscheinliche, daher prinzipiell unsichere Ergebnisse. Das macht instrumentelles Handeln mit ihnen schwierig bis unmöglich, jedenfalls psychisch hoch belastend – gefordert sind daher Transparenz und Kontrolle ihrer algorithmischen Funktionen durch unabhängige Institutionen (wie bei anderer hoch riskanter Technik auch).

Statt auf illusionäre ›KI‹-Hoffnungen zu setzen, erscheint es angesichts dieser Einsichten in die zweifelhafte Tragfähigkeit von Konzepten ›künstlicher Intelligenz‹ weit aussichtsreicher, höhere Flexibilität, Produktivität und Innovationsfähigkeit stattdessen durch soziotechnische Gestaltung »guter Arbeit« zu erreichen. Dazu notwendiges Wissen ist aus über drei Jahrzehnten Forschung zur Gestaltung von Arbeit und Technik verfügbar. Dies erscheint umso notwendiger, je mehr die künftige gesellschaftliche Entwicklung auf die breite Entfaltung menschlichen Arbeitsvermögens

für den produktiven Umgang mit komplexen Beständen explizten Wissens und technischen Systemen angewiesen ist. Gestützt auf dieses Wissen ließe sich das immer bedeutender werdende lebendige Arbeitsvermögen durch Organisation reflexiven und kreativen Zusammenwirkens kompetenter Expert/-innen mit gebrauchstauglich gestalteten Computersystemen weit wirksamer als durch den Einsatz von ›KI‹-Systemen zur Entfaltung bringen.

Literatur

Autorengruppe (2018): The Malicious Use of Artificial Intelligence: Forecasting, Prevention, and Mitigation, Oxford (AR): Future of Humanity Institute u.a. 02/2018, https://arxiv.org/pdf/1802.07228.pdf

Babbage, Charles (1832): On the Economy of Machinery and Manufactures, London: Knight, Reprint New York: Kelley 1971 (deutsch: Die Ökonomie der Maschine, Nachdruck der Originalübersetzung von 1833, hg. von P. Brödner, Berlin: Kulturverlag Kadmos 1999)

Bainbridge, Lisanne (1983): »Ironies of Automation«, in: Automatica 19 (6), S. 775-779.

Bastian, Matthias (2018): »Google-Chef: Künstliche Intelligenz ›wichtiger als Feuer und Elektrizität‹ «, in: Vrodo vom 20.01.2018, https://vrodo.de/google-chef-kuenstliche-intelligenz-wichtiger-als-feuer-und-elektrizitaet/

Baxter, Gordon/Rooksby, John/Wang, Yuanzhi/Khajeh-Hosseini, Ali (2012): »The Ironies of Automation ... still going strong at 30?«, in: Phil Turner/Susan Turner (Hg.): European Conference on Cognitive Ergonomics, ECCE '12, Edinburgh, United Kingdom, August 28 - 31, 2012, S. 65-71.

Bögeholz, Harald (2016): »Mysteriöse Tiefe. Wie Google-KI den Menschen im Go schlagen will«, in: c't 6/2016, S. 148-151.

Brödner, Peter (2018): »Coping with Descartes' Error in Information Systems«, AI & Society Journal of Knowledge, Culture and Communication 2018 (online first).

Brödner, Peter (2014): »Durch „Information" desinformiert? Zur Kritik des Paradigmas der Informationsverarbeitung«, Arbeits- und Industriesoziologische Studien 7 (1), S. 42-59

Brödner, Peter (2008): »Das Elend computerunterstützter Organisationen«, in: Dorina Gumm/Monique Janneck/Roman Langer/Edouard J. Simon (Hg.): Mensch –Technik – Ärger? Zur Beherrschbarkeit soziotechnischer Dynamik aus transdisziplinärer Sicht, Münster: Lit-Verlag, S. 39-60.

Brödner, Peter (1997): Der überlistete Odysseus. Über das zerrüttete Verhältnis von Menschen und Maschinen, Berlin: edition sigma.

Browne, Cameron/Powley, Edward/Whitehouse, Daniel/Lucas, Simon/Cowling Peter I./Rohlfshagen, Philipp/ Taverner, Stephen/Perez, Diego/Samothrakis, Spyridon/Colton, Simon (2012): »A Survey of Monte Carlo Tree Search Methods«, in: IEEE Transactions on Computational Intelligence and AI in Games 4 (1), S. 1-49.

Brynjolfsson, Erik/Rock, Daniel/Syverson, Chad (2017): Artificial Intelligence and the Modern Productivity Paradox: A Clash of Expectations and Statistics, NBER Working Paper No. 24001.

Casner, Stephen M./Geven, Richard W./Recker, Matthias P./Schooler, Jonathan W. (2014): »The Retention of Manual Flying Skills in the Automated Cockpit«, in: Human Factors 56 (8), S. 1506-1516.

Casner, Stephen M./Hutchins, Edwin L./Norman, Don (2016): »The Challenges of Partially Automated Driving«, CACM 59 (5), S. 70-77.

Deutsches Institut für Vertrauen und Sicherheit im Internet (DIVSI) (2016): Digitalisierte urbane Mobilität. Datengelenkter Verkehr zwischen Erwartung und Realität, Hamburg: Deutsches Institut für Vertrauen und Sicherheit im Internet.

Dworschak, Manfred (2018): »Was künstliche Intelligenz schon leisten kann - und was nicht«, in: Der Spiegel 2/2018

Gödel, Kurt (1931): Über formal unentscheidbare Sätze der *principia mathematica* und verwandter Systeme I, Monatshefte für Mathematik und Physik 38 (1), S. 173-198.

Haug, Wolfgang F. (2005): Vorlesungen zur Einführung ins »Kapital«, Hamburg: Argument.

Hilberg, Wolfgang (1995): »Karl Steinbuch – ein zu Unrecht vergessener Pionier der künstlichen neuronalen Systeme«, in: Frequenz 49 (1-2), S. 28-36.

LeCun, Yann/Bengio, Yoshua/Hinton, Geoffrey (2015): »Deep Learning«, in: Nature 521, S. 436-444.

LeCun, Yann/Bottou, Leon/Bengio, Yoshua/Haffner, Patrick (1998): »Gradient-Based Learning Applied to Document Recognition«, Proceedings of the IEEE 11, S. 2278-2324.

McCulloch, Warren S/Pitts, Walter H. (1943): A Logical Calculus of the Ideas Immanent in Nervous Activity, Bulletin of Mathematical Biophysics 5, S. 115-133.

Metzen, Jan H./Kumar, Mummadi C./Brox, Thomas/Fischer, Volker. (2017): Universal Adversarial Perturbations Against Semantic Image Segmentation, arXiv:1704.05712v1, https://arxiv.org/pdf/1704.05712v1.pdf

Minsky, Marvin/Papert, Seymour (1969): Perceptrons, Cambridge, MA: MIT Press

Mittelstadt, Brent D./Allo, Patrick/Taddeo, Mariarosaria/Wachter, Sandra/Floridi, Luciano (2016): »The Ethics of Algorithms: Mapping the Debate«, in: Big Data & Society 3 (2), S. 1-21.

Nake, Frieder (1992): »Informatik und Maschinisierung von Kopfarbeit«, in: Wolfgang Coy/Frieder Nake/Jörg-Martin Pflüger/Arno Rolf/Jürgen Seetzen/Dirk SiefkesReiner Stransfeld. (Hg.): Sichtweisen der Informatik, Braunschweig/ Wiesbaden: Vieweg, S. 181-201.

Nguyen, Anh/Yosinski, Jason/Clune, Jeff (2015): »Deep Neural Networks are Easily Fooled: High Confidence Predictions for Unrecognizable Images«, in: Computer Vision and Pattern Recognition (15), IEEE, https://arxiv.org/pdf/ 1412.1897.pdf.

Norman, Donald A. (1994): »How Might People Interact with Agents«, in: CACM 37 (7), S. 68-71.

Ortega y Gasset, José (1949): Betrachtungen über die Technik, Stuttgart: DVA

Peirce, Charles S. (1983): Phänomen und Logik der Zeichen, Frankfurt/M: Suhrkamp (engl. Original: A Syllabus of Certain Topics of Logic, Boston Alfred Mudge & Son, 1903).

Recke, Martin (2016): »Invasion der Roboter: Künstliche Intelligenz ist bald so normal wie Strom«, in: t3n vom 13.03.2016, https://t3n.de/news/sxsw-traurige-roboter-688398/

Ropohl, Günter (1991): Technologische Aufklärung. Beiträge zur Technikphilosophie, Frankfurt/M: Suhrkamp.

Schäfer, Ulrich (2016): »Leere Büros, leere Fabriken«, in: Süddeutsche Zeitung vom 10.11.2016, http:// www.sueddeutsche.de/wirtschaft/kuenstliche-intelligenz-leere-bueros-leere-fabriken-1.3243399

Schmidhuber, Jürgen (2015): »Deep Learning in Neural Networks. An Overview«, in: Neural Networks 61, S. 85-117.

Sharif, Mahmood/Bhagavatula, Sruti/Bauer, Lujo/Reiter, Michael K. (2016): Accessorize to a Crime. Real and Stealthy Attacks on State-of-the-Art Face Recognition,in: CCS '16 Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, https://users.ece.cmu.edu/~mahmoods/publications/ccs16-adv-ml.pdf (Abruf: 18.02.2018).

Silver, David/Huang, Anja/Maddison, Chris J./Guez, Arthur/Sifre, Laurent/van den Driessche, George/Schrittwieser, Julian/Antonoglou, Ioannis/Panneershelvam, Veda/Lanctot, Marc/Dieleman, Sander/Grewe, Dominik/Nham, John/ Kalchbrenner, Nal/Sutskever, Ilya/Leach, Madeleine/Kavukcuoglu, Koray/Graepel, Thore/Hassabis, Demis (2016): »Mastering the Game of Go with Deep Neural Networks and Tree Search«, in: Nature 529, S. 484-489.

Simon, Hubert A./Newell, Allen (1972): Human Problem Solving, Englewood Cliffs (NJ): Prentice Hall.

Sitawarin, Chawin/Bhagoji, Arjun N./Mosenia, Arsalan/Chiang, Mung/Mittal, Prateek (2018): DARTS: Deceiving Autonomous Cars with Toxic Signs, arXiv:1802.06430v1, https://arxiv.org/pdf/1802.06430.pdf

Sulleyman, Aatif (2017): »Stephen Hawking warns Artificial Intelligence ›may replace humans altogether‹ «, in: Independent vom 02.11.2017, https://www.independent.co.uk/life-style/gadgets-and-tech/news/stephen-hawking-artificial-intelligence-fears-ai-will-replace-humans-virus-life-a8034341.html

Szegedy, Christian/Zaremba, Wojciech/Sutskever, Ilya/Bruna, Joan/Erhan, Dumitru/Goodfellow, Ian/Fergus, Rob (2014): Intriguing Properties of Neural Networks, arXiv:1312.6199v4, https://arxiv.org/pdf/1312.6199.pdf

Trinkwalder, Andrea (2016): »Netzgespinste. Die Mathematik neuronaler Netze: Einfache Mechanismen, komplexe Konstruktion«, in: c't 6/2016, S. 130-135.

Turing, Alan (1936): »On Computable Numbers. With an Application to the Entscheidungsproblem«, in: Martin Davis(Hg.): The Undecidable: Basic Papers on Undecidable Propositions, Unsolvable Problems, and Computable Functions, New York: Raven 1965, S. 116-151.

Verein Deutscher Ingenieure (Hg.) (1991): Technikbewertung – Begriffe und Grundlagen. Erläuterungen und Hinweise zur VDI-Richtlinie 3780, VDI Report 15, Düsseldorf: VDI.

Weyer, Johannes (2007): »Autonomie und Kontrolle. Arbeit in hybriden Systemen am Beispiel der Luftfahrt«, in: Technikfolgenabschätzung – Theorie und Praxis 16 (2), S. 35-42.

Wick, Christoph (2017): Deep Learning, Informatik Spektrum 40 (1), S. 103-107.

Wittgenstein, Ludwig (1984): Philosophische Untersuchungen. Werkausgabe Bd. 1, Frankfurt/M: Suhrkamp: S. 225-580.

Yuandong Tian/Yan, Zhu (2016): Better Computer Go Player with Neural Network and Long-term Prediction, arXiv: 1511.06410v3, https://arxiv.org/pdf/1511.06410.pdf