White Paper

Künstliche Intelligenz

Innovationspark Zentralschweiz – Building Excellence
Melissa Kneubühler, Stephan Keller
BE!conference, 23. November 2018

1. Hintergrund & Zielsetzung

Über künstliche Intelligenz (KI) wird viel berichtet. Schlagzeilen reichen von "unbegrenztem Potenzial" bis zu "das Ende der Menschheit". Begriffe werden beliebig ausgetauscht und selten definiert. Dies führt zu Konfusion und Diskussionen, die am Ziel vorbeiführen. Technische Möglichkeiten vermischen sich mit philosophischen Metaphern. Hohe Erwartungen werden geweckt und Ängste geschürt. Das vorliegende White Papier beabsichtig daher, dem Thema ein Stück Struktur zu geben, in dem es Schlüsselbegriffe definiert, zentrale Konzepte ausführt und sowohl die Anwendungsmöglichkeiten als auch die Limitationen und Kontroversen darlegt. Es stellt augenscheinlich keine abschliessende Behandlung des Themas dar. Vielmehr soll das White Paper einen Gedankenanstoss geben und als Ausgangspunkt für eigene, weiterführende Recherchen der Leser dienen.

2. Definitionen

Das Thema künstliche Intelligenz (*Artificial Intelligence* im Englischen) hat nicht zuletzt auf Grund umstrittener Definitionen Kontroversen hervorgerufen. Im Kontext von Intelligenz taucht häufig das Konzept von Bewusstsein auf. Mit der Frage, was Bewusstsein ist, haben sich Philosophen bereits vor Urzeiten befasst. Eine einheitliche Definition konnte bis anhin nicht gefunden werden. Das vorliegende Papier schlägt daher vorerst die weitgefasste Definition "subjektiver Erfahrung" nach Tegmark (2017) als Arbeitshypothese vor. Ähnliche Herausforderungen zeigen sich beim Begriff der Intelligenz. Allgemein akzeptierte Vorschläge lauten wie folgt:

- Intelligenz ist die Fähigkeit, zu lernen und zu verstehen sowie mit neuen und herausfordernden Situationen umzugehen (Merriam-Webster Dictionary, n.d.)
- Intelligenz ist fähig, natürliche Limitationen zu durchbrechen und die Welt nach eigenem Bild zu transformieren (Kurzweil, 2012).
- Intelligenz ist keine Einzeldimension. Sie umfasst ein komplexes Konstrukt vielfältiger Informationsverarbeitungsfähigkeiten (Boden, 2016).
- Intelligenz ist die Fähigkeit, komplexe Ziele zu erreichen (Tegmark, 2017).

Künstliche Intelligenz zielt darauf ab, menschliche Intelligenz mittels Computern und Algorithmen (Computerprogrammen) nachzubilden. Das Speichern, Verarbeiten und Lernen von Informationen sind zentrale Eigenschaften (Tegmark, 2017). Gemäss Ng (2011) lernt ein Algorithmus von Erfahrung (E) in Bezug auf Task (T) und Performance (P), wenn die Performance von T, gemessen an P, sich mit E verbessert. Es wird dann von *Machine Learning* gesprochen. Machine Learning ist eine Subdisziplin der künstlichen Intelligenz und kommt zum Einsatz, wenn es darum geht, grosse Datenmengen zu verarbeiten (*Data Mining*), *self-*

customizing Programme zu erstellen oder Anwendungen zu programmieren, welche auf Grund hoher Komplexität nicht von Hand programmieren werden können (Ng, 2011).
Ein weiterer Begriff, der in diesem Zusammenhang auftritt, ist *Deep Learning.* Deep Learning ist ein Teilbereich des Machine Learnings. Er basiert auf neuronalen Netzen, welche sich am Modell des menschlichen Gehirns orientieren. Deep Learning folgt einem konnektivistischen Ansatz. Konnektivismus definiert Wissen in Form von Netzwerken und Lernen als Prozess der Mustererkennung (AlDahdouh, Osorio & Caires, 2015).
Abbildung 1 stellt das Verhältnis von künstlicher Intelligenz zu Machine Learning und Deep Learning grafisch dar. Weitere Ausführungen entnehmen sich den nachfolgenden Kapiteln.

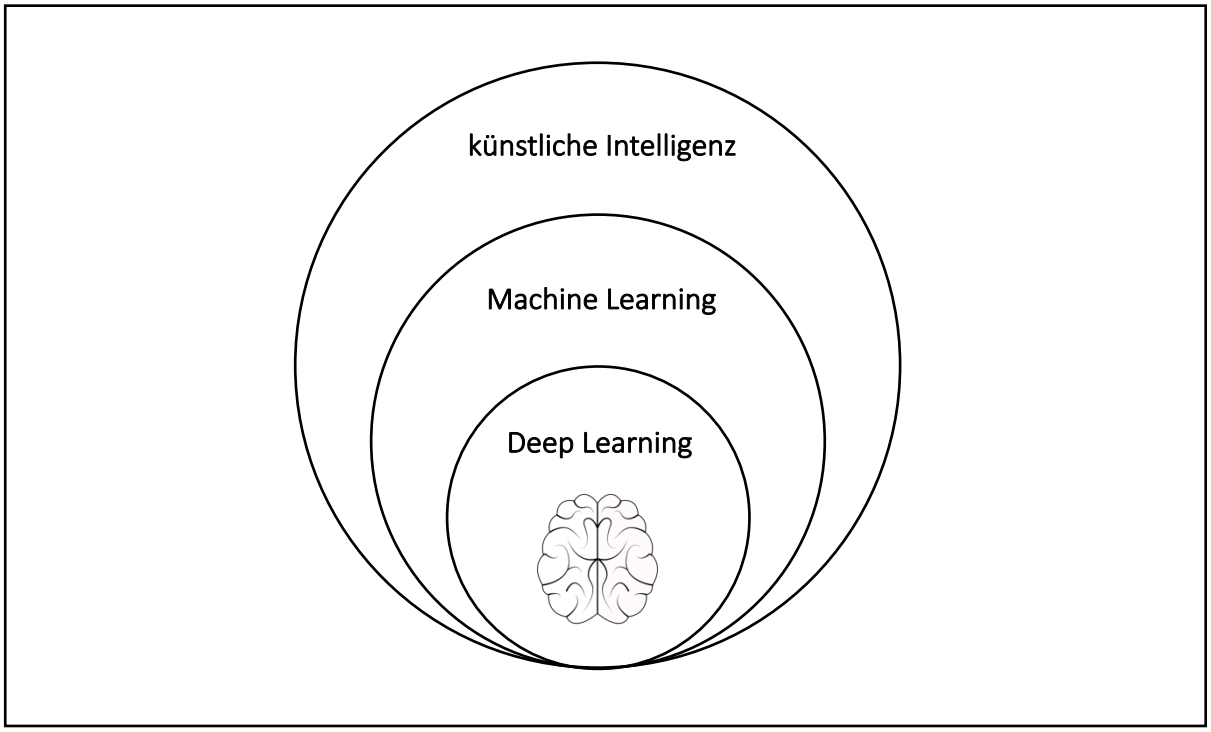

Abbildung 1. Verhältnis von künstlicher Intelligenz zu Machine Learning und Deep Learning.

3. Historische Entwicklung

Automata, Maschinen mit menschenähnlichen Fähigkeiten tauchten bereits vor über 2'500 Jahren in Geschichten und Mythen rund um den Globus auf (u.a. Homer, Aristoteles, Gellius, Mozi). Seither haben sie sowohl die abendländische als auch östliche Philosophie mitgeprägt und dienen als Metapher, wenn es um die Definition von "Menschsein" geht (u.a. Descartes, Leibniz, Condillac). Diverse Science Fiction Autoren und Filmemacher haben die Idee aufgegriffen und zu Meisterwerken verarbeitet (u.a. Shelley, Baum, Asimov). Gleichwohl haben die Automata Forschende im Bereich der künstlichen Intelligenz inspiriert (Nilsson, 2010).
Zu den Gründervätern der künstlichen Intelligenz zählt McCarthy zusammen mit Minsky, Newell und Simon. Im Sommer 1956 rief er zu einer zweimonatigen Konferenz in Dartmouth auf, um herauszufinden, ob es einer Maschine gelingen kann, so zu lernen, wie es ein Kleinkind tut – durch Versuch und Irrtum. McCarthy glaubte, dass Intelligenz etwas natürlich Existentes ist, das sich künstlich mittels Computern replizieren lässt (Roberts, 2016). Schnell musste er jedoch feststellen, dass er die Komplexität menschlicher Intelligenz unterschätzt hatte. Die gewünschten Ergebnisse blieben aus. Trotzdem legten McCarthy et al. den Grundstein für das Feld der künstlichen Intelligenz. Im Laufe der Zeit trugen Forschende aus diversen Disziplinen wie dem Ingenieurwesen (u.a. Wiener), der Biologie (u.a. Ashby, McCulloch, Pitt), der experimentellen Psychologie (u.a. Newell, Simon), der Kommunikationstheorie (u.a.

Shannon), der Spieltheorie (u.a. von Neumann, Morgenstern), der Mathematik und Statistik (u.a. Good), der Logik und Philosophie (u.a. Turing, Church, Hempel) und der Linguistik (u.a. Chomsky) zur Entwicklung des Felds bei (Buchanan, 2005).
Anfänglich verwendeten viele KI-Programme denselben Basisalgorithmus. Um ein gesetztes Ziel zu erreichen, gingen sie schrittweise vor, als würden sie sich in einem Labyrinth bewegen und kehrten immer wieder um, wenn sie in eine Sackgasse geraten waren[1]. Dies führte bei zunehmender Komplexität zu einer kombinatorischen Explosion. Die Anzahl möglicher Wege stieg ins Unermessliche, was die Grenzen verfügbarer Rechenleistung überstieg. Entwickler mussten Heuristik und Faustregeln anwenden (Russel & Norvig, 2003). Hohe Erwartungen trafen auf unterschätzte Komplexität. Resultate hielten nicht, was sie versprachen. Weiter erfuhr der heute so populäre Ansatz des Konnektivismus, welcher auf Rosenblatt (1958) zurückgeht, vernichtende Kritik seitens Minsky. Entsprechend stellte sich der Zufluss an Forschungsgeldern ein. Es folgte der "KI-Winter" (Russel & Norvig, 2003).
Trotz Herausforderungen blieb jedoch das Bestreben, eine Maschine mit menschenähnlichen Fähigkeiten zu erschaffen, bestehen. Schach, ein Spiel, das extensives und komplexes Denken erfordert, wurde als Basis herangezogen, um Repräsentations- und Inferenzmechanismen[2] zu studieren. 1997 gelang es dem *Deep Blue* Computer von IBM, den Schachweltmeister Kasparov zu schlagen (McCorduck, 2004). Deep Blue stützte sich auf den grössten Vorteil, den ein Computer gegenüber einem menschlichen Spieler hat: die Fähigkeit Berechnungen innert kürzester Zeit durchzuführen. Er simulierte jede Sekunde Millionen möglicher Schachzüge und wählte den Zug, der gemäss vorgängiger Programmierung zum bestmöglichen Ergebnis führte – ein Ansatz, den Computerwissenschaftler als *Brute Force* bezeichnen (Roberts, 2016). Dieser Ansatz wurde allerdings erst mit einer massiven Zunahme an Rechenleistung möglich.
Dank der Integration von Wahrscheinlichkeitsrechnung, Entscheidungstheorie, stochastischer Modellierung und klassischer Optimierung erlebten Machine und Deep Learning einen Aufschwung (Pearl, 1988). Präzise, mathematische Beschreibungen für neuronale Netze und evolutionäre Algorithmen etablierten sich (Russel & Norvig, 2003).
Roboter, welche häufig mit künstlicher Intelligenz assoziiert werden, kamen erstmals zum Einsatz, um Ideen intelligenten Verhaltens zu testen. Zuvor handelte es sich bei der Robotik in erster Linie um Maschinenbauthematiken (Buchanan, 2005). Als die Sensortechnik dazu kam, setzte sich der Begriff des intelligenten Agenten durch. Er beschreibt eine autonome Einheit, welche die Umwelt durch Sensoren beobachtet und durch Aktoren darauf reagiert (Russel & Norvig, 2003). Intelligente Agenten differenzieren sich deutlich gegenüber den menschenähnlichen Automata, die sich Denker vor über 2'500 Jahren ausgemalt hatten – insbesondere dadurch, dass sie sich ausschliesslich spezifischen Tasks widmen. Sie besitzen "nur" schwache künstliche Intelligenz (mehr dazu im Kapitel 4). Starke KI, welche diese Limitation durchbrechen und sich zu par mit menschlicher Intelligenz sehen würde, bleibt bis auf Weiteres ein Bestreben für die Zukunft. In den letzten Jahren hat vor allem das Interesse der Industrie am Einsatz künstlicher Intelligenz zur Steigerung von Effizienz und Effektivität in der Produktentwicklung, der Prozessoptimierung und dem Servicedesign zugenommen, was KI zu einem prominenten Thema gemacht hat.

4. Künstliche Intelligenz

Künstliche Intelligenz zielt darauf ab, menschliche Intelligenz nachzubilden. Die künstliche Intelligenz, mit der heute gearbeitet und interagiert wird, klassifiziert sich als "schwache" (*weak*

[1] Dieses Konzept wird in Fachkreisen als "Reasoning as Search" bezeichnet (Russel & Norvig, 2003).
[2] Wissensrepräsentation ist ein Teilbereich der künstlichen Intelligenz, der sich mit der Darstellung von Wissen beschäftigt, sodass dieses von Computern verwendet werden kann, um komplexe Aufgabenstellungen zu lösen. Sowohl Erkenntnisse aus der Psychologie als auch der Logik (Regeln, Beziehungen, Sets, Teilsets) fliessen mit ein (Schank & Abelson, 1977). Inferenz stellt aufbereitetes Wissen dar, das aus logischen Schlussfolgerungen gewonnen werden konnte – entweder via induktiver, deduktiver oder abduktiver Ansätze.

oder auch *narrow* im Englischen) KI. Schwache künstliche Intelligenz kann nur spezifische Tasks ausführen und bloss ein eng begrenztes Set an Zielen verfolgen, so wie das beispielsweise beim Schachspielen der Fall ist (Tegmark, 2017). Sie beruht auf einem spezifischen Datensatz. Bei Online-Firmen wie Google, Facebook und Amazon wirkt schwache KI im Hintergrund, indem sie Suchergebnisse strukturiert, Feeds organisiert und Kaufvorschläge macht. Ein Wissens- und Lerntransfer in andere Bereiche findet nicht statt. Doch gerade diese Fähigkeit ist es, die den Menschen ausmacht. Einige KI-Forschende sind daher bestrebt, eine höhere Form der KI, die sogenannte starke (*strong, general* oder auch *human-level*) künstliche Intelligenz, zu schaffen. Solch einer Form der KI wäre es möglich, in sämtlichen kognitiven Tasks mindestens genauso gute Ergebnisse zu erzielen wie der Mensch. Danach wäre es nach Auffassung einiger KI-Experten nur eine Frage der Zeit, bis sich die starke KI zur superintelligenten KI weiterentwickeln und menschliche Intelligenz übersteigen würde. Die Meinungen darüber, ob ein solches Szenario realistisch ist und welcher Zeithorizont dafür herangezogen werden müsste, gehen allerdings weit auseinander (Tegmark, 2017). Abbildung 2 fasst die drei möglichen KI-Stufen zusammen.

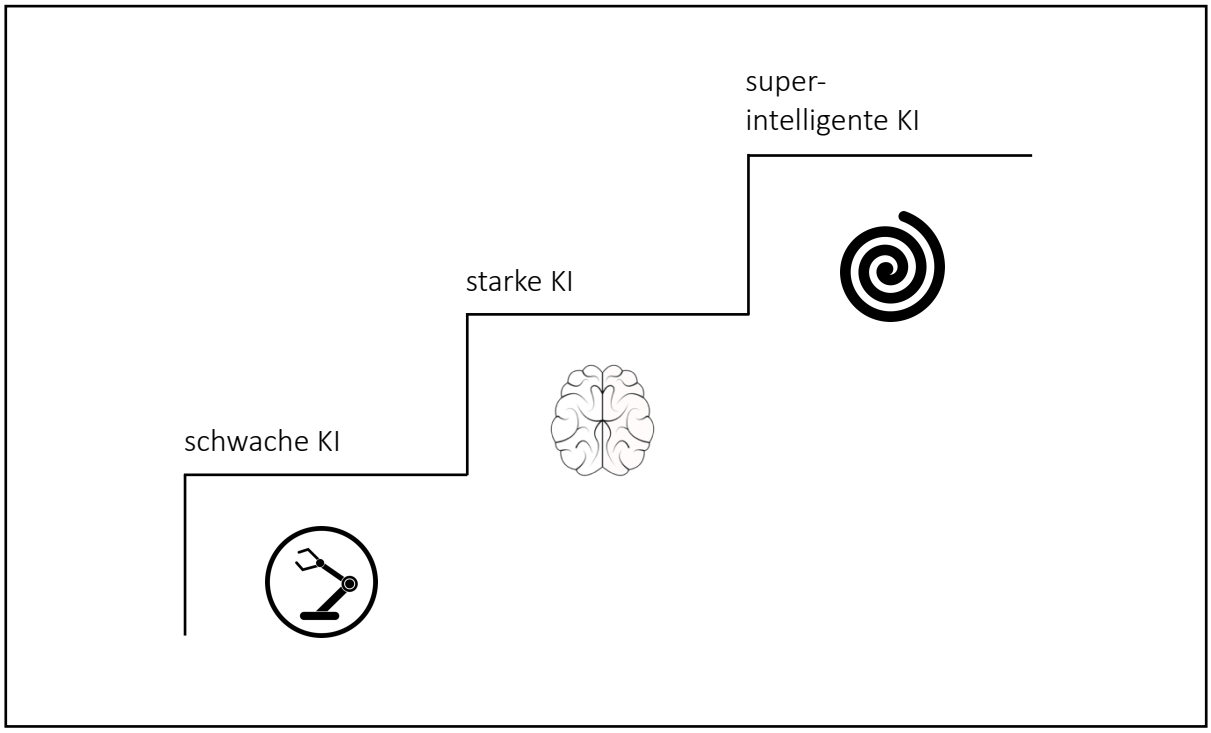

Abbildung 2. Schwache, starke und superintelligente KI.

Searle (1980) postulierte, dass starke Intelligenz Bewusstsein erfordert. Gleichzeitig vertrat er den Standpunkt des biologischen Naturalismus, welcher Bewusstsein als emergentes Phänomen des Gehirns respektive dessen neuronalen Schaltkreise sieht. Computerprogramme mit derselben funktionalen Struktur und dem gleichen Input-Output-Verhalten wie das menschliche Gehirn reichten ihm nach nicht aus, um Bewusstsein und in Konklusion starke künstliche Intelligenz zu erzeugen. Dafür müssten die Programme zusätzlich auf einer Architektur mit vergleichbarer kausalen Kraft wie die der Neuronen (Gehirnzellen) laufen. Demgegenüber steht die substratunabhängige Sichtweise, welche Intelligenz in der Information verortet. Illustrieren lässt sich dies am Beispiel eines Dokuments, das jemandem via E-Mail zum Druck zugestellt wird. Innerhalb kürzester Zeit kopiert sich die Information von Magnetisierungen der Festplatte zu elektrischen Ladungen des Arbeitsspeichers weiter zu Radiowellen im Drahtlosnetzwerk, Stromspannungen im Router, Laserimpulsen in der Glasfaser und schliesslich Molekülen auf dem Papier. Die verschiedenen Substrate sind notwendig für die Übermittlung, der wesentliche Aspekt stellt aber der substratunabhängige Dokumenteninhalt dar (Tegmark, 2017). Kurzweil (2012) argumentierte, dass die Stärke

neuronaler Verbindungen des Gehirns mittels rechnerischer Gewichtungen im Algorithmus abgebildet werden kann, so wie dies beispielsweise bei neuronalen Netzen der Fall ist (mehr dazu im Kapitel 6). Gelingt es, die Komplexität des menschlichen Gehirns auf diese Art zu imitieren, sollte nach Kurzweil Bewusstsein auch bei Computern als emergentes Phänomen auftreten, wodurch starke KI sich als realistisch erweisen würde. Nicht weiter ausgeführt werden an dieser Stelle Denkansätze, die Bewusstsein nicht als emergentes Phänomen physischer Strukturen sondern als eigenständige Komponente sehen (u.a. Dualismus, Idealismus, Animismus). Das würde den Umfang des vorliegenden Papiers übersteigen. Da die Funktionsweise des Gehirns inklusive der ausschlaggebenden Faktoren für das vermutlich emergente Phänomen Bewusstsein noch nicht abschliessend erforscht werden konnte (Damasio 2010), stellt sich eine Beurteilung, was die Möglichkeit einer starken und super-intelligenten KI betrifft, als wenig fundiert heraus. Die folgenden Kapitel widmen sich daher der angewandten schwachen künstlichen Intelligenz.

5. Machine Learning

Machine Learning findet statt, wenn sich die Performance eines Computerprogramms bezogen auf einen spezifischen Task (schwache KI) mit Erfahrung progressiv verbessert (Ng, 2011), ohne explizit programmiert zu werden (Samuel in Koza, Bennet, Andre & Keane, 1999). Im Machine Learning werden statistische Methoden verwendet. Voraussetzung ist eine quantitativ und qualitativ passende Datenbasis. Es gilt zwischen überwachtem (*supervised* im Englischen) und unüberwachtem (*unsupervised)* Lernen zu unterscheiden. Beim überwachten Lernen besteht der Datensatz aus Input- (*x*) und Output-Werten (*y*). Korrekte Labels sind vorhanden. Es geht darum, basierend auf vorliegenden Datenpaaren, Vorhersagen für Output-Werte künftiger Input-Werte zu treffen (Ng, 2011).

> Konkret ist ein Datensatz mit *m* Beispielen von Input-Output-Paaren gegeben:
>
> $(x_1, y_1), (x_2, y_2), \ldots (x_m, y_m)$
>
> Jedes y_j wurde von einer unbekannten Funktion $y = f(x)$ generiert[3].
> Das Ziel ist es, eine Funktion *h* zu finden, welche die wahre Funktion *f* approximiert.
> Die Funktion *h* ist eine Hypothese (Russel & Norvig, 2003).

Lernen ist die Suche nach jener Hypothese, die im Bereich der möglichen Hypothesen am besten performt – nicht nur am bestehenden Datensatz, sondern auch in Bezug auf künftige Beispiele. Das überwachte Lernen unterscheidet zwischen kontinuierlichen und diskreten Werten. Diskrete Werte finden sich vor allem bei Klassifizierungsproblemen wie beispielsweise der Bilderkennung. In simpler Form kommt beim Machine Learning für kontinuierliche Werte lineare Regression, für diskrete Werte logistische Regression zum Einsatz (Ng, 2011). Eine Hypothese "generalisiert" gut, wenn sie den Wert von *y* für neue Beispiele korrekt vorhersagt. Um dies zu prüfen wird *h* auf einen Testdatensatz, der vom Trainingsdatensatz abweicht, angewendet (Russel & Norvig, 2003). *E* (*Error*, Fehler) zeigt die Differenz zwischen dem Soll-Wert der wahren Funktion *f* und dem Schätzungswert der Hypothese *h* an. Die Fehler- respektive Kostenfunktion $J(\theta)$ ist zu minimieren[4]. Dazu bedient sich das Machine Learning dem Gradientenverfahren, dem Verfahren des steilsten Abstiegs. Metaphorisch lässt sich das Gradientenverfahren als Abstieg von einem Berg bei Dunkelheit beschreiben. Das Gelände und die Landschaft sind nicht erkennbar. Bei jedem Schritt muss abgetastet werden, wo sich der

[3] Unter gewissen Umständen ist die Funktion *f* stochastisch und keine strikte Funktion von *x*. Gelernt wird dann eine konditionelle Wahrscheinlichkeitsverteilung $P(Y \mid x)$ (Russel & Norvig, 2003).

[4] Lineare Regression: $J(\theta) = \frac{1}{2m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2$; m = Anz. Trainingspaare, θ = Gewichtungsparameter
Logistische Regression: $J(\theta) = -\frac{1}{m}\left[\sum_{i=1}^{m} y^{(i)} log\, h_\theta(x^{(i)}) + (1 - y^{(i)}) log\left(1 - h_\theta(x^{(i)})\right)\right]$ (Ng, 2011)

Boden am steilsten nach unten neigt, bis das Tal erreicht ist. Der Gradient entspricht der Bodenneigung. Das Gradientenverfahren erlaubt es, das Minimum von *E* zu finden, auch bei komplexen Funktionen, die sich algebraisch nicht ohne Weiteres abbilden lassen. Es eignet sich gut für Funktionen mit vielen Parametern, bei welchen *y* nicht nur von *x*, sondern *a, b, c,* etc. abhängig ist (Rashid, 2017). Die Parameter fliessen mit unterschiedlichen Gewichtungen (θ_j) in den Ouput-Wert *y* ein. Um herauszufinden, wie die zu minimierende Kostenfunktion mit den Parametergewichtungen zusammenhängt, wird die Differenzialrechnung eingesetzt. Der Algorithmus für das Gradientenverfahren lautet (Ng, 2011):

repeat until convergence {

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta j} J(\theta_0, \theta_1, \ldots \theta_n)$$

}

Alpha (α) steht für die Schrittgrösse (*learning rate*). Eine zu kleine Schrittgrösse führt zu langen Rechenzeiten. Bei einer zu grossen Schrittgrösse besteht das Risiko, das Minimum zu verfehlen (*overshooting).* In der Praxis wird der Algorithmus oft in vektorisierter Form implementiert. Dadurch verarbeitet der Algorithmus nicht nur Einzelwerte, sondern ganze Wertesets (Vektoren/Matrizen) gleichzeitig. Unterstützt die Hardware diesen Prozess, verbessert sich die Rechengeschwindigkeit signifikant.
Bei diskreten Werten, auf die logistische Regression angewendet wird[5], kommt häufig die Sigmoidfunktion[6] zum Einsatz. Sie berücksichtigt Schwellenwerte und wird daher auch als Aktivierungsfunktion bezeichnet. Sie hat eine S-Form und erlaubt Werte zwischen 0 und 1, schneidet die y-Achse bei $y = ½$. Der Output beschreibt die Wahrscheinlichkeit von $y = 1$ in Bezug auf Input *x*, was eine Klassifizierung des Outputs *y* erlaubt. Alternativ kann ein "harter Schwellenwert" mittels einer Stufenfunktion eingesetzt werden (Rashid, 2017).
Anders als beim überwachten Lernen beinhaltet der Datensatz im unüberwachten Lernen keine oder nur gleiche Labels. Das unüberwachte Lernen zielt auf eine Gruppierung (*Clustering*) von Werten ab. Ein populärer Algorithmus dieser Art ist der *K-means* Algorithmus. Er gruppiert ähnliche Datenpunkte zusammen und entdeckt unterliegende Muster. Dazu sucht der Algorithmus eine fixe Anzahl Cluster (*k*) in einem Datenset (Bulezyuk, n.d.). In einem ersten Schritt werden *K* Zentroiden nach dem Zufallsprinzip initialisiert. Iterative Berechnungen optimieren anschliessend die Positionen der Zentroiden. Der Algorithmus lautet (Ng, 2011):

zufällige Initialisierung der *K* Clusterzentroiden $\mu_1, \mu_2, \ldots \mu_K \in \mathbb{R}^n$
repeat {
    for $i = 1 : m$
        $c^{(i)}$: = index (von 1 zu *K*) der Clusterzentroiden mit kürzester Distanz zu $x^{(i)}$
    for $k = 1 : K$
        μ_K : = Mittelwert (*mean*) der Punkte, die dem Cluster *k* zugeteilt wurden
}

Der erste Teil dient der Clusterzuweisung. Der zweite Teil bewegt respektive optimiert die Zentroiden. *m* steht für die Anzahl Trainingsbeispiele, wobei $K < m$ gelten muss. Die Anwendungsfälle sind vielfältig und reichen von Verhaltenssegmentierung über Inventar-kategorisierung zur Sortierung von Sensordaten bis hin zur Anomalienerkennung (Trevino, 2016).

[5] Eine Alternative zur logistischen Regression stellen Support Vector Machines (SVM) dar (Ng, 2011).
[6] Sigmoidfunktion: $y = \frac{1}{1+e^{-x}}$; e = Eulersche Zahl

6. Deep Learning

Deep Learning orientiert sich am Modell des menschlichen Gehirns unter Verwendung künstlicher neuronalen Netze und ist für die jüngsten Durchbrüche im KI-Bereich verantwortlich. Beispiele sind Googles AlphaGo, autonomes Fahren und intelligente Sprachassistenten (Nvidia, 2018).
Biologische Neuronen (Gehirnzellen), von welchen der Mensch etwa 100 Milliarden besitzt, empfangen elektrische Eingangssignale und geben ein verarbeitetes elektrisches Signal an angrenzende Neuronen weiter. Dieses Prinzip machen sich künstliche neuronale Netze zu Nutzen. Da biologische Neuronen erst "feuern" beziehungsweise ein Signal abgeben, wenn die Eingabe genügend stark ist, kann keine einfache lineare Funktion verwendet werden. Es braucht eine Aktivierungsfunktion, entweder mit einem harten Schwellenwert (Stufenfunktion) oder einem weichen Schwellenwert (Sigmoidfunktion; mehr dazu im Kapitel 5). Die Eingangssignale werden durch ein Additionsverfahren gesammelt und bei ausreichender Stärke (Überschreitung der Schwelle) an angrenzende Neuronen weitergegeben (Rashid, 2017). In einem künstlichen neuronalen Netz zeigt sich das wie folgt:

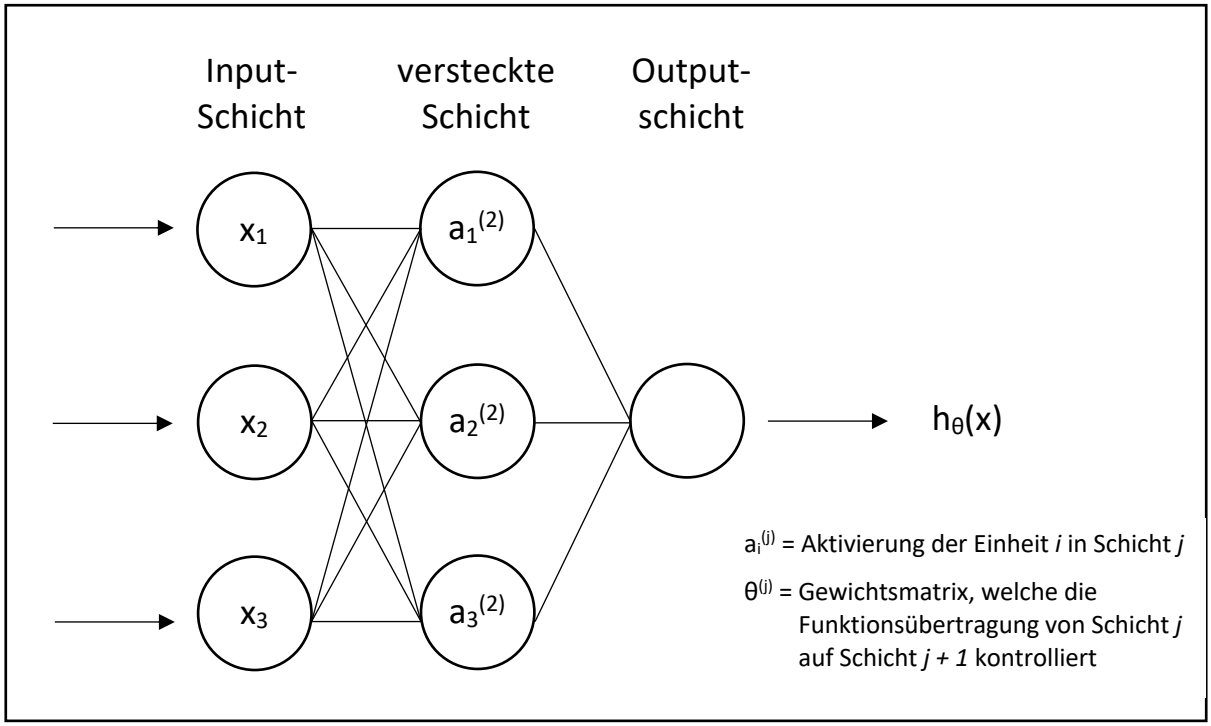

Abbildung 3. Neuronales Netz.

Auch hier gilt es, die Hypothese *h* zu finden (Ng, 2011):

$$a_1^{(2)} = g(\theta_{10}^{(1)}x_0 + \theta_{11}^{(1)}x_1 + \theta_{12}^{(1)}x_2 + \theta_{13}^{(1)}x_3)$$
$$a_2^{(2)} = g(\theta_{20}^{(1)}x_0 + \theta_{21}^{(1)}x_1 + \theta_{22}^{(1)}x_2 + \theta_{23}^{(1)}x_3)$$
$$a_3^{(2)} = g(\theta_{30}^{(1)}x_0 + \theta_{31}^{(1)}x_1 + \theta_{32}^{(1)}x_2 + \theta_{33}^{(1)}x_3)$$
$$h_\theta(x) = a_1^{(3)} = g(\theta_{10}^{(2)}a_0^{(2)} + \theta_{11}^{(2)}a_1^{(2)} + \theta_{12}^{(2)}a_2^{(2)} + \theta_{13}^{(2)}a_3^{(2)})$$

wobei x_0 eine Bias-Einheit (+1) darstellt und $g(z)$ die Sigmoidfunktion bezeichnet.

Zu Beginn werden die Gewichte nach dem Zufallsprinzip initialisiert. Es erfolgt die Berechnung von *h* (*Forward Propagation*). Anschliessend werden die Output-Werte der Hypothese *h* mit den Soll-Werten der wahren Funktion *f* abgeglichen. "Der Fehler eines

neuronalen Netzes ist eine Funktion der internen Verknüpfungsgewichte[7]. Ein neuronales Netz [zu] verbessern, bedeutet, diesen Fehler zu verringern – indem [...] Gewichte [θ_j] geändert werden" (Rashid, 2017, S. 88). Eine Verbesserung kann durch iterative Anpassung mittels dem Gradientenverfahren erreicht werden (mehr dazu im Kapitel 5). Die Berechnung des Gradientenvektors nennt sich *Backpropagation* und zeigt die relative Sensitivität jeder Gewichtung und Einheit in Bezug auf *h* auf. Nach Optimierung der Werte, dem Trainieren des Algorithmus, lässt sich das neuronale Netz respektive die Hypothese *h* für die beabsichtigten Vorhersagen einsetzen.

Es sind diverse Netzarchitekturen denkbar. Entscheidend für die Anzahl Einheiten (Neuronen) der Input-Schicht sind die relevanten Features (x_j). Bei der Output-Schicht sind die Anzahl Klassen (Klassifizierungskategorien) ausschlaggebend. Je mehr versteckte Schichten dazwischen geschaltet werden, desto "tiefer" wird das Netzwerk – daher der Begriff *Deep Learning*. Es benötigt dafür umso mehr Rechenleistung.

Obige Ausführungen beziehen sich auf überwachtes Lernen. Neuronale Netze können jedoch auch für unüberwachtes Lernen genutzt werden – etwa in Form eines *Autoencoders*. Autoencoder nehmen Input auf und geben eine Rekonstruktion dieses Inputs aus. Die Kostenfunktion ergibt sich aus dem Vergleich zwischen dem Originalinput und dem rekonstruierten Input. Autoencoder erlauben *Noise Reduction*, beispielsweise in der Bilderkennung, wodurch bedeutungsvolle Features extrahieren werden können (Chollet, 2016). Nach mehreren Iteration entsteht so ein "Code" (Muster) für spezifische Konzepte wie Katze, Gesicht, etc. Weiter werden unüberwachte Netze verwendet, um bereits trainierte Netze zu verstehen und die unterliegende Logik zu erklären (Zhang, Yang, Liu, Wu & Zhu, 2018). Desweitern gibt es Anwendungen mit semi-überwachtem Lernen, wobei ein kleiner Teil des Datensatzes gelabelt und der Rest ungelabelt ist. Damit lassen sich teilweise deutlich bessere Ergebnisse als mit gänzlich unüberwachtem Lernen erzielen.

Ebenfalls zu erwähnen ist das *Deep Reinforcement Learning* (bestärkendes Lernen). Dieser Ansatz hat sich aus der Tierpsychologie abgeleitet und beruht auf der Interaktion lernender Agenten mit ihrer Umwelt. Die Struktur gestaltet sich ähnlich wie beim überwachten Lernen: Ein Input wird in ein künstliches neuronales Netz eingespeist und gibt einen Output aus – allerdings ohne das korrekte Output-Label zu kennen (Karpathy, 2016). Um zu wissen, ob eine Aktion erfolgreich war, ist Feedback aus der Umwelt (*Reward* oder auch *Reinforcement*) notwendig. Der Reward fliesst als Teil des Inputs ins Netz ein und muss als solcher erkannt werden. Mit dieser Information wird über mehrere Iterationen hinweg eine Evaluationsfunktion gelernt, welche die Wahrscheinlichkeit aller Aktionen angibt, das gewünschte Ziel zu erreichen (Russel & Norvig, 2003). Das Netz, welches die Input-Output-Transformation vornimmt, nennt sich *Policy Network.* Die einfachste Art ein Policy Network zu trainierten, ist mittels Policy-Gradienten. Gestartet wird mit zufälligen Werten. Jeder Aktion mit positivem Reward wird ein positiver Gradient, jeder Aktion mit negativem Reward ein negativer Gradient zugewiesen mit der Idee, dass Aktionen mit negativem Reward aussortiert werden und die "gewinnenden" Strategien (*Policies)* übrigbleiben. Herausforderungen bestehen, wenn der Reward nicht nach jeder Aktion, sondern erst nach einer Episode mehrerer Aktionen vergeben wird (*Credit Assignment Problem; Sparse Reward Setting*) (Karpathy, 2016). Die Chance eine optimale Strategie durch zufälliges Ausprobieren zu finden, wird verschwindend klein, besonders bei komplexen Aufgabenstellungen wie dem Trainieren von Roboterarmen für das Greifen und Stapeln von Objekten. Dem kann durch *Reward Shaping* – dem manuellen Design einer Reward-Funktion, welche das Policy Network lenkt – entgegengewirkt werden. Der Nachteil liegt darin, dass die Reward-Funktion für jede Umgebung neu designt werden muss, was eine Skalierung deutlich einschränkt. Weiter kann Reward Shaping zu unerwarteten Ergebnissen führen, da der Fokus auf dem Reward-Sammeln anstatt dem Erreichen des gewünschten Ziels

[7] $J(\theta) = -\frac{1}{m}\sum_{i=1}^{m}\sum_{k=1}^{K}\left[y_k^{(i)}\log\left(\left(h_\theta\left(x^{(i)}\right)\right)_k\right) + \left(1 - y_k^{(i)}\right)\log\left(1 - \left(h_\theta\left(x^{(i)}\right)\right)_k\right)\right]$; allenfalls zzgl. eines Regularisierungsterms (Ng, 2011)

lieg. Auch können damit keine besseren Ergebnisse erzielt werden, als jene, zu denen die KI-entwickelnden Menschen in der Lage sind.
Nebst der beschriebenen Lernmethode des Gradientenverfahren wenden KI-Entwickler weitere Algorithmen an, die zum Teil weniger klar verständlich sind. Deep Learning wird daher oft als "Black Box" angesehen, wobei der Nachweis meist auf empirischen versus theoretischen Belegen beruht – inklusive der entsprechenden Fehleranfälligkeit.

7. Anwendungsmöglichkeiten

Das McKinsey Global Institute (2018) hat acht Problemtypen im Industrieumfeld identifiziert, die mittels künstlicher Intelligenz angegangen werden können. Untenstehende Tabelle fasst diese zusammen:

Tabelle 1. Anwendungsmöglichkeiten künstlicher Intelligenz (adaptiert von McKinsey Global Institute 2018).

Problemtyp	**Beispiele**	**mögliche KI-Techniken**
Klassifizierung Kategorisierung von Inputs zu einer bestimmten Kategorien	Bild mit spezifischem Objekttypus (Auto, Lastwagen, etc.); Produkt mit akzeptabler/inakzeptabler Qualität	neuronale Netze (*convolutional**), logistische Regression
kontinuierliche Schätzung Schätzung der nächsten numerischen Werte in einer Sequenz; Vorhersage	Forecast von Verkaufszahlen basierend auf historischen Verkaufsdaten; Schatzung von Immobilienwerten	neuronale Netze (*feedforward*), lineare Regression
Clustering Erstellung von Kategorien durch Gruppierung von Daten mit gleichen/ähnlichen Eigenschaften	Kundensegmentierung basierend auf demografischen Informationen, Präferenzen und Kaufverhalten	K-means Algorithmen, Affinity Propagation*
übrige Optimierungen Output-Optimierung einer spezifischen Zielfunktion	Berechnung einer Transportroute mit einer optimalen Kombination von Zeit- und Treibstoffaufwand	genetische Algorithmen*
Anomalienerkennung Feststellung von Normen-/ Durchschnittsabweichungen	Abgleich des Vibrationsverhaltens von Maschinen mit historischen Messungen zur (Früh-)Erkennung möglicher Fehler/Schäden	Support Vector Machines*, K-means Algorithmen, neuronale Netze
Ranking Informationsordnung /-sortierung nach Kriterien	Sortierung von Produktempfehlungen nach Relevanz	Support Vector Machines (*ranking*)*, neuronale Netze
Empfehlungen (Recommender Systems)	Kaufempfehlungen basierend auf Verhaltensmustern der beobachteten Person sowie ähnlicher Individuen	Collaborative Filtering*
Datengenerierung Generierung neuer Daten basierend auf historischen Daten	Komposition neuer Musik- & Kunstwerke in einem ähnlichen Stil wie bestehende Stücke	neuronale Netze (*generative adversarial**), versteckte Markov-Modelle*

*im vorliegenden Papier nicht beschrieben, aber vollständigkeitshalber aufgeführt; aufbauend auf erklärten Methoden/Ansätzen

Zusätzlich ist die Kundeninteraktion durch intelligente Sprachassistenten und *Chatbots* zu erwähnen. Weiter eröffnet das Internet der Dinge (*Internet of Things, IoT*) neue Möglichkeiten. Der intelligente Agent beschreibt eine autonome Einheit, welche die Umwelt durch Sensoren beobachtet und durch Aktoren darauf reagiert (Russel & Norvig, 2003). Sensor- und

aktorgenerierte Daten können fortlaufend in den Algorithmus eingespeist werden, wodurch ein dynamisches System entsteht, das sich kontinuierlich selbst verbessert (Reinforcement Learning). Das erlaubt eine individuelle Anpassung von Maschinen, Geräten und Systemen an die reale Umwelt und die Nutzerpräferenzen, wodurch zusätzlicher Mehrwert geschaffen werden kann.

8. Limitationen & Kontroversen

Anwendungspotenzial für künstliche Intelligenz ist vorhanden und die jüngsten Entwicklungen klingen vielversprechend. Trotzdem bestehen wesentliche Limitationen. KI-Programme funktionieren gut, wenn sie auf spezifische Tasks angewendet werden (schwache KI), sind jedoch weit davon entfernt, in unvorhersehbaren realen Situationen zu bestehen. Die Methoden des Machine Learnings haben sich in den letzten Jahren wenig verändert und weiterentwickelt. Die guten Ergebnisse sind vor allem auf die Zunahme verfügbarer Datenmengen und Rechenleistungen zurückzuführen. Dass teilweise unverständliche Algorithmen eingesetzt werden, die zu "Black Boxen" führen, hat Kontroversen hervorgerufen. Einige Experten haben zwar argumentiert, dass auch das menschliche Gehirn nicht komplett verstanden und trotzdem täglich benutzt wird. Dennoch gibt es Fälle, in denen es von grosser Bedeutung ist, Entscheidungen erklären zu können – beispielsweise wenn richterliche oder soziale Entscheide betroffen sind oder es um die Sicherstellung von Autonomie und persönlicher Würde geht. Algorithmen sollen Ergebnisse ethisch differenzieren. Es stellt sich allerdings die Frage, wie sich Ethikcodes programmieren (International Center for Scientific Debate, 2017) und definieren lassen. Dabei gilt es kulturelle Unterschiede zu beachten. Doch selbst wenn grundsätzlicher Konsens gefunden werden kann, sind Interpretationsdifferenzen bei der Implementierung nicht auszuschliessen. Hinzu kommt, dass Menschen nicht definierte Lücken mit gesundem Menschenverstand schliessen. Dieselben Lücken können von Computerprogrammen gänzlich anders ausgelegt werden.
Weiter beziehen sich Daten, welche der Vorhersage künftiger Werte dienen, naturgemäss auf vergangene Ereignisse. Dies bringt automatisch "mehr vom Gleichen" mit sich, was eine weitere Limitation darstellt. Je nach Situation wäre ein Musterbrechen vielleicht durchaus konstruktiv. Und was Börsenhändler schon lange wissen: Ansätze, die in der Vergangenheit erfolgreich waren, versprechen noch lange keine künftigen Erfolge. Zudem besteht die Gefahr einer Bias-Verstärkung, wenn sich ein solcher in historischen Daten verbirgt (Dastin, 2018). Um keine ungewollte Diskriminierung zu begünstigen, ist ein Verpflichtung zu guter Datenqualität und höchster Entscheidungstransparenz unabdingbar.
Auch die Einsatzbereiche künstlicher Intelligenz haben Diskussionen ausgelöst. So ist beispielsweise der KI-Einsatz für militärische Roboter und autonome Waffensysteme stark umstritten. Dasselbe gilt für die Verwendung von KI zur Misinformation und Wahlmanipulation (Gonzalez-Fierro, 2018). Hier wird im Gegensatz zu obigem Punkt nicht an algorithmische sondern menschliche Ethik appelliert. Und schliesslich bleibt noch die Kontroverse bezüglich starker und superintelligenter KI (mehr dazu im Kapitel 4), welche wohl bis auf Weiteres eine philosophische bleiben wird.

9. Konklusion & Ausblick

Das Erschaffen von Maschinen mit menschenähnlichen Fähigkeiten (Automata) war schon immer etwas, das die Menschen beschäftigt hat – vermutlich daher, weil es die Frage nach dem "Menschsein" und der eigenen Identität aufwirft. Aus diesem Grund gelingt es Schlagzeilen zum Thema künstliche Intelligenz so gut, Aufmerksamkeit auf sich zu ziehen, Erwartungen zu wecken und Ängste zu schüren. Implizit geht es dabei oft um starke oder superintelligente KI.

Wie sich gezeigt hat, liegen jedoch aktuell nicht ausreichend Forschungsergebnisse zur Funktionsweise des Gehirns vor, um eine fundierte Beurteilung vorzunehmen.
Die Methoden und Ansätze der schwachen KI im Machine und Deep Learning wurden bereits vor mehreren Jahrzehnten entwickelt. Ergebnisse können sie heute dank grösseren Datenmengen und verbesserter Rechenleistung erzielen. Bei beiden Faktoren dürfte mit einer weiteren Zunahme gerechnet werden, was zusätzliche Durchbrüche verspricht. Es kann auf Produkte und Services mit Mehrnutzen sowie auf effizientere und effektivere Prozesse gehofft werden. Allerdings ist auch hervorgekommen, dass ernsthafte Limitationen bestehen, die auch in Zukunft menschliches Mitdenken, starke Bereitschaft und ethisches Commitment erfordern wird.

Weiterführende Literatur (Empfehlungen)

Russel, S. J., & Norvig, P. (2003). *Artificial intelligence: A modern approach* [Künstliche Intelligenz: Ein moderner Ansatz]. Upper Saddle River, NJ: Prentice Hall.

Kurzweil, R. (2012). *How to create a mind. The secret of human thought revealed* [Wie Geist geschaffen wird. Das Geheimnis menschlichen Denkens erschlossen]. London: Duckworth Publishers.

Tegmark, M. (2017). *Life 3.0* [Leben 3.0]. London: Penguin Books.

Quellen

AlDahdouh, A. A., Osorio, J., & Caires, S. (2015). Understanding knowledge network, learning and connectivism [Wissensnetzwerk, Lernen und Konnektivismus verstehen

Boden, M. A. (2016). *Artificial intelligence. A very short introduction* [Künstliche Intelligenz. Eine sehr kurze Einführung]. Oxford: Oxford University Press.

Buchanan, B. G. (2005). A (very) brief history of artificial intelligence [Eine (sehr) kurze Geschichte künstlicher Intelligenz]. *AI Magazine 26*(4), 53-60.

Bulezyuk, A. (n.d.). *Machine learning model: Python Sklearn & Kera* [Maschinelles Lenrnen Modell: Python & Kera]. (Online-Kurs) Abgerufen am 19. Oktober 2018 von https://www.liveedu.tv/andreybu/REaxr-machine-learning-model-python-sklearn-kera/oPGdP-machine-learning-model-python-sklearn-kera/

Chollet, F. (2016). *Building autoencoders in Kera* [Autoencoder in Kera erstellen]. Abgerufen am 19. Oktober 2018 von https://blog.keras.io/building-autoencoders-in-keras.html

Damasio, A. (2010). *Self comes to mind* [Das Selbst kommt zum Geist]. New York, NY: Random House.

Dastin, J. (2018). *Amazon scraps secret AI recruiting tool that showed bias against women* [Amazon ausrangiert ein geheimes Rekrutierungstool mit einem Bias gegen Frauen]. Abgerufen am 19. Oktober 2018 von https://www.reuters.com/article/us-amazon-com-jobs-automation-insight/amazon-scraps-secret-ai-recruiting-tool-that-showed-bias-against-women-idUSKCN1MK08G

Gonzalez-Fierro, M. (2018). *10 ethical issues of artificial intelligence and robotics* [10 ethische Herausforderungen bezüglich künstlicher Intelligenz und Robotik]. Abgerufen am 19. Oktober 2018 on https://miguelgfierro.com/blog/2018/10-ethical-issues-of-artificial-intelligence-and-robotics/

International Center for Scientific Debate. (2017). *Artificial intelligence: Dreams, risks and reality* [Künstliche Intelligenz: Träume, Risiken und Realität]. Abgerufen am 19. Oktober 2018 von https://www.bdebate.org/en/synopsis/page/2-limitations-and-ethics-artificial-intelligence

Karpathy, A. (2016). *Deep reinforcement learning: Pong from pixels* [Tiefes, bestärkendes Lernen: Tischtennis von Pixeln]. Abgerufen am 21. Oktober 2018 von http://karpathy.github.io/2016/05/31/rl/

Koza, J. R., Bennett, F. H., Andre, D., & Keane, M. A. (1999). *Genetic programming III: Darwinian invention and problem solving* [Genetische Programmierung III: Darwinistische Erfindung und Problemlösung]. Burlington, MA: Morgan Kaufmann.

Kurzweil, R. (2012). *How to create a mind. The secret of human thought revealed* [Wie Geist geschaffen wird. Das Geheimnis menschlichen Denkens erschlossen]. London: Duckworth Publishers.

Ng, A. (2011). *Machine learning* [Maschinelles Lernen] (Online-Kurs). Abgerufen am 15. Oktober 2018 von https://www.coursera.org/learn/machine-learning

Nilsson, N. J. (2010). *The quest for artificial intelligence. A history of ideas and achievements* [Das Streben nach künstlicher Intelligenz. Eine Geschichte der Ideen und Erfolge]. Abgerufen am 14. Oktober 2018 von https://ai.stanford.edu/~nilsson/QAI/qai.pdf

Nvidia. (2018). *Deep Learning.* Abgerufen am 19. Oktober 2018 von https://developer.nvidia.com/deep-learning

McCorduck, P. (2004). *Machines who think* [Maschinen, die denken]. Natick, MA: A. K. Peters.

McKinsey Global Institute (2018). *Notes from the AI frontier. Insights from hundreds of use cases* [Notizen von der KI-Front. Einsichten aus hunderten Anwendungsfällen] (Diskussionspapier). New York, NY: McKinsey Global Institute.

Merriam-Webster Dictionary. (n.d.). *Intelligence* [Intelligenz]. Abgerufen am 15. Oktober 2018 von https://www.merriam-webster.com/dictionary/intelligence

Pearl, J. (1988). *Probabilistic reasoning in intelligent systems: Networks of plausible inference* [Wahrscheinlichkeitsbezogenes Begründen in intelligenten System: Netzwerke plausibler Inferenz]. San Mateo, CA: Morgan Kaufmann.

Rashid, T. (2017). *Neuronale Netze selbst programmieren.* Heidelberg: O'Reilly.

Roberts, J. (2016). *Thinking machines: The search for artificial intelligence. Does history explain why today's smart machines can seem so dumb?* [Das Suche nach der künstlichen Intelligenz. Kann die Vergangenheit erklären, warum die heutigen, klugen Maschinen so dumm erscheinen?]. Abgerufen am 14. Oktober 2018 von https://www.sciencehistory.org/distillations/magazine/thinking-machines-the-search-for-artificial-intelligence

Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain [Das Perceptron: Ein Wahrscheinlichkeitsmodell für Informationsspeicherung und -organisation im Gehirn]. *Psychological Review, 65*(6), 386-408.

Russel, S. J., & Norvig, P. (2003). *Artificial intelligence: A modern approach* [Künstliche Intelligenz: Ein moderner Ansatz]. Upper Saddle River, NJ: Prentice Hall.

Schank, R. C., & Abelson, R. P. (1977). *Scripts, plans, goals and understanding: An inquiry into human knowledge structures* [Skripte, Pläne, Ziele und Verstehen: Eine Untersuchung der menschlichen Wissensstrukturen]. Hillsdale, NJ: Erlbaum.

Searle, J. R. (1980). Minds, brains, and programs [Geist, Gehirne und Programme]. *BBS, 3*, 417–457.

Tegmark, M. (2017). *Life 3.0* [Leben 3.0]. London: Penguin Books.

Trevino, A. (2016). *Introduction to K-means clustering* [Einführung in das K-means Clustering]. Abgerufen am 19. Oktober 2018 von https://www.datascience.com/blog/k-means-clustering

Zhang, Q., Yang, Y., Liu, Y., Wu, Y. N., & Zhu S. (2018). *Unsupervised learning of neural networks to explain neural networks* [Unüberwachtes Lernen neuronaler Netzwerke zur Erklärung neuronaler Netzwerke] (unveröffentlichter Artikel). Abgerufen am 19. Oktober 2018 von https://arxiv.org/pdf/1805.07468.pdf