



# Automatische Programmierung von Produktionsmaschinen

Florian Eiling<sup>1(✉)</sup> und Marco Huber<sup>1,2</sup>

<sup>1</sup> Institut für Industrielle Fertigung und Fabrikbetrieb IFF, Universität Stuttgart, Stuttgart, Deutschland

florian.eiling@iff.uni-stuttgart.de

<sup>2</sup> Zentrum für Cyber Cognitive Intelligence (CCI), Fraunhofer IPA, Stuttgart, Deutschland

**Zusammenfassung.** Heutige Methoden der Programmierung von Produktionsmaschinen erfordern großen manuellen Aufwand. Dies hat zur Konsequenz, dass der Einsatz heutiger Automatisierungslösungen nur bei großen Stückzahlen wirtschaftlich ist. Im Zuge der Massenpersonalisierung kommt es gleichzeitig zu immer höheren Anforderungen an die Flexibilität der Produktion. Damit kann der Produktionsstandort Deutschland nur mittels einer gesteigerten digitalen Souveränität über die eigenen Produktionsmaschinen durch eine aufwandsreduzierte, flexible Programmiermöglichkeit von Produktionsmaschinen gehalten werden.

Zur Reduzierung des Programmieraufwands sind Methoden des Maschinellen Lernens geeignet, insbesondere das Teilgebiet des Reinforcement Learning (RL). Beides verspricht eine deutlich gesteigerte Produktivität. Im Folgenden werden die Möglichkeiten und die Hindernisse auf dem Weg zur RL-gestützten, flexiblen, autonom handelnden Produktionsmaschine analysiert.

Besonders im Fokus stehen dabei Aspekte der Zuverlässigkeit von Systemen aus dem Feld der Künstlichen Intelligenz (KI). Ein zentraler Aspekt der Zuverlässigkeit ist die Erklärbarkeit der KI-Systeme. Diese Erklärbarkeit ist wiederum eine tragende Säule der digitalen Souveränität auf der Ebene der das System nutzenden Menschen.

**Schlüsselwörter:** Künstliche Intelligenz · Automatisierung · Flexible Programmierung

## 1 Status quo

Der Wunsch nach intelligenten, autonom handelnden Produktionsmaschinen hat sich bisher noch nicht erfüllt. Heutige Produktionsmaschinen führen primär repetitive Tätigkeiten durch. Bei kleinen Änderungen im Prozess müssen die Produktionsmaschinen aufwändig neu angepasst werden. Ihr Einsatz rechnet sich deswegen nur bei der Produktion mit großen Stückzahlen (Losgröße), während bei einer kleinen Losgröße der Aufwand für die Programmierung der Produktionsmaschinen bereits häufig den Aufwand der manuellen Fertigung selbst übersteigt. Der aufwändige Programmierprozess für Produktionsmaschinen führt also zu unflexiblen und nicht

wandlungsfähigen Systemen, die sich nicht für die Massenpersonalisierung eignen (Westkämper 2014; Bauernhansl 2017). Massenpersonalisierung bezeichnet „die Produktion von Gütern und Leistungen, welche die unterschiedlichen Bedürfnisse jedes einzelnen Nachfragers dieser Produkte treffen, mit der Effizienz einer vergleichbaren Massen- beziehungsweise Serienproduktion“ (Piller 2008).

Im Folgenden werden Robotersysteme als Beispiel für Produktionsmaschinen betrachtet. Die dargestellten Methoden sollten sich analog auf andere Produktionsmaschinen übertragen lassen. Moderne Ansätze wie zum Beispiel die intuitive oder die skill-basierte Roboterprogrammierung reichen nicht aus, um dieses fundamentale Problem des Programmieraufwands zu beheben. Bei diesen Methoden können fertige Programmierbausteine in einfacher Art und Weise zu einem Programm zusammengefügt werden. Hierfür gibt es zahlreiche kommerzielle Lösungen (ArtiMinds Robotics 2020; drag&bot 2020). Durch die Verwendung fertiger Programmierbausteine sinken das nötige Expertenwissen und der benötigte Aufwand für die erfolgreiche Neuprogrammierung. Allerdings ist für komplexe Prozesse weiterhin Expertenwissen in der Robotik notwendig. Auch ist der nötige Aufwand zur Neukonfiguration weiterhin signifikant.

Die Roboterprogrammierung durch Demonstration kann diese Probleme ebenfalls nicht lösen. Bei der genannten Programmiermethode kann ein Mensch einen Roboter durch eine Aufgabe führen. Durch mehrmaliges Wiederholen und entsprechende Optimierung auf der Softwareseite lernt der Algorithmus, den Prozess selbstständig durchzuführen, wodurch dem Automatisierungssystem auch komplexe Aufgaben relativ schnell beigebracht werden können. Auch wenn dadurch der Programmieraufwand und das nötige Expertenwissen reduziert werden, ist dieses System zu unflexibel für die Produktion in kleinen Losgrößen. Des Weiteren werden eine menschliche „Lehrperson“ und die reale Hardware benötigt, wodurch es zeitweise zum Stillstand der Produktion kommt (Billard et al. 2008).

Neben dem Programmieraufwand und der benötigten Expertise gibt es noch weitere Gründe, warum die heutigen Automatisierungssysteme nicht geeignet sind für die Produktion in kleinen Losgrößen. Zum einen sind Testläufe nötig, während derer Ausschuss erzeugt wird. In kleinen Losgrößen wird dadurch der Materialbedarf vervielfacht. Darüber hinaus ist die Kooperation der Produktionsmaschinen mit den Menschen nicht optimal. Aufgrund der Inflexibilität übernehmen häufig die Beschäftigten die komplexen Aufgaben, während der Roboter lediglich wartet. Das heißt, die anfallende Arbeit ist nicht optimal verteilt und es sind nicht alle Kapazitäten ausgelastet.

Ein Beispiel für die Produktion in kleinen Losgrößen ist die Montage von elektrischen Schaltschränken. Diese werden häufig in Losgrößen von eins bis zehn gefertigt. Dabei sind die auftretenden einzelnen Arbeitsschritte alle mit dem heutigen Stand der Technik prozesssicher durchführbar. Aufgrund einer Vielzahl an Kombinationsmöglichkeiten gibt es aber kein Automatisierungssystem, das in der Lage ist, die Schaltschrankmontage wirtschaftlich durchzuführen. Hinzu kommt, dass der Aufwand der manuellen Programmierung bereits den Aufwand der manuellen Produktion übersteigt. Aus diesem Grund wird die Montage von Schaltschränken häufig mindestens in Teilprozessen manuell durchgeführt (Tempel et al. 2017).

Ein möglicher Ansatz für die Entwicklung einer Automatisierungslösung für kleine Losgrößen, die die oben genannten Probleme behebt, ist die Verwendung von Methoden des maschinellen Lernens (ML) – insbesondere des sogenannten Reinforcement Learning (RL). Eine solche ML-Lösung muss also die folgenden Anforderungen erfüllen.

(1) Das System muss das nötige Expertenwissen, das zur Programmierung und zur Neukonfiguration der Produktionsmaschinen notwendig ist, weiter reduzieren und eine hohe Flexibilität des Gesamtsystems ermöglichen. Dies steigert sowohl die Souveränität der Unternehmen als auch die Souveränität der beteiligten Beschäftigten über die eigenen Produktionsmaschinen. (2) Weiter muss die Programmierung möglichst defekt- und unterbrechungsfrei stattfinden, damit die Automatisierung der Produktion auch in kleinen Losgrößen wirtschaftlich ist. (3) Durch die Automatisierung sollten möglichst alle nicht wertschöpfenden Tätigkeiten an das Automatisierungssystem verlagert werden können. (4) Wenn weitere Schritte notwendig sind, soll die Zusammenarbeit mit den beteiligten Arbeitern sicher, zuverlässig, und ohne Stillstand sein, wobei der Mensch immer das Sagen haben muss.

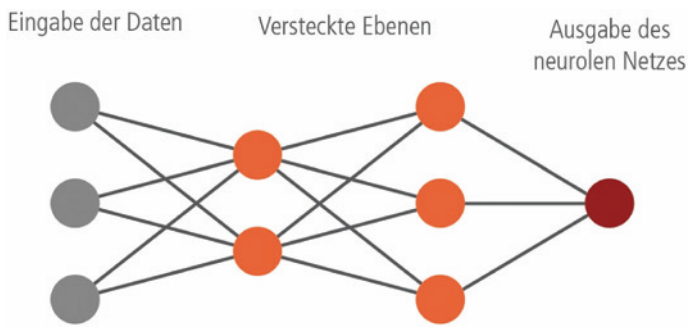
## 2 Aktuelle Entwicklungen

Aktuelle Trends, die die Entwicklung von flexiblen Automatisierungsbedingungen erfordern oder ermöglichen, sind die Massenpersonalisierung, die weiterlaufende Digitalisierung und die Künstliche Intelligenz (KI) mit ihrem Teilgebiet des ML. Konsumierende fordern immer weiter personalisierte Produkte – in diesem Zuge spricht man von der Massenpersonalisierung. Darüber hinaus kommt es zu einem Fachkräftemangel durch den demographischen Wandel – auf Seiten der Produktionsfachkräfte und auf Seiten der Robotikexpertinnen und -experten. Zusammen mit hohen Lohnkosten am Standort Deutschland verlangen diese Trends die Entwicklung einer flexiblen Automatisierungslösung. Bisherige Automatisierungslösungen sind nicht hinreichend flexibel und erfordern zu viel Aufwand beziehungsweise Expertenwissen zur Programmierung und Neukonfiguration.

Einen möglichen Ansatz zur Entwicklung einer solchen Automatisierungslösung bietet das ML. ML-Techniken haben sich in den letzten Jahren in vielen Bereichen verbreitet – sowohl auf Seiten der Forschung als auch auf Seiten der Industrie und der Gesellschaft. Begründet wird dieses Interesse durch vorher nicht für möglich gehaltene Fortschritte zum Beispiel im Bereich der Bildverarbeitung und Bilderkennung, der Entwicklung von ML-Lösungen für Brettspiele und der maschinellen Übersetzung. Hierfür werden moderne ML-Techniken wie tiefe neuronale Netze mit großem Erfolg eingesetzt (Silver et al. 2017; Johnson et al. 2016; Krizhevsky et al. 2012).

Tiefe neuronale Netze setzen sich aus mehreren Schichten von künstlichen Neuronen zusammen. „Tief“ bezeichnet das Hintereinanderschalten von sehr vielen solcher Schichten. Diese orientieren sich lose an biologischen Neuronen und sind mit allen Neuronen der darauffolgenden Schicht verbunden (siehe Abb. 1). Die Verbindungen zwischen den Neuronen können unterschiedlich „stark“ sein und kodieren

das gelernte Wissen des neuronalen Netzes. Die Eingabedaten werden von vorne nach hinten durch das Netz propagiert – also über die Verbindungen zwischen den Neuronen gewichtet an die jeweils darauffolgenden Neuronen weitergegeben, bis die letzte Schicht (die Ausgabeschicht) des Netzes erreicht ist. An der Ausgabeschicht des Netzes wird ein eventueller Fehler rückwärts durch das Netz propagiert und die Verbindungen der Neuronen werden so angepasst, dass das Netz in Zukunft weniger Fehler macht. Diesen Prozess bezeichnet man als das Training des tiefen neuronalen Netzes. Nach Abschluss des Trainings kann das Netz in der Anwendung benutzt werden. Der Erfolg dieser tiefen neuronalen Netze beruht darauf, dass sie beliebige Funktionen approximieren und deswegen vielseitig eingesetzt werden können (Goodfellow et al. 2016; LeCun et al. 2015).



**Abb. 1.** Schematische Darstellung eines neuronalen Netzes. Die Eingabedaten werden in der Eingabeschicht (grau) in das Netz reingegeben und anschließend an die nachfolgenden Schichten (orange) weitergegeben, bis das Ende des Netzes, die Ausgabeschicht (rot), erreicht ist.

Der erfolgreiche Einsatz von modernen ML-Methoden benötigt hochqualitative Daten, genaue digitale Modelle (digitale Zwillinge) und leistungsstarke Rechner. Diese Anforderungen haben sich in den letzten Jahren aufgrund exponentiell gesteigener Rechenleistung und exponentiell angewachsener Datenmengen zunehmend erfüllt, unter anderem durch die Digitalisierung (Alom et al. 2018; LeCun et al. 2015).

## 2.1 ML in der Produktion

Ein wichtiges Ziel des Einsatzes von ML-Methoden in der Produktion war immer die Entwicklung von intelligenten, autonomen Systemen. Darüber hinaus versprechen ML-Methoden großes Potenzial in der gesamten Wertschöpfungskette – von der Produktentstehung, über die Produktionsplanung, bis hin zur Qualitätskontrolle, der automatischen Wartung und dem Vertrieb. Unternehmen erhoffen sich eine gesteigerte Produktqualität, bei größerer Flexibilität, niedrigeren Kosten und eine Reduktion des Energieverbrauchs beziehungsweise der Umweltbelastung. Zu diesen Zwecken werden ML-Techniken in der Industrie bereits erfolgreich eingesetzt (accenture 2017; Kober et al. 2013; Spinnarke 2017).

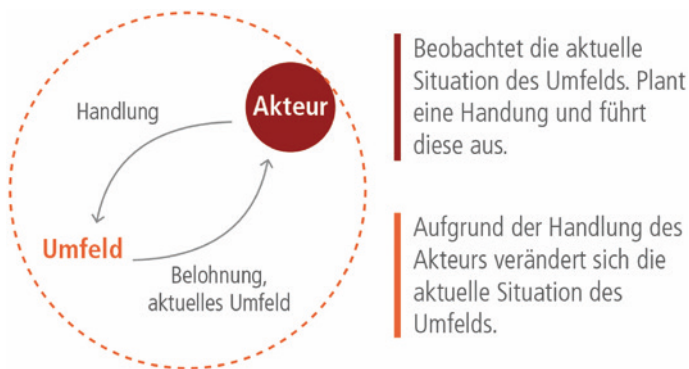
Allerdings gibt es einige typische Hürden, die einem erfolgreichen Produktiveinsatz von ML-Techniken häufig im Wege stehen. Neben mangelnder Datenmenge und -qualität, ist hier insbesondere auch die fehlende Zuverlässigkeit von ML-Methoden zu nennen. Darauf wird in den folgenden Abschnitten weiter eingegangen.

## 2.2 ML zur Steuerung von Produktionsmaschinen

Zur Programmierung und Steuerung von Produktionsmaschinen wird viel an ML-Methoden geforscht. Ein besonderes Interesse kommt hierbei dem sogenannten Reinforcement Learning (RL) zu.

Das Reinforcement Learning („Verstärkendes Lernen“) ist ein allgemeines Lernparadigma. Das Paradigma besteht aus einem Akteur, seinem Umfeld, seinen Handlungsmöglichkeiten und einer Belohnung für seine Handlungen (siehe Abb. 2). Der Akteur befindet sich in einem gewissen Umfeld. Er hat die Möglichkeit, Handlungen auszuführen – also etwas zu tun. Die Handlungen, für die der Akteur sich entscheidet, haben Auswirkungen auf das Umfeld und abhängig von der Handlung verändert sich das Umfeld. Der Akteur erhält nun von seinem Umfeld beziehungsweise durch die Auswirkungen seiner Handlungen eine sogenannte Belohnung, die einen Anreiz darstellt, etwas in Zukunft wieder zu tun oder es zu unterlassen. Anschließend hat der Akteur die Möglichkeit, die Auswirkungen seiner Handlung auf die Umwelt zu beobachten, unter Einbeziehung der Belohnung eine neue Handlung auszuwählen und diese auszuführen. Diesen Vorgang von einer Umfeldveränderung zur nächsten bezeichnet man als einen (Zeit-)Schritt des Lernparadigmas.

Daraufhin wiederholt sich der ganze Schritt immer wieder, bis das Lernen gestoppt wird. Das Ziel des Akteurs ist es, durch seine Handlungen die Belohnungen, die er von seinem Umfeld erhält, zu maximieren. Er möchte also maximal viele Belohnungen erhalten. Die Belohnungsfunktion wird vom Entwickler festgelegt und enthält Informationen über die zu lernende Aufgabe. Der Lernalgorithmus hat das Ziel, die Ausführung der durch die Belohnungsfunktion kodierten Aufgabe optimal zu lernen (Sutton und Barto 2018).



**Abb. 2.** Das Reinforcement-Learning-Paradigma. Man sieht das Zusammenspiel von Akteur und Umfeld: Der Akteur führt eine Handlung aus, erhält dafür eine Belohnung und beobachtet anschließend ein verändertes Umfeld.

Dieses allgemeine Paradigma erlaubt die Modellierung einer Vielzahl von Situationen (Sutton und Barto 2018). Ein Beispiel ist die Produktionsplanung einer Anlage zur Halbleiterfertigung durch einen ML-Algorithmus.

- Der Akteur ist in diesem Beispiel der ML-Algorithmus, der entscheidet, wann die Produktionsschritte auf welcher Maschine ausgeführt werden.
- Das Umfeld ist in diesem Fall ein abstraktes Modell der Fabrik. Das Umfeld beinhaltet die Maschinen in der Fabrik, die Fähigkeiten und aktuellen Zustände der Maschinen (kaputt, wartend usw.) sowie die aktuelle Verteilung der Jobs auf den Maschinen.
- Der Akteur hat die Möglichkeit, bei Eintreffen eines neuen Jobs oder bei Abschluss eines Jobs die aktuelle Planung zu verändern. Das heißt, der Akteur entscheidet darüber, welche Jobs aktuell wo produziert werden, aber auch, welche Jobs in der Warteschleife hängen.
- Die Belohnungsfunktion wird vom Entwickler des ML-Algorithmus ausgewählt und erhält Informationen über die zu erledigende Aufgabe – insbesondere was der Algorithmus optimieren soll. In diesem Fall beinhaltet die Belohnungsfunktion Leistungskennzahlen der Fabrik, wie zum Beispiel die Gerätebetriebszeit (Waschneck et al. 2018).

In den vergangenen Jahren wurden moderne ML-Techniken, wie etwa tiefe neuronale Netze, sehr erfolgreich als der lernende ML-Algorithmus in diesem Lernparadigma eingesetzt. So kann das neuronale Netz zum Beispiel direkt lernen, welche Handlung in welcher Situation ausgeführt werden soll (ein sogenannter modellfreier Algorithmus). Alternativ kann das Netz aber auch zum Einlernen eines Umfeldmodells und anschließend zur Planung der Handlungen verwendet werden (ein sogenannter modellbasierter Ansatz). Insbesondere wird viel daran geforscht, RL zur Steuerung von Produktionsmaschinen und anderen automatisierten Systemen zu verwenden. In speziellen Fragestellungen wurde RL dafür auch bereits erfolgreich eingesetzt. Dies beinhaltet zum Beispiel den Griff in die Kiste und das anschließende Werfen eines Objektes in eine andere Kiste, das Sortieren von Objekten im Produktiveinsatz, die Optimierung von Produktionsplanung, das Einfügen eines festen Stifts in ein deformierbares Loch und das Zusammenfügen von Steckverbindungen. Auch wenn dies hochkomplexe Probleme sind, sind es Spezialaufgaben in einem jeweils sehr genau kontrollierten Setting und keine kompletten Produktionsprozesse (Waschneck et al. 2018; Xanthopoulos et al. 2018; Zeng et al. 2019; covariant.ai 2020; Luo et al. 2018; Schoettler et al. 2019).

### 2.3 Inhärente technische Hürden

Neben der bisher noch fehlenden angewandten Forschung in diesem Bereich weisen Reinforcement-Learning-Algorithmen in der realen Welt typischerweise einige Nachteile auf, die es noch zu überwinden gilt. Die erste Hürde ist die geringe Dateneffizienz beziehungsweise die große Menge an benötigten Trainingsdaten. Insbesondere die häufig verwendeten modellfreien RL-Algorithmen benötigen

typischerweise Millionen von Trainingspunkten, bis sie ihre endgültige Leistungsfähigkeit erreichen. Diese Datenmengen lassen sich mit dem Training auf realen Systemen – also ohne die Verwendung von Simulationen – nur schwer erzeugen. Limitierende Faktoren sind hierbei insbesondere die Voraussetzung einer großen Menge an parallelen Aufbauten, der hohe Zeitaufwand und der Verschleiß der Hardware (Irpan 2020; Hessel et al. 2017; Kober et al. 2013; Mahmood et al. 2018).

Aus diesem Grund sind hochqualitative Simulationen zum Training der Algorithmen erforderlich. Diese haben den Vorteil, dass die ML-Algorithmen kostengünstig und effizient trainiert werden können, aber auch die hohe Flexibilität, die die Verwendung einer Simulationsumgebung bietet. So können die ML-Algorithmen unabhängig von der realen Hardware trainiert werden. Das heißt zum einen ohne Produktionsstillstand zum Training der ML-Algorithmen und zum anderen, dass die simulierte Trainingsumgebung flexibel verändert und an neue Produkte angepasst werden kann, ohne dass Änderungen an der realen Hardware nötig sind. Allerdings ist es nicht möglich, alle auftretenden relevanten physikalischen Prozesse während eines komplexen Vorgangs korrekt zu modellieren – dies beinhaltet zum Beispiel den Einfluss der Temperatur, den Verschleiß der Produktionsmaschinenteile oder Verformungen der Werkstücke. Aus diesem Grund bildet die Simulation die Realität nie genau ab, weswegen es bei dem Transfer von in der Simulation gelernten ML-Steuerungsalgorithmen auf reale Hardware zu einem Leistungsabfall kommt. Moderne Techniken wie Domain-Randomisierung können dem aber effizient entgegenwirken (Chebotar et al. 2019; Peng et al. 2018; Tobin et al. 2017).

Ein weiteres Problem ist, dass die ML-Algorithmen häufig zufällig initialisiert werden, das heißt, dass sie anfangs zufällig handeln und von Grund auf lernen, die Aufgabe auszuführen. Durch die zufällige Initialisierung ist das Lernen allerdings nicht zielgerichtet, sondern basiert auf der zufälligen Annäherung an die korrekte Ausführung. Dies führt gerade zu Beginn des Trainings auf realer Hardware zu einem hohen Verschleiß. Dem kann aber mit der Verwendung von Simulationsumgebungen entgegengewirkt werden. Darüber hinaus steigert diese Art der Initialisierung die benötigten Datenmengen, da der Algorithmus alles von Grund auf lernen muss. Leider ist es insbesondere bei den häufig verwendeten modellfreien Algorithmen nicht möglich, effektiv Vorwissen einzubringen oder bereits gelerntes Wissen auf ähnliche Probleme zu übertragen, was diesem Problem entgegenwirken könnte (Polydoros und Nalpantidis 2017; Mahmood et al. 2018; Schulman et al. 2015).

Schlussendlich sind die nach Abschluss des Trainings erhaltenen ML-Steuerungsalgorithmen häufig instabil gegenüber kleinen Änderungen des Umfelds, weisen eine nicht ausreichende Erfolgsrate auf, und der Transfer auf ähnliche, aber noch nicht gesehene Aufgaben ist nicht ausreichend. Dies kann dazu führen, dass eine kleine Veränderung des Umfelds (wie zum Beispiel eine veränderte Startposition des Roboterarms) dazu führt, dass der Algorithmus die Aufgabe nicht korrekt ausführen kann. Häufig müssen bestimmte Algorithmen zur Erledigung einer weiteren ähnlichen Aufgabe komplett neu trainiert werden, da bereits gelerntes Wissen nicht effektiv genutzt werden kann.

## 2.4 Lösungen für die inhärenten technischen Hürden

Vielversprechende Lösungsansätze für die geringe Dateneffizienz, die nicht effektive Verwendung von vorhandenem Vorwissen und die geringe Verallgemeinerung der gelernten Algorithmen liegen in der Verwendung von modellbasierten RL-Algorithmen. Hierbei lernt der Algorithmus ein Modell seines Umfelds und kann damit seine Handlungen planen. Man kann sich zum Beispiel vorstellen, was passieren würde, wenn man sein Smartphone fallen lässt und beschließen, dies deswegen lieber zu vermeiden, ohne es wirklich ausprobieren zu müssen. Die Verwendung eines Modells erlaubt es dem Algorithmus also, die Auswirkungen von Handlungen abzuschätzen, ohne diese ausprobieren zu müssen (insofern das Modell gut genug ist), wodurch der Algorithmus zielgerichtet mit deutlich weniger Daten die korrekte Ausführung lernt. Darüber hinaus erlaubt die Verwendung eines Modells das Einbringen von Vorwissen („Die Firma weiß schon, was gemacht werden soll und wie das geht, nur der Roboter noch nicht.“) und den Transfer von bereits gelerntem Wissen auf ähnliche Aufgaben. Moderne modellbasierte ML-Algorithmen weisen auch keinen Leistungs Nachteil gegenüber modellfreien ML-Algorithmen mehr auf (Polydoros und Nalpantidis 2017; Ha und Schmidhuber 2018).

Weitere Ansätze, den Transfer auf ähnliche Aufgaben zu verbessern, sind das sogenannte Transfer- und Metalernen. Beim Transferlernen wird dem ML-Algorithmus eine Vielzahl von ähnlichen, aber unterschiedlichen Aufgaben gestellt. Der Algorithmus hat das Ziel, alle Aufgaben gut zu erfüllen. Beim Metalernen hat der Algorithmus das Ziel, Lernen zu lernen. Durch beide Techniken werden die Algorithmen robuster gegenüber kleineren Änderungen und lernen, sich sehr schnell an neue Aufgaben anzupassen, sodass sie flexibler eingesetzt werden können (Tan et al. 2018; Hausman et al. 2018; Duan et al. 2016; Finn et al. 2017).

An letzter Stelle seien hier noch hierarchische Lernalgorithmen genannt. Diese Klasse von ML-Algorithmen setzt sich aus mehreren Ebenen zusammen. Zum Beispiel kann es eine Ebene geben, die bestimmte Skills – wie zum Beispiel das Verdrehen von Bauteilen, das Einfügen in Schnappverbindungen oder das Greifen von Bauteilen lernt. Dann gibt es weitere Ebenen, die diese Skills kombinieren und so selbst komplexe Aufgaben durchführen können. Dieser Ansatz hat den Vorteil, dass skillspezifische Informationen gekapselt und unabhängig voneinander gelernt werden können. Das steigert die Dateneffizienz, da die Skills nur einmal gelernt werden müssen und dann für neue Aufgaben eingesetzt werden können. Dafür muss lediglich die „Kombinationsebene“ lernen, die neue Aufgabe durch Kombination der bekannten Skills auszuführen (Nachum et al. 2018; Vezhnevets et al. 2017; Florensa et al. 2017).

In Summe wirken die in den letzten Jahren stark weiterentwickelten Ansätze den skizzierten Limitierungen des einfachen RL entgegen.

## 2.5 Zuverlässigkeit, digitale Souveränität und regulatorische Anforderungen

Das Interesse an zuverlässigen ML-Methoden ist in den letzten Jahren stark gestiegen. Die Zuverlässigkeit von ML-Algorithmen meint in diesem Kontext Anforderungen an die Erklärbarkeit, die Verifizierbarkeit und die Risikoabschätzung, die bei klassischen



modernen Methoden wie tiefen neuronalen Netzen wenig bis gar nicht erfüllt sind. Insbesondere aber in sicherheitskritischen Anwendungsgebieten wie dem autonomen Fahren oder der Mensch-Roboter-Kooperation sowie in regulatorisch komplexen Anwendungsfällen – zum Beispiel aufgrund von Datenschutzbestimmungen – sind diese unabdingbar: Man stelle sich ein autonomes Auto vor, von dem niemand weiß, wie es eigentlich funktioniert, bekannt ist nur, dass es bei allen Tests erfolgreich abgeschlossen hat. Wirklich sicher wird man sich in diesem Auto nicht fühlen. Das beschreibt die Problematik der fehlenden Erklärbarkeit ganz gut (Wachter et al. 2017; Huang et al. 2017; McAllister et al. 2017).

Zunächst benötigen ML-Algorithmen eine Möglichkeit, Unsicherheiten abzuschätzen. Das ist bei klassischen tiefen neuronalen Netzen nicht möglich. Das heißt, ein neuronales Netz weiß nicht, wenn es etwas nicht weiß. Es trifft mitunter falsche Vorhersagen, obwohl die ausgegebene Konfidenz sehr hoch ist. Abhilfe können hier sogenannte Bayes'sche neuronale Netze schaffen. Diese ermöglichen die Berechnung einer echten Unsicherheit in Form von Wahrscheinlichkeiten, die zum Erkennen von unvorhergesehenen oder gefährlichen Situationen eingesetzt werden kann. Im Falle einer hohen Unsicherheit könnte ein System zum Beispiel stoppen oder einen menschlichen Aufseher herbeirufen.

Aktuelle Ansätze Bayes'scher neuronaler Netze erfordern allerdings mehr Rechenleistung als vergleichbare normale neuronale Netze. Zudem gibt es häufig keine hochqualitativen frei verfügbaren Implementierungen (Maddox et al. 2019; Wang und Yeung 2016; Izmailov et al. 2019; Blundell et al. 2015; Wu et al. 2018).

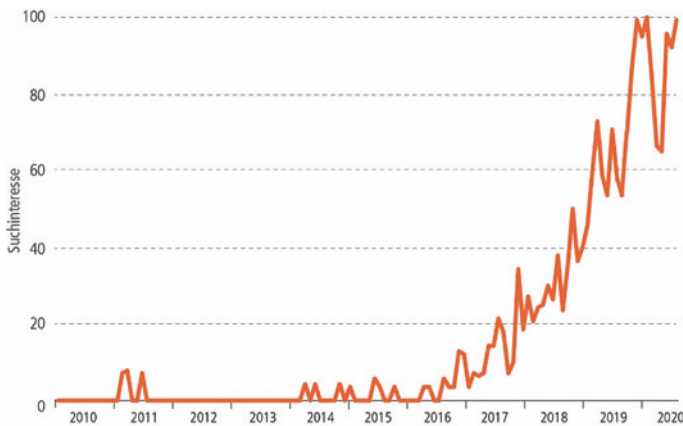
Zusätzlich müssen ML-Algorithmen für viele Anwendungsfälle erklärbar sein. Fast alle modernen ML-Methoden, wie auch tiefe neuronale Netze, sind eine sogenannte Blackbox. Das heißt, dass die innere Funktionsweise und Logik der Algorithmen nach Abschluss des Trainings nicht mehr nachvollzogen werden kann. Erklärbare ML-Verfahren haben hingegen das Ziel, die Funktionsweise nachvollziehbar zu machen. Dies hat den Vorteil, dass man das Risiko eines Einsatzes besser nachvollziehen kann, dass sich die Akzeptanz bei den Mitarbeitern, die von diesen Modellen unterstützt werden, erhöht, und dass Fehler sowie eventuelle Vorurteile der ML-Methoden erkannt werden können. Dies kann zum Beispiel durch die Approximation komplexer Modelle mittels verständlicher Modelle oder durch spieltheoretische Überlegungen ermöglicht werden (Schaaf et al. 2019; Lundberg und Lee 2017).

Im Kontext von KI-Anwendungen ist die Erklärbarkeit zugleich das zentrale Konzept für die digitale Souveränität auf Ebene der Mitarbeitenden. Erklärbarkeit schafft für den Menschen die Voraussetzung, das KI-System soweit näherungsweise zu verstehen, dass er es reflektiert und zielgerichtet im Rahmen seiner Handlungen – basierend auf vorhandenem Wissen und Fähigkeiten – einsetzen kann. Weiterhin eröffnen erklärbare KI-Systeme für die Nutzenden die Chance, über das System und auch über die von System abgebildete Realität (zum Beispiel Produktionsanlagen, Werkstücke) zu lernen und somit die eigene Fähigkeits- und Wissensbasis zu erweitern.

Obwohl in den letzten Jahren viel in diesem Bereich geforscht wurde und das Interesse an erklärbaren ML-Verfahren, sogenannte erklärbare KI, in den letzten Jahren stark gestiegen ist, sind noch zahlreiche Forschungsfragen offen (siehe Abb. 3). Unter anderem gibt es noch keinen standardisierten Prozess, bei dem die

Vergleichbarkeit von Erklärungen, über spezielle Anwendungsfälle hinaus, gegeben ist. Zusätzlich beziehen Erklärungen sich meist auf einzelne Datenpunkte und nicht auf die globale Funktionsweise der Methoden (Ribeiro et al. 2016; Bach et al. 2015; Lundberg und Lee 2017; Wachter et al. 2017).

Schlussendlich ist es bei tiefen neuronalen Netzen nicht möglich, die korrekte Funktionsweise zu verifizieren. Konkret heißt dies, dass man die neuronalen Netze zwar auf sehr vielen Datenpunkten testen kann, aber es nicht möglich ist, sicherzustellen, dass sie in nicht getesteten Situationen auch funktionieren. Darüber hinaus ist es nicht möglich, eine bestimmte Erfolgsrate zu garantieren. Erste Ansätze, diese Probleme zu lösen, versuchen die korrekte Funktionsweise mathematisch zu garantieren und so zum Beispiel eine Sicherheitsgarantie zu liefern. Dies ist mit dem heutigen Stand der Technik für kleine neuronale Netze bereits möglich. Allerdings ist noch zu klären, wie genau die Anforderungen erfasst werden können und wie größere und komplexere ML-Methoden effizient überprüft werden können (Katz et al. 2017; Katz et al. 2019; Huang et al. 2017).



**Abb. 3.** Online-Suchen zum Thema erklärbare KI. Man sieht, dass das Interesse seit 2017 stark ansteigt, wohingegen es vorher kaum bis gar nicht vorhanden war (Google Trends 2020).

### 3 Handlungsbedarfe

Um eine hier beschriebene flexible Automatisierungslösung zu entwickeln, besteht Handlungsbedarf aller beteiligten Akteure.

Auf Seiten der Wirtschaft ist es nötig, das spezialisierte vorhandene Produktionswissen voranzubringen. Das in Europa vorhandene Produktionswissen kann ein entscheidender Faktor in der Entwicklung von hochqualitativen Simulationsumgebungen sein. Die Kombination der Produktionsstärke mit ML-Lösungen kann also die europäische Position festigen – neben der bereits angesprochenen Entwicklung von hochqualitativen Simulationsumgebungen ist auch die Entwicklung von spezialisierten ML-Lösungen für die produzierende Industrie dazu hilfreich.

Darüber hinaus ist es wichtig, die Digitalisierung weiter voranzutreiben, um mit der Zusammenstellung von Datensätzen hoher Qualität und digitalen Zwillingen die Grundlage für die Verwendung von ML-Algorithmen in den zentralen Prozessen zu schaffen.

Auf Seiten der Forschung und Entwicklung ist es essenziell, die anwendungsorientierte Forschung im RL-Bereich voranzutreiben. Es sind große Forschungsanstrengungen und -mittel notwendig, um erstmalig ein Proof of Concept der beschriebenen Lösungen zu entwickeln und diese anschließend zur Marktreife zu bringen. Ein wichtiger Fokuspunkt ist hierbei die Steigerung und die Garantie der Erfolgsrate. Das Ziel muss die flexible prozesssichere Fertigung mittels ML-Methoden sein.

Darüber hinaus müssen auf technischer Seite die Möglichkeiten geschaffen werden, zuverlässige ML-Algorithmen zu entwickeln. Dies umfasst insbesondere das Erkennen und Vermeiden von unvorhergesehenen Situationen sowie die Erklärbarkeit der verwendeten ML-Algorithmen und die Erfüllung einer formalen Sicherheitsgarantie.

Auf der Ebene der Politik muss ein regulatorischer Rahmen geschaffen werden, der die Zertifizierung und Verwendung von ML-Lösungen in sicherheitsrelevanten Situationen möglich macht. Es muss klar werden, was von Seiten der Zertifizierungsstellen verlangt wird, um eine hinreichende Sicherheit der Systeme zu demonstrieren. Dies ist eine Voraussetzung für den breiten Einsatz von ML-Lösungen, da bisher eine große Unsicherheit herrscht. Darüber hinaus ist eine Klärung der Haftungsfrage notwendig. Eine Roadmap zur Normenentwicklung ist aktuell im Entstehen (DIN 2019).

## 4 Ausblick

Im Rahmen der Weiterentwicklung von ML-Methoden wird es möglich sein, eine ML-gestützte Automatisierungslösung zu entwickeln, die den Anforderungen der Zuverlässigkeit – und in diesem Kontext der digitalen Souveränität –, der Sicherheit und der Flexibilität genügt. Wir glauben, dass dies mit den heutigen ML-Algorithmen bereits möglich ist. Es ist wahrscheinlich, dass in den nächsten Jahren ein Proof of Concept hin zu einer robusten Lösung in unterschiedlichen Anwendungsgebieten weiterentwickelt werden kann. In diesem Zuge werden sich forschungsnahe Start-ups bilden und kommerziell verfügbare Lösungen anbieten.

## Literatur

- accenture: Why artificial intelligence is the future of growth (2017). <https://www.accenture.com/za-en/company-news-release-why-artificial-intelligence-future-growth>. Zugegriffen: 1. Juli 2020
- Alom, M.Z., Taha, T.M., Yakopcic, C., Westberg, S., Sidike, P., Nasrin, M.S., van Eesn, B.C., Awwal, A.A.S., Asari, V.K.: The History began from alexnet: a comprehensive survey on deep learning approaches (2018). <https://arxiv.org/pdf/1803.01164>. Zugegriffen: 1. Juli 2020

- ArtiMinds Robotics: Intuitive Roboterprogrammierung mit ArtiMinds Robotics (2020). <https://www.artiminds.com/de/>. Zugegriffen: 1. Juli 2020
- Bach, S., Binder, A., Montavon, G., Klauschen, F., Müller, K.-R., Samek, W.: On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PLOS One* **10** (7), e0130140 (2015). <https://doi.org/10.1371/journal.pone.0130140>. Zugegriffen: 1. Juli 2020
- Bauernhansl, T.: Mass Personalization – der nächste Schritt. wt-online (2017)
- Billard, A., Calinon, S., Dillmann, R., Schaal, S.: Robot programming by demonstration. In: Siciliano, B., Khatib, O. (Hrsg.). *Springer Handbook of Robotics*, S. 1371–1394. Springer & Science+Business Media, Berlin (2008)
- Blundell, C., Cornebise, J., Kavukcuoglu, K., Wierstra, D.: Weight uncertainty in neural networks (2015). <https://arxiv.org/pdf/1505.05424>. Zugegriffen: 1. Juli 2020
- Chebotar, Y., Handa, A., Makoviychuk, V., Macklin, M., Issac, J., Ratliff, N., Fox, D.: Closing the sim-to-real loop: adapting simulation randomization with real world experience. In: 2019 International Conference on Robotics and Automation (ICRA). IEEE, Piscataway (2019)
- covariant.ai: AI robotics for the real world (2020). <https://covariant.ai/>. Zugegriffen: 1. Juli 2020
- DIN: Künstliche Intelligenz I Ohne Normen und Standards geht es nicht. DIN Deutsches Institut für Normung e. V. (2019). <https://www.din.de/de/forschung-und-innovation/themen/kuenstliche-intelligenz/normungsroadmap-ki>. Zugegriffen: 7. Juli 2020
- drag&bot: Industrieroboter wie ein Smartphone bedienen (2020). <https://www.dragandbot.com/de/>. Zugegriffen: 1. Juli 2020
- Duan, Y., Schulman, J., Chen, X., Bartlett, P.L., Sutskever, I., Abbeel, P.: RL<sup>2</sup>: fast reinforcement learning via slow reinforcement learning (2016). <https://arxiv.org/pdf/1611.02779>. Zugegriffen: 1. Juli 2020
- Finn, C., Abbeel, P., Levine, S. Model-agnostic meta-learning for fast adaptation of deep networks (2017). <https://arxiv.org/pdf/1703.03400>. Zugegriffen: 1. Juli 2020
- Florensa, C., Duan, Y., Abbeel, P.: Stochastic neural networks for hierarchical reinforcement learning (2017)
- Goodfellow, I., Bengio, Y., Courville, A.: *Deep Learning*. MIT Press (2016)
- Google Trends: Google trends (2020). <https://trends.google.de/trends/explore?date=2010-01-01%202020-07-01&q=explainable%20ai,interpretable%20machine%20learning,explainable%20artificial%20intelligence>. Zugegriffen: 1. Juli 2020
- Ha, D., Schmidhuber, J.: World models. <https://doi.org/10.5281/zenodo.1207631> (2018). Zugegriffen: 1. Juli 2020
- Hausman, K., Springenberg, J.T., Wang, Z., Heess, N., Riedmiller, M. Learning an embedding space for transferable robot skills. In: International Conference on Learning Representations (2018)
- Hessel, M., Modayil, J., van Hasselt, H., Schaul, T., Ostrovski, G., Dabney, W., Horgan, D., Piot, B., Azar, M., Silver, D.: Rainbow: combining improvements in deep reinforcement learning (2017). <https://arxiv.org/pdf/1710.02298>. Zugegriffen: 1. Juli 2020
- Huang, X., Kwiatkowska, M., Wang, S., Wu, M.: Safety verification of deep neural networks. In: International conference on computer aided verification, 3–29. Springer, Cham (2017)
- Irpan, A.: Deep reinforcement learning doesn't work yet (2020). <https://www.alexirpan.com/2018/02/14/rl-hard.html>. Zugegriffen: 1. Juli 2020
- Izmailov, P., Maddox, W.J., Kirichenko, P., Garipov, T., Vetrov, D., Wilson, A.G.: Subspace inference for Bayesian deep learning (2019). <https://arxiv.org/pdf/1907.07504>. Zugegriffen: 1. Juli 2020

- Johnson, J., Karpathy, A., Fei-Fei, L.: DenseCap: fully convolutional localization networks for dense captioning. In: 29th IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2016: Proceedings : 26 June–1 July 2016, Las Vegas, Nevada. IEEE, Piscataway (2016)
- Katz, G., Barrett, C., Dill, D., Julian, K., Kochenderfer, M.: Reluplex: An efficient SMT solver for verifying deep neural networks. In: International Conference on Computer Aided Verification, S. 97–117. Springer, Cham (2017)
- Katz, G., Huang, D.A., Ibeling, D., Julian, K., Lazarus, C., Lim, R., Shah, P., Thakoor, S., Wu, H., Zeljić, A., Dill, D.L., Kochenderfer, M.J., Barrett, C.: The Marabou framework for verification and analysis of deep neural networks. In: Dillig, I., Tasiran, S. (Hrsg.). Computer Aided Verification. 31st International Conference, CAV 2019, New York, July 15–18, 2019, Proceedings, Part I, Cham, 2019, S. 443–452. Cham, Springer International Publishing (2019)
- Kober, J., Bagnell, J.A., Peters, J.: Reinforcement learning in robotics: a survey. *Int. J. Robot. Res.* **32**(11), 1238–1274 (2013). <https://doi.org/10.1177/0278364913495721>. Zugegriffen: 1. Juli 2020
- Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems, S. 1097–1105 (2012)
- LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. *Nature* **521**(7553), 436–444 (2015). <https://doi.org/10.1038/nature14539>. Zugegriffen: 1. Juli 2020
- Lundberg, S., Lee, S.-I.: A unified approach to interpreting model predictions. In: Advances in Neural Information Processing Systems, S. 4765–4774 (2017)
- Luo, J., Solowjow, E., Wen, C., Ojea, J.A., Agogino, A.M.: Deep reinforcement learning for robotic assembly of mixed deformable and rigid objects. In: Towards a robotic society. 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems: October, 1–5, 2018, Madrid, Spain, Madrid Municipal Conference Centre, 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, 10/1/2018–10/5/2018, Piscataway, NJ, IEEE, S. 2062–2069 (2018)
- Maddox, W.J., Izmailov, P., Garipov, T., Vetrov, D.P., Wilson, A.G.: A simple baseline for Bayesian uncertainty in deep learning. In: Advances in Neural Information Processing Systems, S. 13153–13164 (2019)
- Mahmood, A.R., Korenkevych, D., Vasana, G., Ma, W., Bergstra, J.: Benchmarking reinforcement learning algorithms on real-world robots (2018). <https://arxiv.org/pdf/1809.07731>. Zugegriffen: 1. Juli 2020
- McAllister, R., Gal, Y., Kendall, A., van der Wilk, M., Shah, A., Cipolla, R., Weller, A.: Concrete problems for autonomous vehicle safety: advantages of Bayesian deep learning, S. 1045–0823 (2017). <https://doi.org/10.17863/CAM.12760>. Zugegriffen: 1. Juli 2020
- Nachum, O., Gu, S., Lee, H., Levine, S.: Data-efficient hierarchical reinforcement learning (2018). <https://arxiv.org/pdf/1805.08296>. Zugegriffen: 1. Juli 2020
- Peng, X.B., Andrychowicz, M., Zaremba, W., Abbeel, P.: Sim-to-real transfer of robotic control with dynamics randomization. In: Lynch, K. (Hrsg.). 2018 IEEE International Conference on Robotics and Automation (ICRA). 21–25 May 2018. IEEE, Piscataway (2018)
- Piller, F.T.: Mass Customization. Ein wettbewerbsstrategisches Konzept im Informationszeitalter. Zugl.: Würzburg, Univ., Diss., 1999 u. d. T.: Kundenindividuelle Massenproduktion (mass customization) als wettbewerbsstrategisches Modell industrieller Wertschöpfung in der Informationsgesellschaft. 4. Aufl. Wiesbaden, Dt. Univ.-Verl. (2008)
- Polydoros, A.S., Nalpantidis, L.: Survey of Model-Based Reinforcement Learning: Applications on Robotics. *Journal of Intelligent & Robotic Systems* **86** (2), 153–173 (2017). <https://doi.org/10.1007/s10846-017-0468-y>. Zugegriffen: 1. Juli 2020

- Ribeiro, M., Singh, S., Guestrin, C.: "Why should i trust you?": Explaining the predictions of any classifier. In: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations, Stroudsburg, PA, USA. Association for Computational Linguistics, , Stroudsburg, PA, USA (2016)
- Schaaf, N., Huber, M., Maucher, J.: Enhancing decision tree based interpretation of deep neural networks through L1-orthogonal regularization. In: Wani, M.A. (Hrsg.). ICMLA 2019. 18th IEEE International Conference on Machine Learning and Applications: Proceedings: 16–19 December 2019, Boca Raton, Florida, USA, Conference Publishing Services, IEEE Computer Society, Los Alamitos (2019)
- Schoettler, G., Nair, A., Luo, J., Bahl, S., Ojea, J.A., Solowjow, E., Levine, S.: Deep reinforcement learning for industrial insertion tasks with visual inputs and natural reward signals (2019)
- Schulman, J., Levine, S., Moritz, P., Jordan, M.I., Abbeel, P.: Trust Region Policy Optimization (2015). <https://arxiv.org/pdf/1502.05477>. Zugegriffen: 1. Juli 2020
- Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., Chen, Y., Lillicrap, T., Hui, F., Sifre, L., van den Driessche, G., Graepel, T., Hassabis, D.: Mastering the game of go without human knowledge. *Nature* **550**(7676), 354–359 (2017). <https://doi.org/10.1038/nature24270>. Zugegriffen: 1. Juli 2020
- Spinnarke, S.: So wird Künstliche Intelligenz in der Produktion eingesetzt (2017). <https://www.produktion.de/trends-innovationen/so-wird-kuenstliche-intelligenz-in-der-produktion-eingesetzt-104.html>. Zugegriffen: 1. Juli 2020
- Sutton, R.S., Barto, A.: Reinforcement Learning. An Introduction. MIT Press, Cambridge (2018)
- Tan, C., Sun, F., Kong, T., Zhang, W., Yang, C., Liu, C.: A survey on deep transfer learning. In: Kůrková, V., Manolopoulos, Y., Hammer, B. et al. (Hrsg.). In: Artificial Neural Networks and Machine Learning – ICANN 2018. 27th International Conference on Artificial Neural Networks, Rhodes, Greece, October 4–7, 2018, Proceedings, Part III, Cham, 2018, S. 270–279. Springer International Publishing, Cham (2018)
- Tempel, P., Eger, F., Lechler, A. et al.: Schaltschrankbau 4.0: Eine Studie über die Automatisierungs- und Digitalisierungspotenziale in der Fertigung von Schaltschränken und Schaltanlagen im klassischen Maschinen- und Anlagenbau (2017)
- Tobin, J., Fong, R., Ray, A., Schneider, J., Zaremba, W., Abbeel, P.: Domain randomization for transferring deep neural networks from simulation to the real world. In: IROS Vancouver 2017. IEEE/RSJ International Conference on Intelligent Robots and Systems, Vancouver, BC, Canada September 24–28, 2018, S. 2062–2069. IEEE, Piscataway
- Vezhnevets, A.S., Osindero, S., Schaul, T., Heess, N., Jaderberg, M., Silver, D., Kavukcuoglu, K.: FeUdal networks for hierarchical reinforcement learning (2017). <https://arxiv.org/pdf/1703.01161>
- Wachter, S., Mittelstadt, B., Russell, C.: Counterfactual explanations without opening the black box: automated decisions and the GDPR. *SSRN Electron. J.* (2017). <https://doi.org/10.2139/ssrn.3063289>. Zugegriffen: 1. Juli 2020
- Wang, H., Yeung, D.-Y.: Towards Bayesian deep learning: a survey (2016). <http://arxiv.org/pdf/1604.01662v2>. Zugegriffen: 1. Juli 2020
- Waschneck, B., Reichstaller, A., Belzner, L., Altenmüller, T., Bauernhansl, T., Knapp, A., Kyek, A.: Optimization of global production scheduling with deep reinforcement learning. *Procedia CIRP* **72**, 1264–1269 (2018). <https://doi.org/10.1016/j.procir.2018.03.212>. Zugegriffen: 1. Juli 2020
- Westkämper, E. (Hrsg.): Towards the Re-Industrialization of Europe. A Concept for Manufacturing for 2030. Springer, Berlin (2014)

- Wu, A., Nowozin, S., Meeds, E., Turner, R.E., Hernández-Lobato, J.M., Gaunt, A.L.: Deterministic variational inference for robust Bayesian neural networks (2018)
- Xanthopoulos, A.S., Kiatipis, A., Koulouriotis, D.E., Stieger, S.: Reinforcement learning-based and parametric production-maintenance control policies for a deteriorating manufacturing system. *IEEE Access* **6**, 576–588 (2018). <https://doi.org/10.1109/ACCESS.2017.2771827>. Zugegriffen: 1. Juli 2020
- Zeng, A., Song, S., Lee, J., Rodriguez, A., Funkouser, A.T.: TossingBot: Learning to throw arbitrary objects with residual physics. In: *Robotics: Science and Systems XV*. Robotics: Science and Systems Foundation (2019)

**Open Access** Dieses Kapitel wird unter der Creative Commons Namensnennung 4.0 International Lizenz (<http://creativecommons.org/licenses/by/4.0/deed.de>) veröffentlicht, welche die Nutzung, Vervielfältigung, Bearbeitung, Verbreitung und Wiedergabe in jeglichem Medium und Format erlaubt, sofern Sie den/die ursprünglichen Autor(en) und die Quelle ordnungsgemäß nennen, einen Link zur Creative Commons Lizenz beifügen und angeben, ob Änderungen vorgenommen wurden.

Die in diesem Kapitel enthaltenen Bilder und sonstiges Drittmaterial unterliegen ebenfalls der genannten Creative Commons Lizenz, sofern sich aus der Abbildungslegende nichts anderes ergibt. Sofern das betreffende Material nicht unter der genannten Creative Commons Lizenz steht und die betreffende Handlung nicht nach gesetzlichen Vorschriften erlaubt ist, ist für die oben aufgeführten Weiterverwendungen des Materials die Einwilligung des jeweiligen Rechteinhabers einzuholen.

