

Was leisten die Domänen der Künstlichen Intelligenz?

Machine Learning, Deep Learning, Reinforcement Learning, ...

von Dimitri Gross

Immer mehr führende Technologieunternehmen investieren derzeit in Künstliche Intelligenz und sehen hier einen zukunftsweisenden Markt [For17]. Gleichzeitig tauchen immer wieder negative Schlagzeilen zur modernen Industrialisierung auf: Kritiker warnen davor, dass Menschen auch dort, wo bislang der Faktor Mensch als eine Konstante galt, durch Maschinen ersetzt werden [Spi16]. Umso wichtiger ist es also, sich ein genaues Bild von der Technologie und ihren Möglichkeiten zu machen. In diesem Artikel gehen wir auf die grundlegende Funktionsweise eines neuronalen Netzwerks ein. Neuronale Netzwerke stellen einen Teil der grundlegenden Technologie für die Entwicklung Künstlicher Intelligenz dar. Die Wurzeln dieser Verfahren reichen in die 60er-Jahre des vergangenen Jahrhunderts zurück. Doch wie beim Begriff Künstliche Intelligenz gibt es auch bei den neuronalen Netzwerken viele Ausprägungen. Wir konzentrieren uns auf die Funktionsweise der modernsten Verfahren aus dem Bereich der neuronalen Netze und schauen uns diese genauer an.

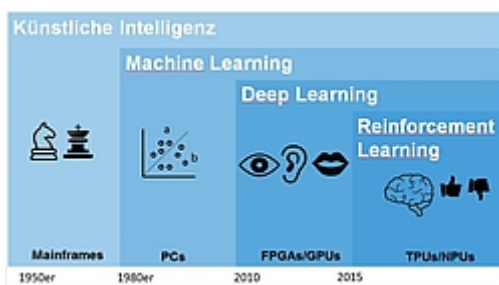


Abb. 1: Entscheidende Schlüsseltechnologien innerhalb der Domänen der Künstlichen Intelligenz (erweiterte Eigendarstellung nach [Nvi16])

Bevor wir uns mit den technologischen Aspekten befassen, betrachten wir die Domänen der Künstlichen Intelligenz. Unter dem Oberbegriff Künstliche Intelligenz werden grob genommen drei Bereiche zusammengefasst, die wir im Alltag bewusst oder unbewusst nutzen: *Machine Learning*, *Deep Learning* und *Reinforcement Learning*. Fangen wir aber zunächst einmal ganz vorne an.

Die Zeitrechnung der Künstlichen Intelligenz beginnt in den 50er-Jahren, als das erste Schachprogramm entwickelt wurde. Das Privileg, mit dem Computer eine Schachpartie spielen zu können, hatten damals nur wenige Menschen. Diese Menschen arbeiteten gewöhnlich in wissenschaftlichen Einrichtungen und verwandelten Mainframes und Großrechner, die eigentlich

militärischen oder wissenschaftlichen Zwecke dienten, heimlich in ihrer Freizeit mithilfe eines Schachprogramms in eine Art künstlich intelligenten Gegenspieler.

Die Entwicklung in der Computerindustrie machte rasante Schritte nach vorn und nur 30 Jahre später waren Computerspiele dieser Art einer breiten Menschengruppe zugänglich. Wenn wir heute von Künstlicher Intelligenz sprechen, meinen wir nicht nur Spiele, sondern wir meinen vor allem auch mathematische Verfahren, die uns helfen, Prozesse zu automatisieren. Dafür werden Algorithmen automatisiert angewendet. Was uns in den Bereich des Machine Learnings bringt.

Machine Learning

Machine Learning (maschinelles Lernen) ist im Grunde nur ein Oberbegriff für statistische Verfahren, mit deren Hilfe eine Klassifikation, Regression oder Clustering innerhalb einer Datenmenge vorgenommen werden kann. Damit können beispielsweise Anomalien entdeckt werden. Machine Learning teilt sich in zwei große Domänen auf, das Supervised und Unsupervised Learning.

Beim *Supervised Learning* (überwachtes Lernen) haben wir es mit einer sogenannten statistischen Modellierung zu tun. Dabei bauen wir auf einem historischen Datensatz zunächst ein statistisches Modell auf und suchen sukzessive passende Verfahren für die zu analysierenden Daten aus, um anschließend mit einem statistischen Modell alle identisch strukturierten neuen Daten zu klassifizieren. Da wir die Vergangenheit in einem Modell gut nachgebildet haben, erkennen wir auch statistische Paare in den Daten und können damit Rückschlüsse auf die Zukunft ziehen. Hierfür werden historische Daten mit bekannten Klassenlabels verwendet. Sowohl das Resultat einer gewissen Datenkonstellation als auch die Datenkonstellation selbst sind somit bekannt. *Unsupervised Learning* (unüberwachtes Lernen) bringt sozusagen Ordnung ins Chaos. Mit Chaos ist in diesem Fall gemeint, dass wir weder historische Daten besitzen noch bei neuen Daten die Korrelationen zwischen Ausgangslage und Resultat kennen. Hier kommen die unsupervised Verfahren zum Einsatz, die in der Lage sind, statistisch ähnliche Paare innerhalb einer großen Datenmenge zu clustern. In einem zweiten Schritt kann anschließend die Klassifikation mit Supervised Learning erfolgen. Und das immer wieder aufs Neue.

Damit wäre der Begriff Machine Learning grob erklärt. Bei den Analyseverfahren des Supervised Learning treffen wir unter anderem ein neuronales Netzwerk an.

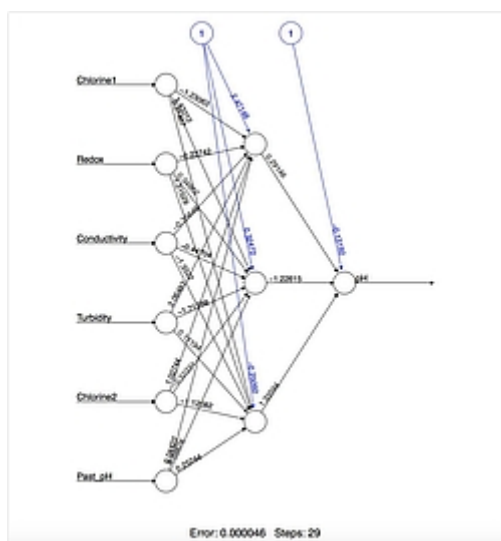


Abb. 2: Schematische Darstellung eines neuronalen Netzwerkes am Beispiel einer pH-Ermittlung im Wasser anhand diverser Messwerte [DSBMS17])

Neuronales Netzwerk

Abbildung 2 zeigt ein neuronales Netzwerk aus einem Beispielprojekt. Ein neuronales Netzwerk besteht aus mehreren Ebenen, im Beispiel sind es drei. Jede Ebene kann mehrere Neuronen haben, hier als Kreise dargestellt. In diesem Modell kann jedes Neuron einen Wert annehmen. Alle Neuronen sind untereinander verknüpft und jede Verbindung wird mit einem Gewichtungsfaktor multipliziert. Der Gewichtungsfaktor passt sich automatisch mit jeder Iteration im Trainingsprozess an. So ist das neuronale Netzwerk bei einer ähnlichen Konstellation an Inputparametern (Ausgangslage) in der Lage, das Modell richtig zu gewichten und den zuvor eintrainierten Output (Resultat) wiederzuerkennen und anzugeben.

Die Möglichkeit, eine Lehre aus Fehlern zu ziehen (Backpropagation) und die Gewichte neu zu justieren, um sich dem gegebenen Output zu nähern, macht dieses Verfahren sehr interessant. Auf der anderen Seite kann das Netzwerk auch „überlernt“ werden

(Overfitting). Das Modell scheint dann perfekt zu funktionieren und erkennt auch jedes Detail, das antrainiert wurde. Doch wird das Modell mit neuen Daten konfrontiert, stimmen die Klassifikationen nicht mehr und die Ergebnisse fallen falsch aus. Ein weiterer Nachteil solcher Modelle ist, dass man nicht genau weiß, wie die Gewichtung verteilt wird und welche Neuronen später einen Einfluss auf die Klassifizierung haben. Das heißt, wir können bei einer Eingabe und einer vom Netzwerk produzierten Ausgabe keinen Rückschluss auf die Elemente der Eingabe ziehen und wissen daher nicht, wie es zu der Ausgabe kam. Es passiert etwas und danach funktioniert es in einigen dafür geeigneten Bereichen ganz gut. So kann zum Beispiel das Rauschen aus Signalen herausgefiltert werden oder anhand von unterschiedlichen Messwerten der pH-Wert von Wasser ermittelt werden.

Dieses einfache, dreischichtige neuronale Netzwerk ist aber nicht die einzige Ausprägung dieses Zweigs der Computerwissenschaft. Es geht noch tiefer. In dem Fall sprechen wir vom Deep Learning.

Deep Learning

Als Unterbereich des Machine Learning fokussiert sich Deep Learning („tiefgehendes Lernen“) nur auf eine bestimmte, komplexe Art von neuronalen Netzwerken, die auch als gefaltete neuronale Netzwerke (Convolutional Neural Networks, kurz CNN) bezeichnet werden. Dieser Zweig beschäftigt sich hauptsächlich mit der Erforschung von Wahrnehmungsproblematiken wie Sehen, Hören und Sprechen. Die Entstehung von Deep Learning und sein Einzug in unseren Alltag ist der allgemeinen Hardwareentwicklung seit den 80er-Jahren zu verdanken. Hier haben die FPGAs (Field Programmable Gate Array) und später auf die GPUs (Graphics Processing Unit) adaptierten Verfahren ihre Wurzeln. Durch die Möglichkeit, große Datenmengen schnell mithilfe von verteilten GPU-Clustern zu analysieren und daraus zu lernen, entstehen beispielsweise Modelle, die zum einen eine nahezu perfekt klingende menschliche Sprache nachbilden und zum anderen auch jede gesprochene Sprache verstehen und in Textform übersetzen können. So werden zum Beispiel Sprachassistenten alltagstauglich. Ein Beispiel, das momentan viel genutzt wird, sind die Assistenzsysteme im Auto. Auch hier geht es nicht ohne Objekterkennung.

Objekterkennung

Wie die Objekterkennung funktioniert, schauen wir uns am Beispiel eines CNN (Convolutional Neural Network) an: Das Beispiel-CNN wurde dem visuellen Cortex einer Katze nachempfunden. Das Besondere an diesem Gehirnbereich ist, dass die Bildsignale ein komplexes Netz an Neuronen durchlaufen, bevor sie im Gehirn weiterverarbeitet werden. Zusätzlich wurden zwei Zelltypen entdeckt, die für das Extrahieren von Merkmalen aus einem Bildbereich zuständig sind. Ein Zelltyp reagiert auf Kanten und grobe Muster innerhalb eines Sichtbereichs. Diese Zellen reagieren auf kleine Ausschnitte im Sichtfeld und fungieren somit als eine Art Filter. Ein weiterer Zelltyp reagiert auf komplexere Muster und dies unabhängig von deren Position im Sichtfeld [CNN16].



Abb. 3: Anwendung eines Filters am Beispiel einer Iteration (nach [CS231n16])

Abbildung 3 zeigt zusammenfassend die Funktionsweise eines CNN anhand einer Faltungsstufe.

1. Im ersten Schritt werden Filter unterschiedlicher Art auf ein Bild angewandt. Diese können Konturen in einem Bild erkennen oder Farbmuster extrahieren. Ähnlich wie die Zellen im visuellen Cortex einer Katze, die auch auf bestimmte Kanten und Muster reagieren.
2. Die mithilfe der Filter errechneten Werte werden in einem neuen Bild persistiert. So wird eine Activation Map berechnet, die durch die Reaktion auf eine bestimmte Filterart die markanten Formen stark hervorhebt. Das Ergebnis der ersten Faltungsstufe wird in Form von mehreren Activation Maps an die nachfolgende Stufe übergeben. Die Anzahl der Maps gleicht der Anzahl der im ersten Schritt angewandten Filter.
3. Im letzten Schritt werden die gewonnenen Merkmale aus jeder Faltungsstufe einer Objektklasse zugeordnet, die stärksten Aktivierungen werden dabei höher gewichtet. Eine Objektklasse beinhaltet somit alle markanten Objektteile von allen Bildern, die zu dieser Objektklasse gehören.

CNNs eignen sich somit nicht nur für Daten wie „Katze“ plus Katzenbild, „Hund“ plus Hundebild, sondern in einer angepassten Variante lassen sich auch Audiosignale einspeisen. Das Audio kommt dabei als Sinusform daher. In regelmäßigen Abschnitten bekommt das neuronale Netzwerk zusätzlich die Labels dazu, in Form von Lauten oder ganzen Wörtern. Als Ergebnis kommt bei

der Spracheingabe Text heraus. Dass das sehr gut funktioniert, beweisen die allgegenwärtigen Sprachassistenten in Mobiltelefonen und stationäre Assistenten wie Alexa von Amazon.

RNN

Eine weitere interessante, häufig genutzte Art eines neuronalen Netzwerks ist das RNN (Recurrent Neural Network). RNNs erzielen gute Resultate bei der Textverarbeitung und kommen auch dort zum Einsatz, wo keine fixen Inputs oder Outputs definiert sind. Sprich: Hier darf die Inputlänge (Wörter) variieren, was bei CNNs nicht möglich ist. Von der Idee her funktionieren diese Netzwerke ähnlich wie CNNs, nur mit dem Unterschied, dass die Ergebnisse nicht streng iterativ an einen weiteren Layer gegeben, sondern innerhalb des gleichen Layers als Input genutzt werden.

Jedes Neuron bekommt also nicht nur eine Information vom vorhergehenden Neuron, sondern kann auch sein eigenes Ergebnis aus der vorherigen Iteration als Input für die nächste Iteration nutzen.

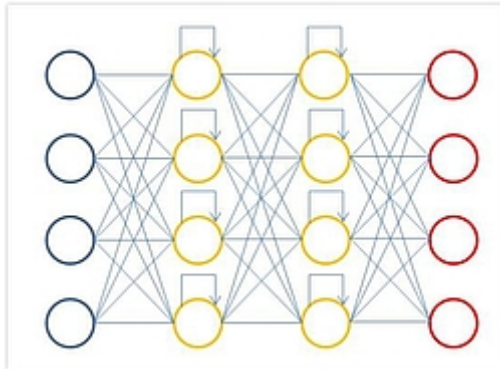


Abb 4: Recurrent Neural Network

Durch diese spezielle Verbindung zwischen Layern und Neuronen spielt die Reihenfolge, in der Trainingsdaten in das Netzwerk eingegeben werden, eine große Rolle. Somit macht es bei dieser Art von neuronalen Netzwerken einen großen Unterschied, wenn beim Training die Wortreihenfolge geändert wird. Sprich: Wenn wir zuerst Brot und dann Butter sagen, ergibt dies einen anderen Zusammenhang als wenn Butter zuerst eingegeben wird. Durch diese Eigenschaft, Eingabewerte miteinander zu verbinden, eignen sich solche Netzwerke beispielsweise für die automatische Wortvervollständigung bei der Eingabe von Texten. Nicht unbedingt geeignet sind RNNs für Aufgaben, die Kontinuität über ein Zeitintervall aufweisen, wie beispielsweise Video oder Audio. Spezielle Ausprägungen wie LSTM-RNNs erzielen hier gute Resultate. Ein richtiger Durchbruch ist hier Deepmind mit WaveNET, das einen anderen Weg verfolgt als LSTM-RNNs.

Reinforcement Learning

Reinforcement Learning (bestärkendes Lernen) ist ein relativ neues Betätigungsfeld. In Kombination mit den Anwendungsfällen, die durch den Fortschritt im Deep Learning getrieben werden, eröffnen sich ungeahnte Horizonte in einer hochautomatisierten Welt. Roboterarme können in der Fertigung selbstständig agieren und fahren nicht nur vorprogrammierte Pfade und Aktivitäten ab [GHLL16]. Beim Reinforcement Learning wird die Umgebung als solche betrachtet und der Algorithmus stellt eine Art Agenten dar. Dieser Agent lernt mit der Zeit, eine gezeigte Aktion in Perfektion auszuführen oder er entwickelt einen völlig neuen Ansatz, um eine ihm bis dahin noch nicht bekannte Aufgabe zu lösen. Das Funktionsprinzip dahinter ist spannend: In regelmäßigen Abständen bekommt der Agent Feedback auf seine Interaktion mit der Umgebung. Das Feedback kann positiv oder negativ sein. Somit ist der Agent in der Lage, über die Zeit mit einer steigenden Anzahl an Iterationen eine Kostenfunktion zu berechnen und anhand dieser seine Strategie zu steuern, um die Effizienz seiner Aktionen zu steigern.

Wie gut solche Verfahren funktionieren, hat eine Gruppe von Wissenschaftlern an der Carnegie Mellon University bewiesen. In einem Versuch ließen sie den Agenten eine Umgebung beobachten und Aktionen testen. Die Umgebung wurde durch eine Pixelmatrix abgebildet, die mit der grafischen Ausgabe des Computerspiels „Doom“ verknüpft war. So konnte der Algorithmus die Pixel beobachten und darin nach und nach Zusammenhänge finden, um mit der Zeit Raumgeometrie und Gegner zu unterscheiden und sich entsprechend zu bewegen und zu schießen. Das Resultat war verblüffend. Der Algorithmus spielte dieses Spiel am Ende in einer hohen Geschwindigkeit und kam mit allen Schwierigkeiten zurecht. [LCh016]

Fazit

Große Grafikchip-Hersteller wie AMD und Nvidia stellen bereits Software zur Verfügung, um auf ihrer hauseigenen Hardware Deep-Learning-Technologien zu nutzen. Entwicklungen wie diese erinnern an die Entwicklung der 3D-Computerspiele, als 3Dfx eine neue Ära einläutete. Dieser Trend setzt sich aktuell fort: Intel ist durch eine große Mergers-&Acquisitions-Aktion in der Lage, eigene KI-Chips mit Neural Processing Units (NPU) zu produzieren und diese auf USB-Sticks anzubieten. Der Fokus des Herstellers liegt dabei auf dem Internet der Dinge (IoT) und auf Drohnen. Künstliche Intelligenz wird genutzt, um etwa autonom zu fliegen [Hei17]. Google hat ebenfalls eigene Chips entwickelt, sogenannte Tensor Processing Units (TPUs). Diese Chips arbeiten viel effizienter und somit auch schneller als GPUs, wenn es um die Berechnung von neuronalen Netzwerken geht. Dies bewies das AlphaGo-System beim Spiel Go im Duell gegen den weltbesten Spieler Lee Sedol [Wired17].

Künstliche Intelligenz bietet uns also derzeit viele neue Möglichkeiten und der Markt dafür ist riesig. Dabei sollte die menschliche Ethik nicht außer Acht gelassen werden. Aber wie bei jeder neuen Technologie, muss sich das Konstrukt erst formen.

Literatur und Links

[For17] The Amazing Ways Google Uses Deep Learning AI,

<https://www.forbes.com/sites/bernardmarr/2017/08/08/the-amazing-ways-how-google-uses-deep-learning-ai>.

[DSBMS17] Trinkwassersicherheit mit Predictive Analytics und Oracle, Dimitri Gross, Ralf Seger, OPITZ CONSULTING Deutschland GmbH, Prof. Thomas Bartz-Beielstein, Steffen Moritz, Jan Strohschein, Technische Hochschule Köln.

[LCh016] Playing FPS Games with Deep Reinforcement Learning, Guillaume Lample, Devendra Singh Chaplot, Carnegie Mellon University 2016.

[CNN16] Convolutional Neural Networks (LeNet) DeepLearning 0.1 documentation, <http://deeplearning.net/tutorial/lenet.html>.

[CS231n16] CS231n Convolutional Neural Networks for Visual Recognition, <http://cs231n.github.io/convolutional-networks/>.

[Nvi16] https://blogs.nvidia.com/wp-content/uploads/2016/07/Deep_Learning_Icons_R5_PNG.jpg.png.

[Wired17] Google's AlphaGo Levels Up From Board Games to Power Grids,

<https://www.wired.com/2017/05/googles-alphago-levels-board-games-power-grids/>.

[Spi16] <https://magazin.spiegel.de/SP/2016/36/>.

[GHLL16] Deep Reinforcement Learning for Robotic Manipulation with Asynchronous Off-Policy Updates, Shixiang Gu, Ethan Holly, Timothy Lillicrap, Sergey Levine, <https://arxiv.org/abs/1610.00633>.

[Hei17] Movidius Neural Compute Stick für maschinelles Sehen,

<https://www.heise.de/newsticker/meldung/1-Watt-Rechenstick-Movidius-Neural-Compute-Stick-fuer-maschinelles-Sehen-3780324.html>.



Dimitri Gross

arbeitet als Managing Consultant Solutions bei der OPITZ CONSULTING Deutschland GmbH. Hier leitet er im Competence Center Big Data den Bereich Big-Data-Architektur. Außerdem beschäftigt er sich

mit Werkzeugauswahl, Lösungsdesign und Aufbauorganisation in Big-Data-Projekten. Darüber hinaus ist er als Dozent an der Hochschule für Ökonomie & Management München tätig, tritt als Sprecher auf TDWI und anderen Konferenzen auf und ist Autor mehrerer Fachpublikationen im Bereich Big Data und Künstliche Intelligenz.

E-Mail: [Dimitri.Gross\(at\)opitz-consulting.com](mailto:Dimitri.Gross(at)opitz-consulting.com)

Bildnachweise:

OPITZ CONSULTING Deutschland GmbH

[Online Themenspecial](#)

[Impressum](#)

|

[Kontakt & Anfrage](#)