Deep Learning

Wikipedia gehört Ihnen.

Bitte ignorieren Sie diese kurze Nachricht nicht. Wir brauchen Ihre Unterstützung. Es ist wichtig. Wikipedia war zu Beginn einer der ersten Orte im Internet, an denen Menschen gratis und werbefrei lernen konnten. Dieser Ort gehört Ihnen. Nur 2 % unserer LeserInnen spenden. Daher hilft alles, was Sie geben können, ob 5 CHF oder 25 CHF. — *Wikimedia CH*

VISA MasterCard AMEX PayPal TWINT Einzahlungsschein anfordern

ICH HABE SCHON GESPENDET SCHLIESSEN ✕

Deep Learning (deutsch: *mehrschichtiges Lernen, tiefes Lernen*[2] oder *tiefgehendes Lernen*) bezeichnet eine Methode des maschinellen Lernens, die künstliche neuronale Netze (KNN) mit zahlreichen Zwischenschichten (englisch *hidden layers*) zwischen Eingabeschicht und Ausgabeschicht einsetzt und dadurch eine umfangreiche innere Struktur herausbildet.[3][4][5] Es ist eine spezielle Methode der Informationsverarbeitung.

Geschichtete Repräsentation von Bildern auf mehreren Abstraktionsebenen[1]

Inhaltsverzeichnis [Verbergen]

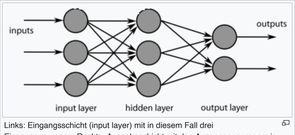
1 Voraussetzungen und Grundlagen
2 Geschichte, Entwicklung und Verwendung
3 Komplexität und Grenzen der Erklärbarkeit
4 Programmbibliotheken
5 Literatur
6 Weblinks
7 Einzelnachweise

Voraussetzungen und Grundlagen [Bearbeiten | Quelltext bearbeiten]

Die in der Anfangszeit der künstlichen Intelligenz gelösten Probleme waren für den Menschen intellektuell schwierig, aber für Computer einfach zu verarbeiten. Diese Probleme ließen sich durch formale mathematische Regeln beschreiben. Die wahre Herausforderung an die künstliche Intelligenz bestand jedoch in der Lösung von Aufgaben, die für die Menschen leicht durchzuführen sind, deren Lösung sich aber nur schwer durch mathematische Regeln formulieren lassen. Dies sind Aufgaben, die der Mensch intuitiv löst, wie zum Beispiel Sprach- oder Gesichtserkennung.[3][6]

Eine computerbasierte Lösung für diese Art von Aufgaben beinhaltet die Fähigkeit von Computern, aus der Erfahrung zu lernen und die Welt in Bezug auf eine Hierarchie von Konzepten zu verstehen. Hierbei ist jedes Konzept durch seine Beziehung zu einfacheren Konzepten definiert. Durch das Sammeln von Wissen aus der Erfahrung vermeidet dieser Ansatz die Notwendigkeit für die menschlichen Bediener, all das Wissen, das der Computer für seine Arbeit benötigt, formal spezifizieren zu müssen. Die Hierarchie der Konzepte erlaubt es dem Computer, komplizierte Konzepte zu erlernen, indem er sie aus einfacheren zusammensetzt. Wenn man ein Diagramm zeichnet, das zeigt, wie diese Konzepte übereinander aufgebaut werden, dann ist das Diagramm tief, mit vielen Schichten. Aus diesem Grund wird dieser Ansatz in der künstlichen Intelligenz „Deep Learning" genannt.[3][7]

Es ist schwierig für einen Computer, die Bedeutung von rohen sensorischen Eingangsdaten zu verstehen, wie beispielsweise in der Handschrifterkennung, wo ein Text zunächst nur als eine Sammlung von Bildpunkten existiert. Die Überführung einer Menge von Bildpunkten in eine Kette von Ziffern und Buchstaben ist sehr kompliziert. Komplexe Muster müssen aus Rohdaten extrahiert werden. Das Lernen oder Auswerten dieser Zuordnung scheint unüberwindbar schwierig, wenn sie manuell programmiert werden würde.[3]



Links: Eingangsschicht (input layer) mit in diesem Fall drei Eingangsneuronen. Rechts: Ausgabeschicht mit den Ausgangsneuronen, in diesem Bild zwei. Die mittlere Schicht wird als verborgen bezeichnet (hidden layer), da ihre Neuronen weder Eingänge noch Ausgänge sind. Hier ist nur eine verborgene Schicht zu sehen, aber viele Netzwerke haben deutlich mehr. Die notwendige Anzahl von Ebenen, ab denen man von „Deep Learning" spricht, ist nicht genau festgelegt.

Eine der häufigsten Techniken in der künstlichen Intelligenz ist maschinelles Lernen. Maschinelles Lernen ist ein selbstadaptiver Algorithmus. Deep Learning, eine Teilmenge des maschinellen Lernens, nutzt eine Reihe hierarchischer Schichten bzw. eine Hierarchie von Konzepten, um den Prozess des maschinellen Lernens durchzuführen. Die hierbei benutzten künstlichen neuronalen Netze sind wie das menschliche Gehirn gebaut, wobei die Neuronen wie ein Netz miteinander verbunden sind. Die erste Schicht des neuronalen Netzes, die sichtbare Eingangsschicht, verarbeitet eine Rohdateneingabe, wie beispielsweise die einzelnen Pixel eines Bildes. Die Dateneingabe enthält Variablen, die der Beobachtung zugänglich sind, daher „sichtbare Schicht".[3]

Diese erste Schicht leitet ihre Ausgaben an die nächste Schicht weiter. Diese zweite Schicht verarbeitet die Informationen der vorherigen Schicht und gibt das Ergebnis ebenfalls weiter. Die nächste Schicht nimmt die Informationen der zweiten Schicht entgegen und verarbeitet sie weiter. Diese Schichten werden als versteckte Ebenen (englisch *hidden layers*) bezeichnet. Die in ihnen enthaltenen Merkmale werden zunehmend abstrakt. Ihre Werte sind nicht in den Ursprungsdaten angegeben. Stattdessen muss das Modell bestimmen, welche Konzepte für die Erklärung der Beziehungen in den beobachteten Daten nützlich sind. Dies geht über alle Ebenen des künstlichen neuronalen Netzes so weiter. Das Ergebnis wird in der sichtbaren letzten Schicht ausgegeben. Hierdurch wird die gewünschte komplizierte Datenverarbeitung in eine Reihe von verschachtelten einfachen Zuordnungen unterteilt, die jeweils durch eine andere Schicht des Modells beschrieben werden.[3][4][7][8]

Geschichte, Entwicklung und Verwendung [Bearbeiten | Quelltext bearbeiten]

Die *Group method of data handling*-KNNs (GMDH-ANN) der 1960er-Jahre von Oleksij Iwachnenko waren die ersten Deep-Learning-Systeme des Feedforward-Multilayer-Perzeptron-Typs.[9][10][11] Karl Steinbuchs Lernmatrix[12] war eines der ersten künstlichen neuronalen Netze, das aus mehreren Schichten von Lerneinheiten oder lernenden „Neuronen" bestand. Damit war er einer der Wegbereiter des Deep Learning, bei dem es um tiefe neuronale Netze geht, die viele Aufgaben erlernen können, bei denen früheren einschichtige Perzeptronen scheitern. Weitere Deep-Learning-Ansätze, vor allem aus dem Bereich des maschinellen Sehens, begannen mit dem Neocognitron, einer Convolutional Neural Network (CNN) Architektur, die von Kunihiko Fukushima 1980 entwickelt wurde.[13] Alex Waibels CNN namens TDNN (1987) wurde durch backpropagation trainiert und erzielte Bewegungsinvarianz.[14] Im Jahr 1989 verwendeten Yann LeCun und Kollegen den Backpropagation-Algorithmus für das Training mehrschichtiger KNNs (siehe Multi-Layer-Perzeptron), mit dem Ziel, handgeschriebene Postleitzahlen zu erkennen.[15] Sven Behnke hat seit 1997 in der Neuronalen Abstraktionspyramide[16] den vorwärtsgerichteten hierarchisch-konvolutionalen Ansatz durch seitliche und rückwärtsgerichtete Verbindungen erweitert, um so flexibel Kontext in Entscheidungen einzubeziehen und iterativ lokale Mehrdeutigkeiten aufzulösen.

Der Begriff „Deep Learning" wurde im Kontext des maschinellen Lernens erstmals 1986 von Rina Dechter verwendet, wobei sie hiermit ein Verfahren bezeichnet, bei dem alle verwendeten Lösungen eines betrachteten Suchraums aufgezeichnet werden, die zu keiner gewünschten Lösung geführt haben. Die Analyse dieser aufgezeichneten Lösungen soll es ermöglichen, anschließende Versuche besser zu steuern und somit mögliche Sackgassen in der Lösungsfindung frühzeitig zu verhindern.[17] Heutzutage wird der Begriff jedoch vorwiegend im Zusammenhang mit künstlichen neuronalen Netzen verwendet und tauchte in diesem Kontext erstmals im Jahr 2000 auf, in der Veröffentlichung *Multi-Valued and Universal Binary Neurons: Theory, Learning and Applications* von Igor Aizenberg und Kollegen.[18][19][20]

Zwischen 2009 und 2012 gewannen die rekurrenten bzw. tiefen vorwärtsgerichteten neuronalen Netze der Forschungsgruppe von Jürgen Schmidhuber am Schweizer KI Labor IDSIA eine Serie von acht internationalen Wettbewerben in den Bereichen Mustererkennung und maschinelles Lernen.[21] Insbesondere gewannen ihre rekurrenten LSTM-Netze[22][23] drei Wettbewerbe zur verbundenen Handschrifterkennung bei der *2009 Intl. Conf. on Document Analysis and Recognition (ICDAR)* ohne eingebautes A-priori-Wissen über die drei verschiedenen zu lernenden Sprachen. Die LSTM-Netze erlernten gleichzeitige Segmentierung und Erkennung. Dies waren die ersten internationalen Wettbewerbe, die durch Deep Learning[24] oder durch rekurrente Netze gewonnen wurden.

Die jüngsten Erfolge von Deep Learning Methoden, wie der Go-Turniergewinn des Programmes AlphaGo gegen die weltbesten menschlichen Spieler, gründen sich neben der gestiegenen Verarbeitungsgeschwindigkeit der Hardware auf den Einsatz von Deep Learning zum Training des in AlphaGo verwendeten neuronalen Netzes.[25] Gleiches gilt für die seit 2020 gelungene Vorhersage von Protein-Faltungen.[26] Diese Netze nutzen künstlich erzeugte Neuronen (Perzeptron), um Muster zu erkennen.

Für Beiträge zu neuronalen Netzwerken und Deep Learning erhielten Yann LeCun, Yoshua Bengio und Geoffrey Hinton 2018 den Turing Award.[27]

Komplexität und Grenzen der Erklärbarkeit [Bearbeiten | Quelltext bearbeiten]

Tiefe neuronale Netze können eine Komplexität von bis zu hundert Millionen einzelnen Parametern und zehn Milliarden Rechenoperationen pro Eingangsdatum aufweisen. Die Interpretierbarkeit der Parameter und Erklärbarkeit des Zustandekommens der Ergebnisse ist hier nur noch eingeschränkt möglich und erfordert den Einsatz spezieller Techniken, die unter Explainable Artificial Intelligence zusammengefasst werden. Ein weiterer Ansatz ist die Verwendung von Methoden aus der Physik von Vielteilchensysteme (Statistische Physik).[28] Eine weitere Begleiterscheinung des Deep Learning ist die Anfälligkeit für Falschberechnungen, die durch subtile, bei zum Beispiel Bildern für Menschen nicht sichtbare, Manipulationen der Eingabesignale ausgelöst werden können. Dieses Phänomen wird unter Adversarial Examples zusammengefasst.[29]

Es gibt zwei Konzepte zu Grenzen und Erklärbarkeit: „Opake KI" und „transparente KI". Ersteres, Opake KI, beinhaltet neuronale Netze, Deep Learning, genetische Algorithmen etc. Bei beiden Konzepten ist gemeinsam, dass die Logik dahinter inkl. der Vorhersagen und Entscheidungen nicht einfach ausgedrückt werden kann. Transparente KI kann hingegen die Entscheidungen erklären und für den Menschen verständlich machen. Die Entscheidung für oder gegen eines der beiden Konzepte endet schnell in ethischen und moralischen Vorstellungen. Ohne einen T-Switch (für vertrauensvoll oder transparent) ist Opake KI kaum zu kontrollieren. Je mehr Daten in einem Opaque-System verwendet werden (Big Data), umso eher wird die künstliche Intelligenz ein Wertesystem für sich selbst entwickeln, welche die Menschen nicht mehr verstehen können. Ein T-Switch ermöglicht es, bewusste Entscheidungen durch die Opake KI zu erlauben und an welcher Stelle man auf Transparenz bestehen sollte.

Alles in allem betrachtet hat Opake KI einen Nachteil: Im Jahr 2016 hat Microsoft ein Experiment durchgeführt. In weniger als in 24 Stunden wurden sehr negative Ergebnisse bei der Veröffentlichung von einem Twitter Chatbot namens Tay erzielt.

Transparente KI hingegen unterstützt eine exakte Erklärung. So kann immer die Entscheidung nachvollzogen werden.

Programmbibliotheken [Bearbeiten | Quelltext bearbeiten]

Neben der meist in Schulungsbeispielen zum Verständnis der internen Struktur vorgestellten Möglichkeit, ein neuronales Netz komplett eigenhändig zu programmieren, gibt es eine Reihe von Softwarebibliotheken,[30] häufig Open Source, lauffähig auf meist mehreren Betriebssystemplattformen, die in gängigen Programmiersprachen wie zum Beispiel C, C++, Java oder Python geschrieben sind. Einige dieser Programmbibliotheken unterstützen GPUs oder TPUs zur Rechenbeschleunigung oder stellen Tutorials zur Benutzung dieser Bibliotheken bereit. Mit ONNX können Modelle zwischen einigen dieser Tools ausgetauscht werden.

- TensorFlow (Python, JavaScript, C++, Java, Go, Swift) von Google
- Keras (Python, ab Version 1.4.0 auch in der TensorFlow-API enthalten)[31] – populäres Framework (2018) neben Tensorflow.[32]
- Caffe vom Berkeley Vision and Learning Center (BVLC)
- PyTorch (Python), entwickelt vom Facebook-Forschungsteam für künstliche Intelligenz
- Torch (C, Lua)[33] (Community) und das darauf aufbauende Facebook-Framework Torchnet[34]
- Microsoft Cognitive Toolkit (C++)[35]
- PaddlePaddle (Python) vom Suchmaschinenhersteller Baidu[36][37]
- OpenNN (C++), implementiert ein künstliches neuronales Netz.
- Theano (Python) von der Université de Montréal[38]
- Deeplearning4j (Java) von Skymind
- MXNet von der Apache Software Foundation[39]

Literatur [Bearbeiten | Quelltext bearbeiten]

- François Chollet: *Deep Learning mit Python und Keras: Das Praxis-Handbuch vom Entwickler der Keras-Bibliothek*. mitp, 2018, ISBN 978-3-95845-838-3.
- Ian Goodfellow, Yoshua Bengio, Aaron Courville: *Deep Learning: Adaptive Computation and Machine Learning*. MIT Press, Cambridge USA 2016, ISBN 978-0-262-03561-3.
- Jürgen Schmidhuber: *Deep learning in neural networks: An overview*. In: *Neural Networks*, 61, 2015, S. 85, arxiv:1404.7828 [cs.NE].
- Rob F. Walker: *Artifical Intelligence in Business: Balancing Risk and Reward*, 2017, Seite 1–23.

Weblinks [Bearbeiten | Quelltext bearbeiten]

- **Commons: Deep learning** – Sammlung von Bildern, Videos und Audiodateien
- Luis Serrano: *A friendly introduction to Deep Learning and Neural Networks* auf YouTube, 26. Dezember 2016, abgerufen am 7. November 2018.
- Deep Learning – Einführung. Übersichtsartikel zum Thema Deep Learning
- *Thema: Deep Learning.* heise online
- *Deep Learning: Wie Maschinen lernen.* spektrum.de – Übersetzung des Artikels *The learning machines*. In: *Nature*, 505, S. 146–148, 2014

Einzelnachweise [Bearbeiten | Quelltext bearbeiten]

1. ↑ Hannes Schulz, Sven Behnke: *Deep Learning: Layer-Wise Learning of Feature Hierarchies*. In: *KI - Künstliche Intelligenz*. Band 26, Nr. 4, November 2012, ISSN 0933-1875, S. 357–363, doi:10.1007/s13218-012-0198-z (springer.com [abgerufen am 13. Januar 2020]).
2. ↑ Herbert Bruderer: *Erfindung des Computers, Elektronenrechner, Entwicklungen in Deutschland, England und der Schweiz*. In: *Meilensteine der Rechentechnik*. 2., völlig neu bearbeitete und stark erweiterte Auflage. Band 2. De Gruyter, 2018, ISBN 978-3-11-060261-6, *Wörterverzeichnis zur Technikgeschichte*, S. 408 (eingeschränkte Vorschau in der Google-Buchsuche [abgerufen am 23. November 2019]).
3. ↑ a b c d e f Ian Goodfellow, Yoshua Bengio, Aaron Courville: *Deep Learning*. MIT Press, abgerufen am 19. Februar 2017 (englisch).
4. ↑ a b David Kriesel: *Ein kleiner Überblick über Neuronale Netze*. (PDF; 6,1 MB) In: *dkriesel.com*. 2005, abgerufen am 21. Januar 2019.
5. ↑ Li Deng, Dong Yu: *Deep Learning: Methods and Applications*. 1. Mai 2014 (microsoft.com [abgerufen am 17. Mai 2020]).
6. ↑ Mathias Schuh, Lukas Friehoff: *Deep Learning im Rechtsmarkt*. In: LEGAL REVOLUTIONary. 23. Mai 2019, abgerufen am 23. August 2021.
7. ↑ a b Michael Nielsen: *Neural Networks and Deep Learning*. Determination Press, abgerufen am 21. Februar 2017 (englisch).
8. ↑ Li Deng, Dong Yu: *Deep Learning: Methods and Applications*. In: Microsoft Research (Hrsg.): *Foundations and Trends in Signal Processing Volume 7 Issues 3-4*. 1. Mai 2014, ISSN 1932-8346 (englisch, microsoft.com [abgerufen am 22. Februar 2017]).
9. ↑ Ivakhnenko, A. G. and Lapa, V. G. (1965). Cybernetic Predicting Devices. CCM Information Corpo-ration.
10. ↑ Jürgen Schmidhuber: *Deep learning in neural networks: An overview*. In: *Neural Networks*. 61, 2015, S. 85, arxiv:1404.7828 [cs.NE].
11. ↑ Jürgen Schmidhuber: *Critique of Paper by "Deep Learning Conspiracy" (Nature 521 p 436)*. In: *people.idsia.ch*. Juni 2015, abgerufen am 12. April 2019.
12. ↑ Karl Steinbuch: *Die Lernmatrix*. In: *Kybernetik, 1(1):36-45*. 1961.
13. ↑ Kunihiko Fukushima: *Neocognitron: A Self-organizing Neural Network Model for a Mechanism of Pattern Recognition Unaffected by Shift in Position*. In: *Biological Cybernetics*. 36. Jahrgang, Nr. 4, 1980, S. 193–202, doi:10.1007/BF00344251 (princeton.edu [PDF]).
14. ↑ Alex Waibel: *Phoneme Recognition Using Time-Delay Neural Networks*. Meeting of the Institute of Electrical, Information and Communication Engineers (IEICE), Tokyo, Japan, 1987. 1987 (englisch).
15. ↑ Yann LeCun et al.: *Backpropagation Applied to Handwritten Zip Code Recognition, Neural Computation, 1, pp. 541–551, 1989*, abgerufen am 11. Mai 2020.
16. ↑ Sven Behnke: *Hierarchical Neural Networks for Image Interpretation* (= *Lecture Notes in Computer Science*. Band 2766). Springer Berlin Heidelberg, Berlin, Heidelberg 2003, ISBN 978-3-540-40722-5, doi:10.1007/b11963 (springer.com [abgerufen am 28. November 2020]).
17. ↑ Rina Dechter: *Learning while searching in constraint-satisfaction problems.* (PDF; 531 kB) In: *fmdb.cs.ucla.edu*. University of California, Computer Science Department, Cognitive Systems Laboratory, 1985, archiviert vom Original (nicht mehr online verfügbar) am 16. September 2021; abgerufen am 9. Juli 2020 (englisch). Info: Der Archivlink wurde automatisch eingesetzt und noch nicht geprüft. Bitte prüfe Original- und Archivlink gemäß Anleitung und entferne dann diesen Hinweis.
18. ↑ Horváth & Partners: *„Künstliche Intelligenz wird alles ändern"* (ab 0:11:30) auf YouTube, 9. Mai 2016, abgerufen am 6. November 2018 (Vortrag von Jürgen Schmidhuber).
19. ↑ Jürgen Schmidhuber: *Deep Learning since 1991*. In: *people.idsia.ch*. 31. März 2017, abgerufen am 1. Dezember 2018.
20. ↑ Igor Aizenberg, Naum N. Aizenberg, Joos P.L. Vandewalle: *Multi-Valued and Universal Binary Neurons: Theory, Learning and Applications*. Springer Science & Business Media, 2013, ISBN 978-1-4757-3115-6 (google.com).
21. ↑ 2012 Kurzweil AI Interview (Memento vom 31. August 2018 im *Internet Archive*) mit Jürgen Schmidhuber zu den acht Wettbewerben, die sein Deep Learning Team zwischen 2009 und 2012 gewann
22. ↑ Alex Graves, Jürgen Schmidhuber: *Offline Handwriting Recognition with Multidimensional Recurrent Neural Networks*. In: Yoshua Bengio, Dale Schuurmans, John Lafferty, Chris K. I. Williams, Aron Culotta (Hrsg.): *Advances in Neural Information Processing Systems 22 (NIPS'22), December 7th–10th, 2009, Vancouver, BC*. Neural Information Processing Systems (NIPS) Foundation, 2009, S. 545–552
23. ↑ A. Graves, M. Liwicki, S. Fernandez, R. Bertolami, H. Bunke, J. Schmidhuber: *A Novel Connectionist System for Improved Unconstrained Handwriting Recognition*. IEEE Transactions on Pattern Analysis and Machine Intelligence, Band 31, Nr. 5, 2009.
24. ↑ Y. Bengio: *Learning Deep Architectures for AI* (Memento vom 21. März 2014 im *Internet Archive*) Now Publishers, 2009.
25. ↑ Demis Hassabis: *AlphaGo: using machine learning to master the ancient game of Go*. In: *blog.google*. Google, 27. Januar 2016, abgerufen am 16. Juli 2017 (englisch).
26. ↑ *AlphaFold: a solution to a 50-year-old grand challenge in biology.* In: *deepmind.com*. 30. November 2020, abgerufen am 19. Dezember 2020 (englisch).
27. ↑ Alexander Neumann: *Deep Learning: Turing Award für Yoshua Bengio, Geoffrey Hinton und Yann LeCun – heise online.* In: *heise.de*. 28. März 2019, abgerufen am 29. März 2019.
28. ↑ John Hertz, Anders Krogh, Richard G. Palmer: *Introduction to the theory of neural computation* (= *Santa Fé Institute studies in the sciences of complexity Lecture notes*. Nr. 1). Addison-Wesley, Reading, Mass. 1999, ISBN 978-0-201-51560-2.
29. ↑ Leilani H. Gilpin, David Bau, Ben Z. Yuan, Ayesha Bajwa, Michael Specter: *Explaining Explanations: An Overview of Interpretability of Machine Learning*. In: *2018 IEEE 5th International Conference on Data Science and Advanced Analytics (DSAA)*. IEEE, Turin 2018, ISBN 978-1-5386-5090-5, S. 80–89, doi:10.1109/DSAA.2018.00018 (ieee.org [abgerufen am 8. Dezember 2019]).
30. ↑ Dan Clark: *Top 16 Open Source Deep Learning Libraries and Platforms.* KDnuggets, April 2018, abgerufen am 8. Januar 2019 (englisch).
31. ↑ *Keras Documentation.* In: *Keras: Deep Learning library for Theano and TensorFlow*. Abgerufen am 6. März 2017 (englisch).
32. ↑ *Why use Keras?* In: *keras.io*. Abgerufen am 8. Januar 2020 (englisch): „Keras is also a favorite among deep learning researchers, coming in #2 in terms of mentions in scientific papers uploaded to the preprint server arXiv.org. Keras has also been adopted by researchers at large scientific organizations, in particular CERN and NASA."
33. ↑ *Torch | Scientific computing for LuaJIT.* Abgerufen am 17. Februar 2017 (englisch).
34. ↑ Rainald Menge-Sonnentag: *Maschinelles Lernen: Facebook veröffentlicht Open-Source-Framework für Torch.* In: *heise.de*. 24. Juni 2016, abgerufen am 17. Februar 2017.
35. ↑ *The Microsoft Cognitive Toolkit.* In: *microsoft.com*. Abgerufen am 11. August 2017 (amerikanisches Englisch).
36. ↑ *Startseite.* In: *paddlepaddle.org*. Abgerufen am 17. Februar 2017 (englisch).
37. ↑ Alexander Neumann: *Baidu gibt Deep-Learning-System als Open Source frei.* In: *heise.de*. 2. September 2016, abgerufen am 17. Februar 2017.
38. ↑ *Theano.* In: *deeplearning.net*. Archiviert vom Original (nicht mehr online verfügbar) am 8. November 2020; abgerufen am 20. September 2019 (englisch). Info: Der Archivlink wurde automatisch eingesetzt und noch nicht geprüft. Bitte prüfe Original- und Archivlink gemäß Anleitung und entferne dann diesen Hinweis.
39. ↑ *Apache MXNet (Incubating) – A flexible and efficient library for deep learning.* In: *mxnet.apache.org*. Abgerufen am 5. September 2019 (englisch).

Kategorien: Neuroinformatik | Deep Learning

Der Text ist unter der Lizenz „Creative-Commons Namensnennung – Weitergabe unter gleichen Bedingungen" verfügbar; Informationen zu den Urhebern und zum Lizenzstatus eingebundener Mediendateien (etwa Bilder oder Videos) können im Regelfall durch Anklicken dieser abgerufen werden. Möglicherweise unterliegen die Inhalte jeweils zusätzlichen Bedingungen. Durch die Nutzung dieser Website erklären Sie sich mit den Nutzungsbedingungen und der Datenschutzrichtlinie einverstanden.
Wikipedia® ist eine eingetragene Marke der Wikimedia Foundation Inc.

Get citation