Web-Veröffentlichung, Nov. 2025;
Update 20. 12. 2025
https://homepages.uni-paderborn.de/winkler/Winkler--Neuronale-Netze.pdf

Hartmut Winkler

Neuronale Netze

Was sollte man als Medien- und Kulturwissenschaftler/in über das technische Funktionieren von KI-Systemen wissen?

1. Intro

Über die Frage, was KI will, kann und tut, wird in der Presse, im Alltag und in den Kulturwissenschaften häufig, und fast immer in den gleichen Stereotypen gesprochen. Es geht darum, ‚welche Auswirkungen die KI auf die Gesellschaft hat', auf die Arbeitswelt, auf die Wissenschaft und die Kreativen; es geht um ‚Big data', ‚Maschinelles Lernen', Sprachmodelle und die Generierung von Bildern, um die Frage, ob und wann die Modelle ‚den Menschen' mittel- oder langfristig hinter sich lassen; dass sie Mittel der Täuschung und der politischen Beeinflussung sind, parteiisch, oder nicht verlässlich, weil sie ‚halluzinieren'…

All das wäre interessanter und weniger stereotyp, wenn die Beteiligten mehr Detailwissen im Feld der KI hätten. Sich das zu erarbeiten aber ist nicht trivial, denn die Modelle, um die es hier geht, sind auf spezifische Weise opak.[1] Im Folgenden will ich den Versuch machen, zumindest den wichtigsten Begriffen und Funktionsweisen etwas näher zu kommen.

Mein Interesse war, die grundlegenden Strukturen zunächst selbst zu verstehen. Und da ich zwar in ferner Vergangenheit selbst Programme geschrieben habe, über die Technik der KI aber nicht viel wusste, habe ich – vor allem anderen – das eigene Verständnis zum Maßstab gewählt. Das entsprechende Material zusammenzutragen war nicht einfach. Einschlägige Lehrbücher der Informatik waren die erste Wahl, allerdings setzen sie Kenntnisse voraus, über die man als Medien- und Kulturwissenschaftler schlicht nicht verfügt; sie gehen schnell ins Detail und liefern mathematische Formeln, bevor auch nur die groben Konturen geklärt sind. Ganz schlimm sind die gedruckten Einführungen, die ich auf der Suche nach Informationen gesichtet habe; entweder wollen sie – eine Art Motivationsliteratur – Wirtschaftsleuten die Angst nehmen, KI einzusetzen, oder sie greifen, anders als der Titel verspricht, willkürlich Einzelaspekte heraus oder gehen vorschnell zu Beispielen über. An einem strukturellen Verständnis scheinen sie desinteressiert.

---

[1] Softwareentwicklung ist generell eine Sache von Spezialisten, und für Außenstehende schwer zu durchschauen. Für die KI gilt dies in besonderem Maß: Die Firmen schotten sich aus Konkurrenzgründen sorgfältig ab und legen die wichtigen Algorithmen nicht offen; der Umfang der Trainingsdaten übersteigt jedes menschliche Maß, und da die Systeme nicht im engeren Sinne programmiert, sondern ‚trainiert' werden, können – Stichwort: ‚hidden layers' – nicht einmal diejenigen, die die Algorithmen schreiben, vollständig kontrollieren, was im Inneren der Neuronalen Netze tatsächlich vor sich geht,

Ich habe deshalb vor allem populärwissenschaftliche Websites herangezogen. Auch diese sind nicht unproblematisch, weil sie häufig von Firmen stammen, die Software oder Dienstleistungen im Themenfeld anbieten, und insofern immer einen PR-Aspekt haben. Gleichzeitig haben sie den Vorteil, dass sie sich an ein Laien-Publikum wenden und versuchen, die Sachverhalte so einfach wie möglich zu fassen. Das hat sich bei der Recherche als hilfreich erwiesen. Ebenfalls hilfreich und ein sehr guter Einstieg in die Materie sind die Lehrfilme, die das Computer History Museum (CHM) der Stadt Mountain View (Cal.) unter dem Channel-Namen ‚3Blue1Brown' veröffentlicht hat.[2]

Am brauchbarsten aber waren – und das ist durchaus kurios und scheint mir nur bei einem technischen Thema einigermaßen möglich – Informationen, die verschiedene KIs, und vor allem ChatGPT, selbst zum Thema bereitgestellt haben. Die Pointe hier ist, dass man immer wieder ***zurückfragen*** kann, und das verändert den Lernprozess wirklich. Ich habe – in ironischer Erinnerung an die Platonischen Lehrdialoge – mit den tatsächlich schafsgeduldigen Systemen tagelange Verhandlungen geführt; und Schritt für Schritt eine Unzahl von Fragen für mich klären können, über die die populärwissenschaftlichen Texte leichtfüßig, generös oder schlampig hinweggehen. Mein Projekt war, die Einzelinformationen zu sammeln, sie in eine für Kulturwissenschaftler verständliche Sprache zu übersetzen und in eine nachvollziehbare Folge zu bringen.

Was also kann man aus diesen Darstellungen lernen?

2. KI-Modelle

2.1 Was man als ‚Künstliche Intelligenz' begreift, hat sich im Verlauf der Geschichte immer wieder drastisch verändert.[3] Hatte man in den 1950er Jahren den Computer insgesamt als ein „Elektronengehirn" verstanden, beginnt mit dem ‚General Problem Solver' 1957 die Geschichte der KI im engeren Sinne; man entwarf ‚Expertensysteme', versprach, binnen weniger Jahre die natürliche Sprache zu modellieren, oder simulierte ein psychiatrisches Erstinterview;

> „John McCarthy schlug 1958 vor, das gesamte menschliche Wissen in eine homogene, formale Darstellungsform, die Prädikatenlogik […] zu bringen." [4]

Und von Beginn an gab es den Paralleldiskurs einer ***KI-Kritik***, die die Grundannahmen, die Rhetorik und die überzogenen Ansprüche immer wieder gnadenlos demontierte.[5] Die KI-Entwicklung verlief entsprechend in Wellen: Immer wieder nahmen die Protagonisten Argumente der Kritiker auf und bestimmten neu, was im nächsten Zyklus als ‚KI' gelten sollte.

---

[2] Siehe z. B.: [3Blue1Brown]: Large Language Models for the Curious Beginner [2024]; https://www.youtube.com/watch?v=LPZh9BOjkQs;
- die Serie umfasst acht weitere Videos zur Einführung in die Materie, diese allerdings sind schon wesentlich komplizierter: [3Blue1Brown]: But What is a Neural Network? Deep Learning Chapter 1 [2017]; https://www.youtube.com/watch?v=aircAruvnKk; usf.
die Seite des Museums selbst ist: https://computerhistory.org/.

[3] Vgl. – informativ und brauchbar: – Wikipedia (dt.): Geschichte der künstlichen Intelligenz; https://de.wikipedia.org/wiki/Geschichte_der_k%C3%BCnstlichen_Intelligenz.

[4] Ebd.

[5] Ich möchte beispielhaft zwei Autoren der 70er Jahre nennen: - Dreyfus, Hubert L.: Was Computer nicht können. Die Grenzen künstlicher Intelligenz [1972]. Frankfurt a. M.: Athenäum 1989; - Weizenbaum, Joseph: Die Macht der Computer und die Ohnmacht der Vernunft [1976]. Frankfurt a. M.: Suhrkamp 1978. Viele weitere Beispiele nennt Wikipedia [Künstliche Intelligenz; Abschnitt: Kritik an der KI-Forschung]; https://de.wikipedia.org/wiki/K%C3%BCnstliche_Intelligenz#Kritik_an_der_KI-Forschung.

2.2 Die gegenwärtigen KI-Systeme sind durch folgende Merkmale gekennzeichnet:
- Technische Grundlage sind ‚Neuronale Netze', die große Ressourcen beanspruchen und nur auf sehr leistungsfähigen Großrechnern lauffähig sind.
- Man ist erstmals in der Lage, nun auch ***‚schwach strukturierte Daten'*** zu verarbeiten. ‚Schwach strukturiert' sind z. B. natürlichsprachliche Texte (an deren Analyse herkömmliche Programme weitgehend scheitern), Bilder und Bewegtbilder, sowie Audiodaten. Traditionelle Datenbanken dagegen sind ‚strukturiert'.
- Die Systeme akzeptieren natürlichsprachliche Texte, Bild, Bewegtbild und Audio als ***Eingabe***, und können diese, und z. B. Programmcode, auch als ***Ausgabe*** selbst synthetisieren/generieren.
- Die Systeme stützen sich auf eine überwältigend große Datenbasis, die sie sich einverleiben, um daraus eine neue Art von strukturierten Informationen zu ziehen.
- Einige der Systeme arbeiten multimedial, können also Texte, Bilder, Audio und Bewegtbild bruchlos parallel und im gleichen System verarbeiten.[6]
- Die Systeme können Muster erkennen und klassifizieren/kategorisieren/abstrahieren;
- Viele der Systeme arbeiten in ***Realzeit,*** sodass sie unmittelbar dialogfähig sind.

- Und schließlich beanspruchen die KI-Systeme, Zugang nun auch zu ‚Sinn' und ‚Bedeutung' gefunden zu haben. Dies vor allem anderen – denke ich – hat in der Öffentlichkeit wie in den Medien- und Kulturwissenschaften einen Schock ausgelöst.[7]

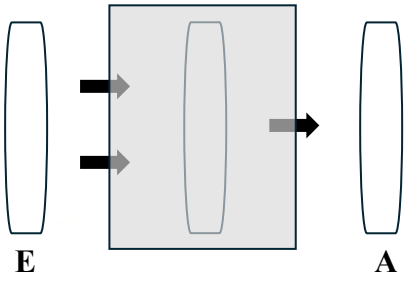
2.3 KI-Systeme sind – wie alle IT-Systeme – ***Modelle***, die eine eigene Logik, eigene Gesetze und eigene Grenzen haben, und weder mit dem Material, von dem sie ausgehen, noch mit dem, was sie modellieren, einfach zusammenfallen.

2.4 KI-Systeme werden nicht im klassischen Sinn programmiert, sondern – Stichwort: „Machine Learning" – angelernt oder „trainiert".

Wenden wir uns nun also dem technischen Funktionieren zu.

3. Neuronale Netze – Grundanordnung, Architektur

3.1 KI-Systeme arbeiten mit sogenannten ‚neuronalen Netzen', und diese bestehen aus mehreren Schichten.



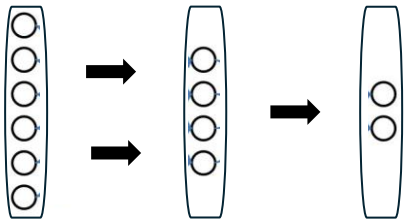
Die Eingabe- und die Ausgabeschicht liegen offen zutage, für die dazwischenliegenden Schichten gilt dies nicht, sie werden deshalb ‚hidden layers' genannt.

---

[6] "[Current AI-] models have emerged multimodal architectures that can process text, images, audio, and videos." (Rothman, Denis: Transformers for Natural Language Processing and Computer Vision. Birmingham/Mumbai: Packt 2024, S. 1 (Erg. H. W.))

[7] So entschieden die Medien- und Kulturwissenschaften ‚Sinn' und ‚Bedeutung', Interpretation und Hermeneutik in der Vergangenheit in Zweifel gezogen haben, in so klarer Weise ist man an sie gebunden. (Vgl. die poststrukturalistischen Texte zur Dekonstruktion, oder auch: Sontag, Susan: Against Interpretation. In: dies.: Against Interpretation and Other Essays. NY: Dell 1966, S. 13-23).

3.2 Die Schichten stellen jeweils eine bestimmte Zahl von ‚Neuronen' bereit; deren Anzahl ist unterschiedlich; im Fall der gegenwärtig führenden Modelle ist sie sehr groß; wie groß aber tatsächlich, veröffentlichen die Betreiber nicht.[8]

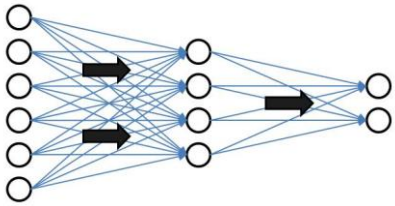


3.3 „[Schichten und Neuronen] sind Softwarekomponenten, keine Hardwarekomponenten. Sie sind mathematische Modelle, die in Software implementiert werden, um bestimmte Berechnungen durchzuführen." [9]

3.4 ***Die Neuronen sind untereinander verbunden***, sie sind ***Knoten in einem Netzwerk***; und nur aus diesem Grund spricht man – analog zu den Nervenzellen – metaphorisch von ‚Neuronen' und ‚Netzen'.[10]

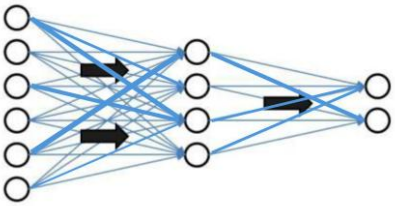
3.5 Diese Verbindungen allerdings gibt es nicht in der gleichen Schicht, sondern ausschließlich ***zwischen*** den Schichten.

3.6 Jedes einzelne Neuron einer Schicht ist ***mit allen Neuronen der nächsten Schicht*** verbunden.[11]



Bis zu diesem Punkt ist das Netzwerk ‚neutral' und hat keine Struktur.

3.7 Allerdings – und das ist der wohl wichtigste Punkt – ***wird jeder Verbindung ein spezifisches ‚Gewicht' zugewiesen***; das Gewicht bestimmt, welchen Stellenwert die Verbindung relativ zu den anderen in der gleichen Schicht hat, wie stark das Neuron der vorangegangenen Schicht auf das Neuron in der aktuellen Schicht einwirkt.



---

[8] „Die genaue Anzahl der Neuronen in der Eingabeschicht von ChatGPT […] ist nicht öffentlich bekannt, da die Architektur von OpenAI nicht vollständig offengelegt wurde." – so viel zum Thema ***Open***AI!

[9] ChatGPT auf die Frage: Sind die Neuronen eines Neuronalen Netzwerks Hardware- oder Softwarekomponenten? - 3. 11. 25 (Erg. H. W.).

[10] „In einem neuronalen Netzwerk (wie es in der Künstlichen Intelligenz verwendet wird) sind die Neuronen und Verbindungen ***metaphorische Konzepte***, die von der Funktionsweise des menschlichen Gehirns inspiriert wurden. Die Begriffe ‚Neuronen' und ‚Verbindungen' sind aber nicht ganz gleichbedeutend mit den biologischen Äquivalenten, sondern eher abstrahierte Modelle." (ChatGPT auf die Frage: Wofür stehen die Neuronen eines Neuronalen Netzwerks? Und wofür stehen deren Verbindungen? – 1. 11. 25 (Hervorh. H. W.)).

[11] Grafik: Dörn, Sebastian: Neuronale Netze. https://sebastiandoern.de/neuronale-netze/, 31. 10. 25.

3.8 Die gewichteten Verbindungen werden ***‚Parameter'*** genannt;

> „Parameter sind die ‚einstellbaren Werte' eines Modells, die während des Trainingsprozesses feinjustiert werden, um das Modell an die Trainingsdaten anzupassen. ***Sie sind die grundlegenden Bausteine eines Modells***, die dessen Fähigkeit bestimmen, Muster und Beziehungen in den Daten zu lernen." [12]

3.9 Die Gesamtzahl dieser Parameter kann sehr groß sein;

> „GPT-3 zum Beispiel hat ***175 Milliarden Parameter***, die es in die Lage versetzen, eine Vielzahl von Aufgaben wie Textgenerierung, Sprachverständnis und Übersetzung zu meistern." [13]

3.10 Erst dadurch, dass die Gewichte ***unterschiedlich*** sind, bekommt das neuronale Netzwerk eine jeweils spezifische ***Struktur.***

4. Ein Neuronales Netz ist eine Maschine

4.1 Eigentlich aber ist ein Neuronales Netzwerk nichts Statisches, keine Architektur, sondern eine ***Maschine***. Und interessant wird eine Maschine erst dann, wenn sie ***läuft***, wenn das Programm ***ausgeführt*** und wenn Daten ***prozessiert*** werden. Eigentlich also geht es nicht um die Struktur, sondern um das Laufzeitverhalten.

4.2 Zunächst verhält sich ein Neuronales Netzwerk wie jedes Programm: Es werden Daten eingegeben, dann werden sie verarbeitet/prozessiert, und dann werden veränderte Daten ausgegeben.

4.3 Allerdings muss man unterscheiden:
- in der Trainingsphase geht es um die Zuweisung der Gewichte selbst; dies geschieht in einer großen Zahl von Iterationen, in denen die Gewichte immer wieder angepasst werden. („Maschinelles Lernen").
- Ist das Training abgeschlossen, liegen die ‚Gewichte' fest, und das Neuronale Netzwerk kann wie ein Programm (oder eben eine Maschine) zur Verarbeitung von neuen Daten eingesetzt werden.

4.4 (Eine Besonderheit der Neuronalen Netze allerdings ist, dass in bestimmten Fällen diese Trennung nicht gilt: Hier lässt man zu, dass auch die Nutzerdaten, die verarbeitet werden, die Gewichte weiter verändern. Die Struktur der Maschine also wird im Prozess ihrer Nutzung immer weiter aus- und umgebaut).

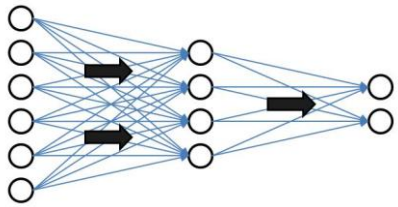
---

[12] ChatGPT auf die Frage: Was ist die Bedeutung von ‚Parametern', ‚Dimensionen' und ‚Gewichten' in ChatGPT? – 1. 11. 25 (Hervorh. H. W).

[13] Ebd. (Hervorh. H. W).
Eine Einführung nennt folgende Vergleichsdaten für das menschliche Gehirn:
„Biologisches Vorbild: Gehirn mit $10^{10} – 10^{11}$ [= 10 bis 100 Mrd.] Neuronen, 20 Typen,
- Axon bis zu 1 m lang […]
- Jeweils Verbindungen (Synapsen) zu 10 – 100.000 Neuronen
- ***Insgesamt 10^{14}*** [= 100.000 Mrd. = 100.000.000.000.000] ***Synapsen***
- Schaltzeit 10^{-3} Sekunden [= 0,001 Sek.] (Computer 10^{-10} [= 0,0000000001 Sek.])
- Erfassung einer Szene [?]: 10^{-1} Sekunden, d. h. ca. 100 Verarbeitungsschritte
- hohe Parallelität, Lernfähigkeit."
(Burkhard, Hans Dieter; Döhring, Daniel: Einführung in die KI – Neuronale Netze. https://www.informatik.hu-berlin.de/de/forschung/gebiete/ki/lehre/ws0405/vlki-skript/eki0405NeuronaleNetze.pdf (Erg. u. Hervorh. H. W.)).

5. Der Fluss der Daten

5.1 Den Fluss der Daten durch ein neuronales Netzwerk nennt man ‚Propagation'.[14]



5.2 „In einem künstlichen neuronalen Netzwerk ***stellt ein Neuron einen Rechenschritt oder eine Verarbeitungseinheit dar***. […] [Jedes Neuron einer Schicht erhält] Eingaben ***von allen Neuronen der vorherigen Schicht und gibt seine Ausgabe an alle Neuronen der nächsten Schicht weiter.***" [15]

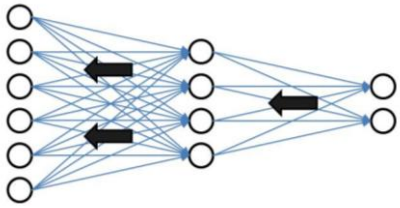
5.3 Ein Neuron ist eine kleine Rechenmaschine, die drei Typen von Werten miteinander verrechnet:
- die ***Eingabewerte***, die von ***allen*** Neuronen der vorangegangenen Schicht kommen,
- die ***Gewichte***, die diesen eingehenden Verbindungen jeweils zugeordnet sind,
- und die sogenannte ***‚Aktivierungsfunktion'***. Sie bestimmt, ab welchem Schwellenwert das Neuron der Schicht ‚feuert', also auf die Neuronen der nächsten Schicht einwirkt.

> „Ein Neuron in einem künstlichen Netzwerk ist also wie ***eine kleine Entscheidungseinheit***, die eine gewichtete Summe von Eingaben empfängt und daraufhin eine Ausgabe berechnet." [16]

5.4 (Die Neuronen der Eingabeschicht allerdings starten anders: Sie bekommen einen bestimmten Wert zugewiesen; in der Bildanalyse kann dies z. B. der Grauwert eines einzelnen Pixels sein).

5.5 Und in der Phase, in der das Neuronale Netzwerk trainiert wird, kehrt man den Datenfluss zeitweise um: Abhängig vom Ergebnis geht man ‚rückwärts', um die Gewichte anzupassen und zu verändern;

> „während des Trainingsprozesses wird das Netzwerk durch den ***Backpropagation-Algorithmus*** optimiert." [17]



Soweit zur grundlegenden Architektur der KI-Systeme.

---

[14] Grafik: Dörn, Neuronale Netze, a. a. O.

[15] ChatGPT auf die Fragen: Sind die Neuronen eines Neuronalen Netzes mit anderen Neuronen der gleichen Schicht verbunden, oder nur mit Neuronen der nächsten Schicht? – 1. 11. 25, sowie: Wofür stehen die Neuronen eines neuronalen Netzwerks? Und wofür stehen deren Verbindungen? – 31. 10. 25 (Erg. u. Hervorh. H. W).

[16] ChatGPT auf die Frage: Wofür stehen die Neuronen eines Neuronalen Netzwerks? Und wofür stehen deren Verbindungen? – 31. 10. 25 (Hervorh. H. W).

[17] ChatGPT auf die Frage: Wie werden Vektoren in den Neurons Neuronaler Netzwerke implementiert? - 3. 11. 25 (Hervorh. H. W); Grafik: Dörn, Neuronale Netze, a. a. O.

6. Wie werden Daten in Neuronalen Netzwerken repräsentiert ? Was ist ein ‚Vektor' ?

6.1 Alle Daten, die in einem neuronalen Netzwerk verarbeitet werden, müssen die Form von ***Vektoren*** haben. Vektoren beschreiben die Objekte, indem sie ihnen für bestimmte ***Merkmale*** numerische Werte zuweisen; eine Firmen-Website veranschaulicht dies so:

> „Zum besseren Verständnis nehmen Sie Breiten- und Längengrad als Beispiel. Diese beiden Dimensionen – Nord-Süd bzw. Ost-West – können die Lage eines beliebigen Ortes auf der Erde angeben. Die Stadt Vancouver […] kann durch die Breiten- und Längengradkoordinaten {{49°15'40'' N, 123°06'50'' W}} dargestellt werden. ***Diese Liste von zwei Werten ist ein einfacher Vektor***." [18]

Und die Autoren schildern – im Vorgriff – gleich eine Verwendung:

> „Angenommen Sie versuchen, eine Stadt zu finden, die ganz in der Nähe von Vancouver liegt. Ein Mensch würde einfach auf eine Karte schauen, während ein maschinelles Lernmodell stattdessen den Breiten- und Längengrad (oder Vektor) betrachten und einen Ort mit einem ähnlichen Breiten- und Längengrad finden könnte. Die Stadt Burnaby liegt bei {{49°16'00'' N, 122°58'00'' W}} – sehr nahe an {{49°15'40'' N, 123°06'50'' W}}. Daher kann das Modell korrekt schlussfolgern, dass Burnaby in der Nähe von Vancouver liegt." [19]

Das Beispiel ist plastisch, weil man sich Länge und Breite als zwei ‚Dimensionen' leicht vorstellen kann. Die Merkmale, die in den ‚Vektor' eingehen, aber sind auf die zwei Dimensionen des geographischen Raums nicht beschränkt; und in der Vektortheorie werden grundsätzlich alle Merkmale ‚Dimensionen' genannt. Die Erklärung fährt deshalb fort:

> „Stellen Sie sich nun vor, dass Sie eine Stadt suchen, die nicht nur in der Nähe von Vancouver liegt, sondern auch eine ähnliche Größe hat. Fügen wir diesem Modell der Standorte eine dritte ‚Dimension' zum Breiten- und Längengrad hinzu: die Bevölkerungszahl. Die Bevölkerungszahl kann zum Vektor jeder Stadt hinzugefügt werden, und kann wie eine Z-Achse behandelt werden, wobei der Breiten- und Längengrad die Y- und X-Achsen darstellen.
> Der Vektor für Vancouver lautet nun {{49°15'40'' N, 123°06'50'' W, **662.248**}. Mit dieser dritten Dimension liegt Burnaby nicht mehr besonders nahe an Vancouver, da seine Einwohnerzahl nur 249.125 beträgt. Das Modell könnte stattdessen die Stadt Seattle im US-Bundesstaat Washington, finden, die einen Vektor von {{47°36'35'' N, 122°19'59'' W, **749,256**} hat." [20]

6.2 Die Vektor-Theorie nun – das ist ihre Besonderheit – macht auch bei der dritten Dimension keineswegs halt. Vielmehr kann nach dem gleichen Muster die Zahl der ‚Dimensionen' beliebig erhöht werden, je nachdem, wie viele Merkmale man den einzelnen Daten-Objekten zuordnen will. Die Vektortheorie hat keinerlei Probleme, n-dimensionale Räume zu konstruieren. Unsere Alltags-Vorstellung kommt hier an Grenzen; an die sinnliche Anschauung gebunden, können wir uns mühelos die drei Dimensionen des physischen Raumes vorstellen, und dazu vielleicht noch als Viertes die Achse der Zeit. Die Mathematik aber ist an den physischen Raum nicht gebunden und kennt vergleichbare Grenzen

---

[18] Ohne Autor [Cloudflare]: Was sind Einbettungen beim maschinellen Lernen? https://www.cloudflare.com/de-de/learning/ai/what-are-embeddings/ (Hervorh. H. W.; ebenso habe ich typographische Fehler in der Bezeichnung der Längen- und Breitengrade korrigiert).

[19] Ebd.

[20] Ebd. (Hervorh. H. W.).

nicht; mit jeder zusätzlichen Dimension würde sich die Zeichenkette des Vektors einfach um ein Glied verlängern.

> „Anstelle von zwei, drei oder fünf Dimensionen ist eine TV-Serie in einem Modell für maschinelles Lernen ein Punkt entlang von vielleicht hundert oder tausend Dimensionen – je nachdem, wie viele das Modell berücksichtigen möchte.“ [21]

6.3 Auf Papier kann man Vektoren entweder als Spalten- oder als Zeilenvektoren schreiben:

$$\boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ \vdots \\ x_n \end{bmatrix} \qquad \boldsymbol{x}^T = \begin{bmatrix} x_1 & x_2 & x_3 & .. & .. & x_n \end{bmatrix}$$

6.4 Wichtig nun ist, dass Vektoren ausschließlich numerische Werte enthalten;[22] und für jede Dimension muss feststehen, wie sie skaliert ist. ***Natürlichsprachliche Texte oder Bilder müssen deshalb erst in Zahlen umgeformt werden***. Die Ausgangsobjekte werden mathematisch/numerisch durch jeweils ein Merkmalsbündel, den Vektor, repräsentiert. Dieser fasst alle Merkmale zusammen, die das Einzelobjekt beschreiben.

6.5 Dies ist die Voraussetzung dafür, dass man die Werte, die verschiedene Datenobjekte/ Items auf einer einzelnen Dimension haben, ***vergleichen*** kann. Datenobjekte miteinander vergleichbar zu machen, ist der eigentliche Zweck, auf den die Vektorisierung abzielt. Allerdings meist nicht entlang nur einer einzelnen Dimension; denn wirklich interessant wird es erst, wenn, wie im Beispiel schon angedeutet, mehrere, viele oder alle Dimensionen in den Vergleich einbezogen werden. Man hat Algorithmen entwickelt, die exakt das leisten.[23] Und nur in diesem Fall würde man von ‚Nähe‘ oder ‚Ähnlichkeit‘ sprechen.

> “Vector embeddings work by representing features or objects as points in a multi-dimensional vector space, where the relative positions of these points represent meaningful relationships between the features or objects. As mentioned, they capture semantic relationships between features or objects by placing similar items closer together in the vector space.
> Then, distances between vectors are used to quantify relationships between features or objects. ***Common distance metrics*** […] ***measure how ‘close’ or how ‘far’ vectors are to each other in the multidimensional space.***” [24]

> “[Vektoren], die nahe beieinander liegen – so wie Seattle und Vancouver [im Beispiel oben] nahe beieinander liegende Breiten- und Längengrade ***und*** vergleichbare Bevölkerungszahlen haben – ***können als ähnlich angesehen werden.***“ [25]

---

[21] Ebd.

[22] …wenn die Längen- und Breitengrade im Beispiel die Buchstaben N, S, W und O, also auch nicht-numerische Werte, enthalten, ist das ein Sonderfall; und die Werte sind auch hier klar skaliert…

[23] Die Seite nennt “Euclidean distance, cosine similarity, and Manhattan distances”…

[24] O. A. [Tigerdata]: A Beginner’s Guide to Vector Embeddings. (Hervorh. H. W.); https://www.tigerdata.com/blog/a-beginners-guide-to-vector-embeddings.

[25] Cloudflare, a. a. O. (Erg. u. Hervorh. H. W.).

Im Universum der KI ist ‚Ähnlichkeit' definiert als geringe *Distanz*. Die Distanz aber – das ist wichtig – ist kein tatsächlich *räumlicher* Abstand, weil der Vektorraum, anders als der physische Raum, nicht nur drei, sondern ‚n' Dimensionen hat.

6.6 Und es scheint mir noch eine weitere Klarstellung nötig, denn einige der genannten Punkte würden auch für eine konventionelle, relationale Datenbank gelten; und auch der zitierte Beispielfall der kanadischen Städte wäre mit einer konventionellen Datenbank gut zu realisieren. Warum also ‚Vektoren'?

> „Herkömmliche Datenbanken reichen nicht aus, um die Komplexität von Vektordaten zu bewältigen. Dies erschwert deren Analyse und die Gewinnung aussagekräftiger Erkenntnisse. Hier kommen Vektordatenbanken ins Spiel – spezialisierte Datenbanken, die für die Verarbeitung von Vektoren entwickelt wurden und Vektoreinbettungen effizient speichern und abrufen können." [26]

Warum aber ist das so? Warum reichen herkömmliche Datenbanken nicht aus?[27] Hier kann man, nach dem was ich lese, verschiedene Gründe nennen: So geht es [1.] um besondere Typen von Daten; und vor allem um Daten, die – es wurde oben gesagt – so, wie sie gegeben sind, wenig Struktur haben. Beispiel seien natürlichsprachliche Texte. Diese kann man zwar abspeichern und auf Stichworte durchsuchen; sobald man aber zwei Texte miteinander vergleichen und auf inhaltliche ‚Ähnlichkeit' prüfen will, wird man auf große Probleme stoßen. Und das Gleiche gilt auch für digitale Bilder oder Videos, die, weil sie zunächst nichts als eine sehr große Menge von Pixeln bieten, kaum zu vergleichen oder nach ‚Inhalten' zu durchsuchen sind. Offenbar also lassen schwach strukturierte Daten [2.] nur bestimmte Auswertungen zu; und man setzt Vektordatenbanken ein, um, gestützt auf anders geartete Algorithmen, andere Arten von Auswertungen möglich zu machen.

Und schwach strukturierte Daten müssen [3.] ***aufgearbeitet*** werden. Vektordatenbanken also sind ein Zwischenprodukt auf dem Weg zu bestimmten Folgeoperationen.

Weiter unterscheiden sich [4.] auch die Herkunft der Daten und der Prozess, in dem die Daten entstehen.

Und schließlich bietet sich eine Vektordatenbank [5.] offenbar an, wenn es darum geht, Querverbindungen zwischen den einzelnen Dimensionen, und Zusammenhänge ***innerhalb*** des Datenbestandes, zu zeigen. In konventionellen Datenbanken sind die Einzelvariablen zunächst unabhängig voneinander; man kann man zwar Korrelationen rechnen,[28] das aber ist eher ein Sonderfall; im Fall von Vektordaten ist dies anders: Hier zielt bereits die Datenstruktur darauf ab, solche Zusammenhänge zu zeigen.

Beginnen wir mit [3.], der Aufbereitung der Daten.

7. Vektorisierung, Aufbereitung der Daten, ‚Einbettung'

7.1 Die Erstellung von Vektor-Datenbanken wird – rätselhafte Wortwahl der Informatik – ‚Einbettung' (Embedding) genannt.

---

[26] Tigerdata, a. a. O.

[27] Zu den Unterschieden zwischen herkömmlichen Datenbanken und Vektordatenbanken siehe z.B.: www.hosteurope.de/blog/vektordatenbanken-vs-herkoemmliche-datenbanken-was-sind-die-unterschiede/

[28] Beispiel wäre eine Frage wie: Gibt es einen Zusammenhang zwischen Einkommen und Geschlecht…

7.2 Vorbereitung, Tokenisierung

> „Der erste Schritt besteht darin, die Rohdaten zu sammeln, die Sie verarbeiten möchten. Dies können Text, Audio, Bilder, Zeitreihendaten oder jede andere Art strukturierter oder unstrukturierter Daten sein.
> Anschließend müssen Sie die Daten vorverarbeiten, um sie zu bereinigen und für die Analyse geeignet zu machen. Je nach Art Ihrer Daten kann dies Aufgaben wie ***Tokenisierung*** (bei Textdaten), Rauschentfernung, Größenanpassung von Bildern, Normalisierung, Skalierung und andere Datenbereinigungsvorgänge umfassen.“ [29]

> „Als nächstes müssen Sie die Daten in Blöcke aufteilen. Je nach Art der Daten müssen Sie möglicherweise Text in Sätze oder Wörter gliedern (wenn Sie Textdaten haben), Bilder in Segmente [?] oder Patches unterteilen […] oder Zeitreihen in Intervalle oder Fenster […].“ [30]

> „Bei modernen Modellen wie BERT oder GPT kann das eine Wort-Tokenisierung oder eine Subwort-Tokenisierung sein. Zum Beispiel wird das Wort ‚unhappily‘ in zwei Token unterteilt: ‚un‘, ‚happily‘.“ [31]

7.3 „Einbettung“

> „Nachdem Sie die Daten vorverarbeitet und in geeignete Blöcke aufgeteilt haben, besteht der nächste Schritt darin, ***jeden Block in eine Vektordarstellung umzuwandeln. Dieser Vorgang wird als Einbettung bezeichnet.***“ [32]

Dabei werden auch die alphanumerischen Daten natürlichsprachlicher Texte in eine numerische Form übersetzt:

> „Einbettungen wandeln [die schwach strukturierten Ausgangsdaten, also z. B. die Wörter des natürlichsprachlichen Textes] in komplexe mathematische Repräsentationen um, die inhärente Eigenschaften und Beziehungen zwischen realen Daten erfassen.“ [33]

Am Ende des ‚Embedding‘ steht die Vektordatenbank; sie ist das Produkt der Bearbeitung der ursprünglichen Daten. Und gleichzeitig ist sie Startpunkt dafür, dass nun all diejenigen Auswertungsalgorithmen laufen können, die eine Vektordatenbank zur Voraussetzung haben.

Und wichtig: Einmal erstellt, ***ersetzen*** Vektoren das Ausgangsmaterial; es wird bei allen Auswertungen in der Folge nicht mehr mit den Ausgangsdaten, sondern nur noch mit den Vektoren gerechnet.

Der Rückbezug zu den Ausgangsdaten (z. B. zu den Wörtern des natürlichsprachlichen Textes) allerdings muss wiederhergestellt werden können, wenn die Ergebnisse wieder in Textform dargestellt werden sollen.

---

[29] Tigerdata, a. a. O. (Hervorh. H. W.).

[30] Ebd. (Erg. H. W.).

[31] ChatGPT auf die Frage: In der Antwort ‚Diese Modelle arbeiten nicht direkt mit den Rohdaten (also den Worten als Zeichenketten), sondern sie enthalten eingebettete Schichten, die die Wörter automatisch in Vektoren umwandeln.‘ verstehe ich das ‚sondern‘ nicht. Was wird denn eingegeben, wenn nicht die Zeichenkette? – 4. 11. 25.

[32] Ebd. (Hervorh. H. W.).

[33] O. A. [Amazon/AWS]: Embeddings in Machine Learning (Erg. H. W.; das Original spricht irreführend von „reale Objekten“). https://aws.amazon.com/de/what-is/embeddings-in-machine-learning/

8. Wie werden *Vektoren* in Neuronalen Netzen repräsentiert?

8.1 Die Umwandlung der einzelnen Items/Tokens (z. B. die einzelnen Worte oder Bildelemente) in Vektoren (das ‚Embedding') kann durch spezielle Programme geschehen, die man dem Neuronalen Netz vorschaltet, oder bei den gegenwärtigen, großen sogenannten ‚Transformern' übernimmt dies das Neuronale Netzwerk selbst.

8.2 Hierbei nimmt jedes Item

> „in der Eingabeschicht ***so viele Neuronen in Anspruch, wie die Dimension des Vektors***, der es repräsentiert." „Wenn also jedes Wort durch einen Vektor der Dimension 300 […] dargestellt wird, dann benötigt das Wort 300 Neuronen in der Eingabeschicht". „Ein einzelnes Neuron repräsentiert in der Eingabeschicht in der Regel ***eine Dimension*** eines Vektors für ein einzelnes Wort oder Item." [34]

8.4 Stellt die Eingabeschicht mehr Neuronen bereit, als ein einzelnes Item Dimensionen hat, kann die Eingabeschicht mehrere Items gleichzeitig aufnehmen (also z. B. mehrere Wörter eines Satzes, ganze Sätze oder Sequenzen).

8.5 Die ***Gewichte*** werden für jede Schicht des Neuronalen Netzwerks in einer sogenannten ***Matrix*** gespeichert. Eine Matrix ist eine Art Tabelle.

8.6 Vektoren und Matrizen müssen komplett und vollständig im Arbeitsspeicher (RAM) zur Verfügung stehen.

> „Bei sehr großen Netzwerken wie GPT-3 (mit Milliarden von Parametern) kann der Speicherbedarf extrem hoch werden." [35]

9. Noch einmal zurück zum einzelnen Neuron

9.1 Ein einzelnes Neuron, hatte ich oben gesagt, ist eine kleine Rechenmaschine, die aus einer Anzahl von Eingabewerten einen Ausgabewert ermittelt.

> „In jedem Neuron wird eine ***gewichtete Summe*** der Eingabewerte errechnet. Dies ist eine ***Matrix-Vektor-Multiplikation***, die die Eingaben mit den Gewichten kombiniert." [36]

Die Eingaben kommen von der vorangegangenen Schicht, die Gewichte werden der ‚Matrix' entnommen.

9.2

> „Diese gewichtete Summe wird dann an die Aktivierungsfunktion übergeben, die entscheidet, ob und wie stark das Neuron ‚feuert' oder aktiv wird. Die Aktivierungsfunktion ist eine mathematische Formel; das Resultat ist der ***Ausgabewert*** des Neurons." [37]

---

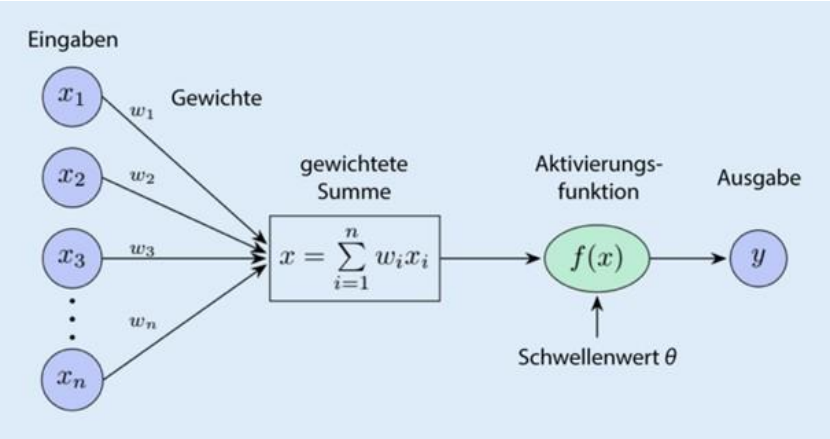
[34] ChatGPT auf die Frage: Wie viele Neuronen der Eingabeschicht nimmt ein Wort in Anspruch? – 03. 11. 25 (Hervorh. H. W.); dies gilt nur, wenn das NN bereits vektorisierte Daten als Eingabe erhält; wandelt es die Daten selbst in Vektoren um, bilden die Rohdaten die erste Schicht, die Dimensionen des ersten Vektors die zweite.

[35] ChatGPT auf die Frage: Ist das gesamte Neuronale Netz im Arbeitsspeicher geladen? – 02. 11. 25.

[36] ChatGPT auf die Frage: Wie werden Vektoren in den Neurons neuronaler Netzwerke implementiert? – 2. 11. 25 (Hervorh. u. Erg. H. W.).

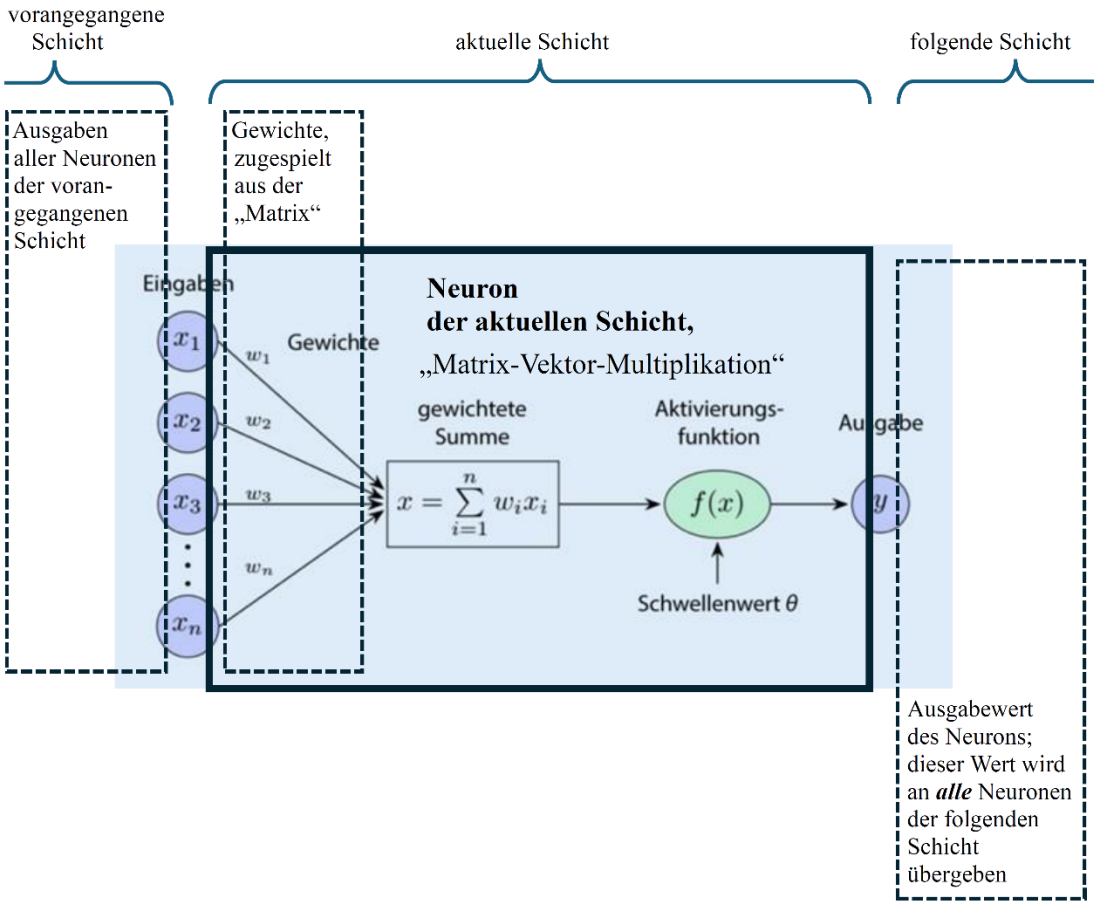
[37] Ebd. (Hervorh. H. W.).

9.3 Weberskirch liefert hierzu eine Skizze, die zunächst etwas kryptisch erscheint, zum Verständnis aber durchaus beitragen kann:



38

Wesentlich deutlicher nämlich wird die Sache, wenn man die Schichten etwas expliziter markiert und sich klarmacht, ***dass die Ein- und Ausgaben tatsächlich Schnittstellen (oder Übergabepunkte) zwischen den Schichten sind***: [39]



[38] Grafik: Weberskirch, Carolin et al.: Einsatz neuronaler Netze in der Notaufnahme. https://link.springer.com/article/10.1007/s10049-021-00974-x

[39] Eigene Darstellung auf Basis von Weberskirch.

9.4 Die Ausgabe liefert jeweils ***einen*** Wert für die Vektoren der folgenden Schicht; die Werte aller Neuronen werden zusammengetragen, und der Vektor wird aus diesen Einzelwerten zusammengesetzt.

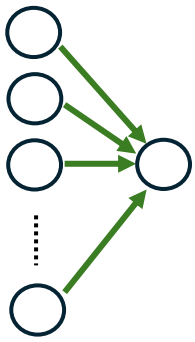
> „Wenn man eine ganze Schicht von Neuronen hat, wird diese Berechnung für alle Neuronen parallel durchgeführt, sodass der Ausgang der Schicht ebenfalls ein Vektor ist.“ [40]

9.5 „In einem tiefen neuronalen Netzwerk (mit mehreren Schichten) wird dieser Vorgang iterativ wiederholt, wobei jeder Vektor die Eingaben für die nächste Schicht darstellt. Der Eingabevektor wird direkt als Input an das Netzwerk gegeben. Jede der verborgenen Schichten verarbeitet die Eingaben durch Matrix-Vektor-Multiplikation und Aktivierungsfunktionen. Der letzte Vektor repräsentiert die Ausgabe des Netzwerks (z. B. eine Vorhersage oder Klassifikation).“

10. Vernetzungslogik

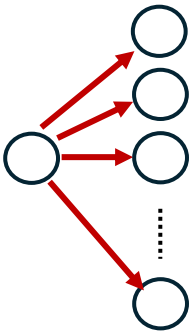
10.1 Und jetzt wird der eigentliche Grund deutlich, warum man für die Konstruktion Neuronaler Netze ***Vektoren*** verwendet: Vektoren sind als Elemente für den Bau von Netzen geeignet, weil sie einen ‚Knoten‘ (**x**) mit bestimmten ‚Kanten‘ (x_1, x_2, x_3 … x_n) verbinden:

$$\boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ \vdots \\ x_n \end{bmatrix}$$



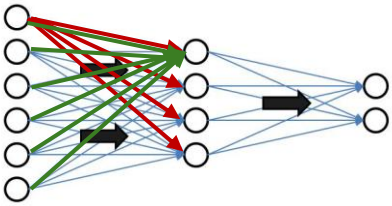
Ein Vektor bestimmt ein Item **‚x‘** mit Hilfe einer Anzahl von Merkmalen (x_1, x_2, x_3 … x_n); jedes Neuron erhält als Eingabe die Ausgabewerte ***aller Neuronen*** der vorangegangenen Schicht. ***Ein Vektor hat insofern immer eine many-to-one-Struktur***.

10.2 Gleichzeitig übergibt das Neuron seinen Ausgabewert an alle Neuronen der Folgeschicht; hier ist die Vernetzungslogik ***one-to-many***;



[40] Ebd. (Hervorh. H. W.).

10.3 Es sind insofern zwei verschiedene Vernetzungslogiken, die sich in der Konstruktion der Neuronalen Netze überlagern.



10.4 Beide aber haben nicht den gleichen Rang: Da das Neuron mehrere ***unterschiedliche*** Eingabewerte erhält, an die Neuronen der Folgeschicht aber nur ***einen einzigen*** Wert ausgibt, ist die many-to-one-Logik auf der Eingabeseite dominant. ***Und diese Relation ist es, die als Vektor gespeichert wird.***

11. Training, maschinelles Lernen

11.1 Neuronale Netze werden ‚trainiert', und dabei – auch das wurde schon gesagt – geht es vor allem um die Festlegung der ‚Gewichte'.

> „Technisch gesehen sind Einbettungen Vektoren, ***die von maschinellen Lernmodellen*** [ML] ***erstellt werden***, um aussagekräftige Daten über jedes Objekt zu erfassen. [...] Einbettung ist der Prozess der Erstellung von Vektoren mithilfe von Deep Learning." [41]

11.2 Der erste Fall ist der ***Neuaufbau*** eines Neuronalen Netzwerks. Hier wird das Netzwerk zunächst mit beliebigen oder zufälligen Gewichten versehen, und dann werden Trainingsdaten zugespielt,

> „[d. h. die] Techniker versorgen das neuronale Netzwerk mit einigen vektorisierten Proben, die manuell vorbereitet wurden." [42]

In diesen Daten ist das gewünschte Ergebnis (also z. B. einer Mustererkennung oder der Zuordnung zu einer Kategorie) bereits enthalten; man spricht deshalb von bereits ‚gelabelten' Daten.

In einer zweiten Phase greift man nur noch dann korrigierend ein, wenn das Netzwerk falsche Ergebnisse liefert, auch hier also müssen Menschen dem Netzwerk sagen, ob das Ergebnis der Berechnung (also z. B. einer Mustererkennung oder der Zuordnung zu einer Kategorie) korrekt ist oder nicht.

> „[Während dieser Phase] des Trainings wird das Embedding selbst durch die Backpropagation und das Update der Modellparameter kontinuierlich angepasst." [43]

Das Netzwerk verändert die Gewichte immer wieder, um Schritt für Schritt bessere Ergebnisse zu erzielen.

Bis, in der dritten Phase, das Modell schließlich selbstständig läuft und die Aufgabe ohne weiteren Eingriff des Menschen erfüllt, oder nur noch so viele Fehler macht, wie die Macher für vertretbar halten. Für dieses Training sind eine sehr große Menge von Daten und sehr viele Lernzyklen nötig.

---

[41] Cloudflare, a. a. O. (Erg. u. Hervorh. H. W.).

[42] Amazon/AWS, Embeddings, a. a. O.

[43] ChatGPT auf die Frage: Gehört das Training des Modells zum Embedding dazu, oder ist das Embedding abgeschlossen, wenn das Training beginnt? – 4. 11. 25.

11.3 Anders ist der Fall, wenn ein bereits bestehendes Neuronales Netzwerk (also z. B. ein Sprachmodell) für die Verarbeitung jeweils aktueller Daten genutzt werden soll. (Ein Beispiel hierfür wären Übersetzungsprogramme, die strikt zwischen Anbietern und Usern trennen).

11.4 Und auch der dritte Fall wurde schon genannt: Hier ist zwar ein trainiertes Modell bereits vorhanden, dieses aber soll durch die jeweils aktuellen Daten, die die User liefern, schrittweise ausgebaut und weiter verbessert werden. Hier lässt man die ‚Backpropagation' weiter zu.

12. Supervisied und unsupervised learning

Wenn Menschen das Neuronale Netzwerk trainieren, wäre dies ‚supervised learning'. Daneben aber gibt es – möglicherweise noch interessanter – auch ein

> „‚unsupervised learning', bei dem [um noch einmal beim Beispiel natürlichsprachlicher Texte zu bleiben] nur die Texte vorhanden sind und etwa Kategorien auf der Grundlage von Ähnlichkeiten gefunden werden. Topic-Modeling ist eine Technik des ‚unsupervised learning' und ist in der Lage, verborgene semantische Strukturen in einem Dokument zu erfassen." [44]

Ein Beispiel wären Cluster-Analysen, die Gruppierungen oder Kategorien im Material suchen:

> „Cluster analysis is a branch of machine learning ***that groups the data that has not been labelled, classified or categorized***. Instead of responding to feedback, cluster analysis identifies commonalities in the data and reacts based on the presence or absence of such commonalities in each new piece of data." [45]

Hier kann die Maschine auch solche Cluster im Material finden, die vorher nicht bekannt waren, die überraschend sind.

13. Merkmalsextraktion, ‚Feature Learning'

In bestimmten Fällen also man gibt dem Programm die ‚Merkmale', um die es gehen soll, vor. In anderen Fällen überlässt man es der Maschine, im Ausgangsmaterial jene ‚Dimensionen'/Features/Merkmale zu finden, die dann den Vektor bilden. Dieser Vorgang wird ‚Feature Learning' genannt.

> „Die Merkmale, die die Dimensionen des Vektors bilden, werden durch ein Verfahren namens ***Feature Learning*** extrahiert. In diesem Prozess lernt das Modell, die relevanten Informationen aus den Eingabedaten zu extrahieren und diese in eine niedrigdimensionale Repräsentation (also den Vektor) zu kodieren. Beispielsweise wird ein Bild von einem Hund so umgewandelt, dass das Modell lernt, Merkmale wie Form, Textur und Farbe zu extrahieren, ohne dass explizit jemand im Voraus sagt, was wichtig ist." [46]

---

[44] https://intrafind.com/de/blog/natural-language-processing-best-practice (Erg. H. W.);
die zitierte Seite ist ganz ausgezeichnet; sie gibt Schritt für Schritt viele Informationen, die anderswo kaum zu finden sind.

[45] Wikipedia (Engl.): Unsupervised Learning.
https://en.wikipedia.org/wiki/Unsupervised_learning; (Hervorh. H. W.).

[46] ChatGPT auf die Frage: Woher kommen bei KI-Modellen die ‚Merkmale', die die Dimensionen des Vektors bilden? - 9. 11. 25; ChatGPT ist stilistisch einigermaßen unsensibel und neigt dazu, Wortwiederholungen zu produzieren.

14. Mit Vektoren rechnen

14.1 Der große Vorteil von Vektoren ist, dass man mit ihnen ***rechnen*** kann. Man kann Vektoren addieren, subtrahieren und multiplizieren. Hierbei werden jeweils alle Dimensionen des Vektors einbezogen. Und das Neuronale Netzwerk wird mit bestimmten Algorithmen verknüpft, die festlegen, was mit den Daten von Schicht zu Schicht geschehen soll.

14.2 Diese Algorithmen sind unterschiedlich, je nachdem, welchen Zweck das Neuronale Netzwerk erfüllen soll; „die Vorgabe für ein Modell ist immer eine bestimmte Aufgabe, die es zu lösen gilt.“ [47]

14.3 Eine häufig verwendete Art der Berechnung wurde schon erwähnt: Man kann Vektoren vergleichen, um ***Ähnlichkeiten*** zu finden.

15. Reduzierung der Anzahl der Dimensionen

15.1 Die Materialmenge, mit der KI-Systeme starten, ist unvorstellbar riesig, und überschaubar groß ist anfangs auch die Anzahl der Parameter, der Dimensionen des Vektors.

> “Transformers can perform self-supervised learning ***on billions of records*** of raw unlabeled data ***with billions of parameters***. From these billion-parameter models have emerged multimodal architectures that can process text, images, audio, and videos.“ [48]

15.2 Aus diesem Grund ist eine der Hauptaufgaben des Netzwerks, die Anzahl der Dimensionen zu ***reduzieren***. Auch diese Reduktion wird zur ‚Einbettung‘ gerechnet;[49] und ***erst diese Reduktion ist es, die den Vektor zu einem schlanken, eleganten und mächtigen Werkzeug macht.***

> „Hochdimensionale Daten“, schreibt ‚Tigerdata‘, „werden in einen niedrigerdimensionalen Raum abgebildet (eine Art ‚verlustbehaftete Komprimierung‘), der strukturelle oder semantische Beziehungen innerhalb der Daten erfasst. Dadurch können

---

[47] ChatGPT auf die Frage: Wie kann ich mir das ‚Feature Learning‘ konkret vorstellen? Was ist die Vorgabe? – 11. 11. 25.

[48] Rothman, Transformers, a. a. O., S. 1 (Hervorh. H. W.).
ChatGPT sagt bescheidener: “Im Fall von ChatGPT, speziell bei der Repräsentation von Text oder Sprache in einem mathematischen Modell, wird ein Vektor oft als hochdimensionale Darstellung eines Textes oder eines Worts verwendet. Diese Vektoren werden in einem hochdimensionalen Raum erzeugt, der oft zwischen 100 und 1000 Dimensionen oder mehr umfassen kann, abhängig vom spezifischen Modell.“
(ChatGPT auf die Frage: Wie viele Dimensionen hat ein Vektor in ChatGPT? – 29. 10. 25)
„In modernen GPT-Modellen, wie GPT-3 und GPT-4, handelt es sich bei den Vektoren, die in den internen Schichten des Modells verwendet werden, um hochdimensionale Vektoren. Diese Dimensionen sind Teil der sogenannten ‚Hidden Layers‘ des Modells, und die Anzahl dieser Dimensionen ist eng mit der Anzahl der Parameter des Modells verbunden.
GPT-3, das 175 Milliarden Parameter hat, verwendet 12.288 Dimensionen in seinen versteckten Schichten. Obwohl GPT-4 nicht offiziell in Bezug auf die genaue Anzahl der Parameter und Dimensionen veröffentlicht wurde, wird angenommen, dass es eine ähnliche Architektur wie GPT-3 hat oder sogar noch größere Dimensionen verwendet (vielleicht 16.000 oder mehr Dimensionen), da es größere Modelle und feinere Repräsentationen von Sprache ermöglicht.“ (Ebd.)
Warum ist die Anzahl der Dimensionen der Vektoren nicht gleich der Anzahl der Parameter?
„Die Anzahl der Parameter hängt von der Anzahl der Attention-Köpfe [‚Attention‘ ist ein Spezifikum der gegenwärtigen, sehr leistungsfähigen ‚Transformer‘-Modelle] und der Dimension der Eingabedaten ab, aber sie ist nicht direkt mit der Dimension der Vektoren gleich, weil das Modell zusätzliche Gewichtsmatrizen für die Transformation und Kombination der verschiedenen Teile des Modells benötigt.“ (ChatGPT – 12. 11. 25 (Erg. H. W.)).

[49] Ebd.

wichtige Informationen erhalten bleiben und gleichzeitig der Rechenaufwand bei der Verarbeitung großer Datensätze reduziert werden. Außerdem hilft sie, Muster und Beziehungen in Daten aufzudecken, die im ursprünglichen Raum möglicherweise nicht erkennbar waren.“ [50]

15.3 Die Reduzierung wird im Durchgang durch die verschiedenen Schichten auf verschiedene Schritte verteilt:

„In einem tiefen neuronalen Netzwerk […] gibt es viele Schichten von ‚Neuronen‘. Jede Schicht lernt etwas anderes. Die unteren [, früheren] Schichten lernen oft einfache Merkmale, während die höheren Schichten komplexere Muster und Abstraktionen erkennen. […]
- (z. B. bei einem Bild): In den [früheren] Schichten eines neuronalen Netzwerks werden grundlegende Merkmale wie Kanten, Linien, Farben und Texturen erkannt. Das sind einfache, niedrigdimensionale [?] Merkmale, die noch nicht viel über den Inhalt des Bildes aussagen, aber grundlegende Informationen bereitstellen.
- Die mittleren Schichten beginnen, aus diesen einfachen Merkmalen komplexere Formen zu kombinieren, wie etwa Ecken oder Konturen von Objekten.
- In den [späteren,] höheren Schichten kombinieren sich diese Merkmale zu komplexeren Konzepten. Zum Beispiel könnte ein Modell, das auf Bildern von Tieren trainiert wird, in den höheren Schichten lernen, was ein Hund oder eine Katze ist, basierend auf den zuvor erlernten Kanten, Formen und Texturen.“ [51]

(Die Darstellung kann verwirren, weil sie auf den ersten Blick einen Weg vom Einfachen zum Komplizierten zu beschreiben scheint, was der Aussage, dass das Netz eine Reduzierung von Komplexität leistet, zunächst widerspricht. Der Widerspruch, denke ich, löst sich auf, wenn man sich klar macht, dass am Anfang immer eine unübersehbare Menge von Material und am Ende z. B. eine sehr begrenzte Zahl von Konzepten (Hund oder Katze) steht. Diese Konzepte sind ***in sich*** komplex, ihre Zahl aber ist gering, weshalb die Vektoren der letzten Schichten nur noch auf relativ wenige Knoten zielen).

15.4 Wie aber geschieht die genannte Reduzierung konkret? Eines der Verfahren, die hierbei eingesetzt werden, ist die Hauptkomponentenanalyse:

„Die Hauptkomponentenanalyse (PCA) ist ein statistisches Verfahren zur Dimensionsreduktion, das eine große Anzahl korrelierter Variablen in eine kleinere Anzahl unabhängiger Hauptkomponenten zusammenfasst.“ [52]

„PCA wird häufig zur Datenvorverarbeitung bei der Verwendung mit Algorithmen des maschinellen Lernens genutzt. ***Es kann die informativsten Merkmale aus großen Datensätzen extrahieren und gleichzeitig die relevantesten Informationen aus dem ursprünglichen Datensatz beibehalten***. Dies reduziert die Komplexität des Modells, da das Hinzufügen jedes neuen Merkmals die Leistung des Modells beeinträchtigt, was allgemein auch als ‚Fluch der Dimensionalität‘ bezeichnet wird.“ [53]

---

[50] Tigerdata, a. a. O.

[51] ChatGPT auf die Frage: Wie kann ich mir das ‚Feature Learning‘ konkret vorstellen? Was ist die Vorgabe?

[52] Google KI [Hauptkomponentenanalyse]; neben der PCA nennt die Website die Singulärwertdekomposition, Word2Vec und BERT, wobei es sich bei letzteren offenbar bereits um proprietäre Softwarepakete handelt.

[53] O. A. [IBM]: Was ist die Principal Component Analysis (PCA)?
https://www.ibm.com/de-de/think/topics/principal-component-analysis (Hervorh. H. W.).
Wikipedia erläutert die Algorithmen näher (https://de.wikipedia.org/wiki/Hauptkomponentenanalyse), mathematisch wenig geschult aber verstehe ich diese Erläuterung nicht.

Die Hauptkomponentenanalyse, schreibt der Autor weiter, ‚projiziert' einen hochdimensionalen Datensatz (einen sehr umfangreichen Vektor) in einen kleineren Merkmalsraum.

15.5 Wichtig ist die Nebenbemerkung im Zitierten, dass man mit den ‚Hauptkomponenten' möglichst ‚unabhängige' Variablen zu finden versucht.[54] Andere Variablen, die ohnehin hoch korrelieren, also ‚abhängig' voneinander sind,[55] können auf einen gemeinsamen Vektor zusammengezogen werden. Hierdurch wird im Datenbestand Redundanz reduziert.

16. Vereinfachung, Reduzierung von Komplexität

16.1 Die Tatsache, dass man nicht mit allen aufgefundenen Dimensionen arbeitet, sondern deren Anzahl gezielt reduziert, ist eine eigene Überlegung wert. Auf der einen Seite handelt es sich, wie gesagt wurde, um eine ‚verlustbehaftete Komprimierung' und es gehen Informationen, die in den ursprünglichen Daten vorhanden waren, verloren. ***Auf der anderen Seite aber steigert man gerade dadurch die Aussagekraft***. Und dies scheint mir der zentrale Punkt: Durch die Vektorisierung (das Embedding) und die Reduzierung der Dimensionen ***generiert man Struktur***.

Die wenig strukturierten, ursprünglichen Daten werden ‚abgemagert' zu einer Struktur. Repräsentiert durch den Vektor sind die Daten nicht mehr „schwach strukturiert".

16.2 Und genau hier ergibt sich eine Parallele, die die Kulturwissenschaft mehr als hellhörig machen müsste: Exakt die gleiche Funktion nämlich haben – außerhalb der Computer, im Verlauf der Mediengeschichte – schon eine große Anzahl anderer, traditioneller Kulturtechniken gehabt.[56] Und mehr noch: Gestützt auf die Schematheorie kann man zeigen, dass eine der Hauptfunktionen der Medien und Zeichensysteme darin besteht, die beängstigende Komplexität, die sich in der Welt vorfindet, einigermaßen in den Griff zu bekommen.[57]

An dieser Stelle also schließen die Neuronalen Netze an historische Medien an, die – jedes für sich – völlig anders funktioniert haben; dies zu zeigen, aber würde den hier gesteckten Rahmen sprengen. Ich werde den Bezug deshalb an anderer Stelle weiter verfolgen.[58]

17. Statistik und Wahrscheinlichkeit

17.1 „Die Wahrscheinlichkeitsrechnung spielt eine zentrale Rolle in der Künstlichen Intelligenz, da sie eine mathematische Grundlage für das Verständnis und die Modellierung von Unsicherheit bietet. In vielen KI-Anwendungen müssen Systeme Entscheidungen treffen oder Vorhersagen machen, obwohl sie nicht über vollständige oder fehlerfreie Informationen verfügen." [59]

---

[54] FN 51.

[55] …In Fall eines Autos etwa Motorgröße, Fahrzeuggewicht und Benzinverbrauch…

[56] Über diese Eigenschaft der Medien habe ich an anderer Stelle geschrieben: W., H.: Kulturtechniken zur Reduzierung von Komplexität. /Winkler--Kulturtechniken-zur-Reduzierung-von-Komplexitaet--2022.pdf.

[57] Vgl. W., H.: Ähnlichkeit. Berlin: Kadmos 2021, Kap. 4, 9-11, 16; d. h. S. 41-58, 133-194, 273-290; das Buch ist online verfügbar: …Winkler--Aehnlichkeit.pdf.

[58] W., H.: Ähnlichkeit berechnen. …/Winkler--Aehnlichkeit-berechnen.pdf (in Vorber.).

[59] ChatGPT auf die Frage: Welche Rolle spielt die Wahrscheinlichkeitsrechnung in der KI? – 6. 11.25.

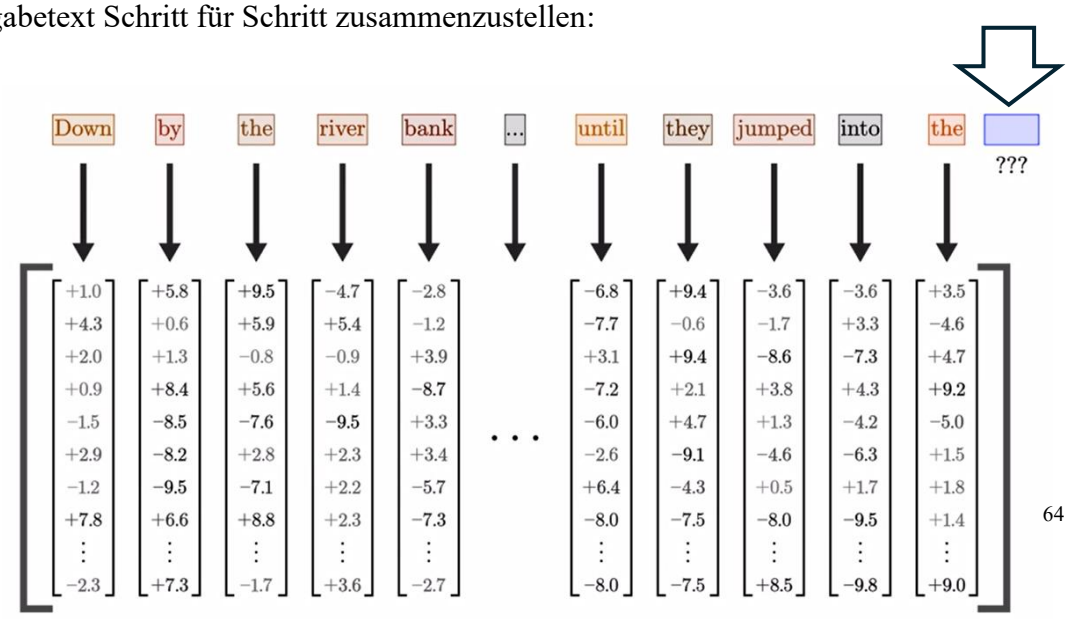
„In der realen Welt sind viele Daten unsicher oder unvollständig. Wahrscheinlichkeitsmodelle erlauben es, diese Unsicherheit mathematisch zu erfassen. Zum Beispiel in der […] Spracherkennung müssen KI-Systeme mit unscharfen und unvollständigen Eingaben umgehen. In der Bilderkennung können Objekte in Bildern variieren, was zu Unsicherheit bei der Klassifikation führt.“ [60]

17.2 Immer wieder ist in den erläuternden Texten von ‚Vorhersagen‘ die Rede.[61] Dies erscheint unverständlich, weil Prognosen zumindest auf den ersten Blick kaum eine Rolle spielen. Auch in diesem Falle aber sind statistische Methoden gemeint, und es wird etwa die Text- und Bildsynthese als Beispiel genannt:

„Einige der modernsten und spannendsten KI-Modelle basieren auf Wahrscheinlichkeitsberechnungen. ***Generative Modelle*** […] erzeugen neue Datenproben (z. B. Bilder oder Texte), indem sie Wahrscheinlichkeitsverteilungen lernen und simulieren. Sie modellieren die zugrunde liegende Verteilung der Daten ***und erzeugen neue, ähnliche [Bilder und Texte]***.“ [62]

„[Und auch] im Reinforcement Learning (RL) wird Wahrscheinlichkeitsrechnung oft verwendet, um die Unsicherheit in den Belohnungen und Zuständen zu modellieren. [Das Programm] muss die Wahrscheinlichkeit verschiedener zukünftiger Belohnungen in verschiedenen Zuständen abschätzen, um die besten Handlungen zu wählen.“ [63]

17.3 Dass generative Modelle mit ‚Vorhersagen‘ arbeiten, wird besonders deutlich, wenn das System natürlichsprachliche Texte ausgeben soll: Hier muss das System jeweils wissen (also ‚vorhersagen‘), welches Wort es als nächstes auswählen und anreihen soll, um den Ausgabetext Schritt für Schritt zusammenzustellen:



64

---

[60] Ebd.

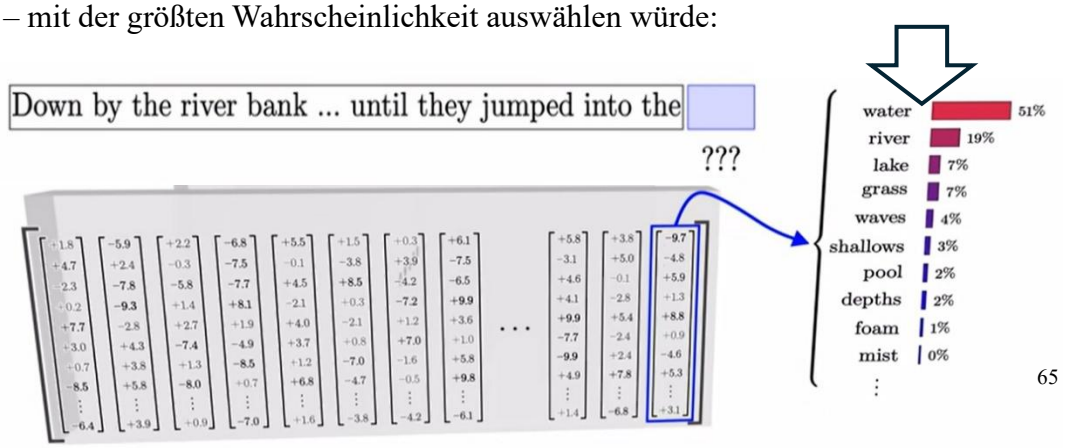
[61] „Diese Gewichte werden während des Trainings so angepasst, dass das Modell die bestmöglichen Vorhersagen macht […].“ (ChatGPT auf die Frage: was ist die Bedeutung von ‚Parametern‘, ‚Dimensionen‘ und ‚Gewichten‘ in Chatgpt? – 1. 11. 25).

[62] ChatGPT auf die Frage: Welche Rolle spielt die Wahrscheinlichkeitsrechnung in der KI? – 6. 11.25 (Erg. u. Hervorh. H. W.).

[63] Ebd. (Erg. H. W.).

[64] Grafik aus dem Video: 3Blue1Brown: Große Sprachmodelle kurz erklärt, https://www.youtube.com/watch?v=LPZh9BOjkQs, Min: 05:15.

Basis ist auch hier die ‚Wahrscheinlichkeit'; das System muss also wissen, mit welcher Wahrscheinlichkeit die verschiedenen Worte in tatsächlichen Texten aufeinander folgen. Und das wird im Trainingsprozess auf verblüffend einfache Weise gelernt: Man gibt dem Modell die Trainings-Texte schrittweise, also ***Wort für Wort*** ein; und Schritt für Schritt soll das System dann sagen, welches Folgewort es – aufgrund seiner bisherigen ‚Kenntnis' – mit der größten Wahrscheinlichkeit auswählen würde:



65

Dann wird verglichen, welches Wort im Trainingstext ***tatsächlich*** als nächstes steht. Wenn dies ein ***anderes*** ist, werden dessen Parameter entsprechend hochgesetzt, was die Wahrscheinlichkeit, dass es beim nächsten Mal ausgewählt wird, entsprechend erhöht. Diese Technik ist wirklich genial, weil hier das verarbeitete Textmaterial selbst – vollautomatisch/unsupervised – das Training des Modells übernimmt.

17.4 Ein Grundmechanismus also besteht darin, dass die Systeme sehr große Datenmengen analysieren, um daraus die jeweils wahrscheinlichste Lösung zu folgern. Und ‚Voraussage' meint die generative Funktion der KI.

18. Leistungen der KI

Auf Basis der skizzierten Vektor-Algorithmen sind die unterschiedlichsten Typen von Auswertungen möglich. Und da das Feld in ständiger Veränderung ist und die Grenze des Möglichen sich ständig verschiebt, möchte ich nur einige wenige, sehr kurze Beispiele nennen:

18.1 Neben der Analyse von natürlicher Sprache hat man spektakuläre Erfolge bei der Analyse von Audiodaten und im Feld der Bild- und Bewegtbildanalyse erzielt, und ist in der Lage all diese auch zu synthetisieren. Im Fall der Bildsynthese ist der Maßstab besonders streng (oder das Publikum besonders anspruchsvoll): 170 Jahre Technische Bilder (Fotografie und Film) haben die Bildkompetenz soweit gesteigert, dass nur tatsächlich fotoähnliche Darstellungen als ‚realistisch' durchgehen. (Gleichzeitig ist interessant, wo die augenfälligen Grenzen liegen: 99% der synthetisierten Bilder wirken eigentümlich steril; und zudem fällt auf, dass sich in den Darstellungen bestimmte Stereotypen auf eine fast aggressive Weise reproduzieren).

18.2 Ebenfalls große Erfolge gibt es in der Mustererkennung; Beispiel sind die Systeme zum ‚Autonomen Fahren', die in Realzeit mit einer Unzahl weitgehend unstrukturierter Daten umgehen müssen. Allerdings muss man sagen, dass ein tatsächlich autonomes Fahren, den vollmundigen Versprechungen zum Trotz, noch immer nicht realisiert werden konnte.

---

[65] Grafik: Ebd., Min: 06:23.

18.3 Unter theoretischem Aspekt ist interessant, dass Clustering-Algorithmen in der Lage sind, selbstständig zu klassifizieren. Das stellt die Frage neu, worum es sich bei der ***Abstraktion***, die sicherlich eine der stolzesten Fähigkeiten des Menschen und die entscheidende Leistung vieler Kulturtechniken ist, eigentlich handelt.

19. „Bedeutung“

19.1 Und schließlich, hatte ich oben geschrieben, beanspruchen die KI-Systeme, Zugang nun auch zu ‚Sinn‘ und ‚Bedeutung‘ gefunden zu haben; und dies ist tatsächlich ein entscheidender Sprung, weil man ‚Bedeutung‘ traditionell der Semantik zugerechnet und zwingend an einen menschlichen Interpreten gebunden hätte. Was also ist gemeint, wenn geschrieben wird:

> „[Die] Vektoren repräsentieren die ***Bedeutung von Wörtern oder Konzepten***“ [66] ?

oder:

> „Topic-Modeling […] ist in der Lage, verborgene ***semantische Strukturen*** in einem Dokument zu erfassen.“ [67] ?

Selbst wenn man das Vektormodell einigermaßen versteht, und im Kopf hat, dass die KIs in der Lage sind, Ähnlichkeit zu berechnen, eine Ähnlichkeit, die man als einen quasiräumlichen Abstand begreift, scheint noch immer relativ unklar, was der eigentliche Gegenstand der ‚Ähnlichkeit‘ ist.

Werden Objekte (Wörter oder Bilder der gering strukturierten Ausgangsdaten) anhand ihrer Merkmale miteinander verglichen? Oder deren Merkmale selbst? Oder deren Relationen, ihr Zusammenhang? Was ist gemeint, wenn Tigerdata, ebenfalls wenig klar, formuliert:

> “[…] where the relative positions of these [vector-] points represent ***meaningful relationships*** between the features ***or*** [?] objects.” [68]

Welchen Sinn hat es, Objekte/Items (also z. B. Wörter) allein darauf zu prüfen, ob sie – und sei es semantisch – einander „ähnlich“ sind? Würde man so nicht zunächst nur auf Synonyme kommen? Meint der Anspruch, ‚Bedeutung‘ zu repräsentieren, nicht wesentlich mehr? Auf dem Weg von den Vektoren über die ‚Ähnlichkeit‘ hin zur Bedeutung scheinen noch einige Fragen offen. Setzen wir also noch einmal an.

19.2 Merkmalsextraktion.
Zunächst: Es sind tatsächlich die Ausgangsobjekte (die Wörter oder Bilder der gering strukturierten Ausgangsdaten), die miteinander verglichen werden. Das Mittel dazu ist, sie – das war das Stichwort der ‚Merkmalsextraktion‘ oben – in ***Merkmale*** zu zerlegen, jene Merkmale eben, die die Dimensionen des Vektors bilden.

19.3 Diese Merkmale, auch das wurde gesagt, werden dem System nicht von außen vorgegeben, sondern vom System selbst ***aus dem Material extrahiert***.

---

[66] ChatGPT auf die Frage: Gehört das Training des Modells zum Embedding dazu, oder ist das Embedding abgeschlossen, wenn das Training beginnt? (Erg. u. Hervorh. H. W.).

[67] https://intrafind.com/de/blog/natural-language-processing-best-practice (Erg. H. W.);
die zitierte Seite ist ganz ausgezeichnet; sie gibt Schritt für Schritt viele Informationen, die anderswo kaum zu finden sind.

[68] Ebd., (Erg. u. Hervorh. H. W.).
Oder ähnlich AWS: „Einbettungen reduzieren die Anzahl der Dimensionen, indem Gemeinsamkeiten und Muster [?] ***zwischen verschiedenen Merkmalen*** [?] identifiziert werden.“ (Amazon, a. a. O. (Erg. u. Hervorh. H. W.)).

19.4 Und hier ist vor allem der ***Kontext*** wichtig; bei natürlichsprachlichen Texten ist dies der Kon-Text, der die einzelnen Wörter umgibt:

> „***Die Grundidee des Word Embedding besteht darin, dass der Kontext eines Wortes durch seine benachbarten Wörter definiert wird*** – meist in einer Spanne von zwei Wörtern je Richtung.[69] ***Die Umgebung eines Wortes definiert also die Semantik (Bedeutung) und die Syntax (Verwendung in der Satzstruktur) des Wortes.***
> Beim Word Embedding werden Wörter in einen Vektor umgewandelt. […] Die Wörter, die häufig zusammen im gleichen Kontext vorkommen, [...] haben auch näher beieinander liegende Vektoren.“ [70]

19.5 Was hier behauptet wird, wie gesagt, ist von großer Bedeutung. Im Kern nämlich heißt dies, dass die Neuronalen Netze ***„Bedeutung“ den tatsächlichen Texten/Kontexten entnehmen***. Nähe im materiellen Kontext wird als semantische Nähe interpretiert.

19.6 Für Probleme der Desambiguierung (die Auflösung von Mehrdeutigkeiten) ist dies plausibel:

> „Ein wesentlicher Vorteil von Transformer-basierten Modellen ist, dass sie kontextualisierte Wort-Embeddings erzeugen. Das bedeutet, dass die Repräsentation eines Wortes abhängig vom Kontext des gesamten Satzes variiert. Zum Beispiel wird das Wort ‚Bank‘ in den Sätzen ‚Ich gehe zur Bank, um Geld abzuheben‘ und ‚Ich setze mich auf die Bank im Park‘ unterschiedliche Bedeutungen annehmen, weil der Kontext bestimmt, ob es sich um eine Finanzinstitution oder eine Sitzgelegenheit handelt.“ [71]

Aber genügt das, um tatsächlich ‚Bedeutung‘ zu modellieren?

19.7 An dieser Stelle wird man sich klarmachen müssen, dass es nicht ausreicht, nur den jeweils aktuellen, einzelnen Kontext zu betrachten, weil dieser Teil einer langen Kette von immer neuen Kontexten ist. Das System wird in einer überwältigend großen Zahl von Zyklen trainiert,[72] und in jedem dieser Fälle wird der Kontext, in dem das Element steht, ein – und sei es geringfügig – anderer sein. Erst in der Gesamtheit dieser Iterationen bekommen die Verbindungen ihre Gewichte und das Netz der Vektoren insgesamt seine Form.

Und dasselbe gilt auch für die Merkmale (die Dimensionen der Vektoren). Wenn ein tiefgestaffeltes Netzwerk tatsächlich ***„billions of parameters”*** enthält,[73] dann sind das sehr viele, sehr differenzierte Merkmalsbündel, und insgesamt ein sehr großes und fein gestricktes Geflecht. Erst dieses Gesamtsystem offenbar hat die Eigenschaft, ‚Bedeutung‘ technisch zu implementieren.

Die Aufmerksamkeit für das einzelne Item verdeckt einen strukturellen Aspekt. Denn wichtig ist offenbar, dass sich, einmal vektorisiert, ***alle Items relational auf alle anderen Items beziehen***; und zwar entlang ***aller*** ‚Dimensionen‘ der Vektoren.

Es entsteht eine Verweis-Struktur, die (wie könnte es bei ‚Relationen‘ anders sein) ihren Ort im Raum ***zwischen*** den Items hat. Von jedem Item gehen Verweise aus, hin zu allen

---

[69] …das allerdings wäre, nach dem, was gesagt wurde, sehr wenig; offenbar sind es in vielen Fällen wesentlich mehr…

[70] https://intrafind.com/de/blog/natural-language-processing-best-practice

[71] ChatGPT-Antwort auf die Frage: Wie extrahieren Transformers ‚Bedeutung‘ aus dem Kontext?

[72] „Transformers can perform self-supervised learning on billions of records […]“ (Rothman, Transformers, a. a. O., S. 1).

[73] Ebd.

anderen Items; und ‚Struktur' meint nun ein ganzes Gestrüpp von Verweisen. In ihrer Gesamtheit überfordert diese Struktur (ein weiteres Mal) unser Vorstellungsvermögen; jedes einzelne Item aber wird ***relativ zu allen anderen bestimmt***.

19.8 Exakt dies nun erinnert unmittelbar an die Theorie der sprachlichen ‚Werte', die einen Kern der Saussureschen Sprachtheorie bildet. So hatte Saussure die semantische Struktur der Sprache schon 1916 als ein System bezeichnet,

> „dessen Glieder sich alle gegenseitig bedingen und ***in dem Geltung und Wert des einen nur aus dem gleichzeitig Vorhandensein des anderen sich ergeben*** […]. Da es Teil eines Systems ist, hat [das einzelne Wort] nicht nur eine Bedeutung, sondern zugleich und hauptsächlich einen Wert, und das ist etwas ganz anderes." [74]
>
> „[Eine Vorstellung ist mit einem Lautbild verbunden, es stellt die Bedeutung dar]; aber diese Vorstellung ist, wohlverstanden, nichts Primäres, sondern nur ein ***Wert***, der durch seine Verhältnisse zu andern ähnlichen Werten bestimmt ist, und ohne diese Verhältnisse würde die Bedeutung nicht existieren." [75]

Folgt man Saussure also ist Bedeutung immer schon ein Verweissystem, immer schon Struktur und in sich ***plural***. Zeichnet das Geflecht der Vektoren das nur nach?

19.9 Als zweites ist zu bedenken, dass auf Basis der Vektordatenbank ***Wahrscheinlichkeiten*** gerechnet werden: Wenn wir im fraglichen Kontext von ‚Bedeutung' sprechen, beurteilen wird nicht die Systeme selbst, sondern wir messen an dem, was sie ausgeben, ob wir die Ausgabe als ‚sinnvoll' empfinden.

Und dieser Prüfung halten die Systeme weitgehend stand, denn die Ausgaben sind zwar nicht makellos, zweifellos aber mehr als beeindruckend. Wenn KIs in der Lage sind, auf Basis von Vektordaten natürlichsprachliche Texte, Bilder oder Videos zu generieren, oder fremdsprachige Texte (in pragmatischen Grenzen) plausibel zu übersetzen, so ist das sicherlich mehr, als man noch vor 10 Jahren für irgend möglich gehalten hätte. Und wo Chomsky mit seiner generativen Grammatik noch ein mehr als kompliziertes Regelwerk entworfen hatte, scheint inzwischen, insofern die Systeme grammatikalisch korrekte Sätze ausgeben, auch dieses Regelwerk durch implizite Vorannahmen pragmatisch ersetzt.

Immer wieder aber wird von den Machern betont, dass all dies tatsächlich nur [?] Wahrscheinlichkeit und Statistik, ‚Vorhersagen' auf Basis zurückliegender Äußerungen und Texte – und insofern Fortschreibungen – sind.

19.10 ‚Fortschreibung' verweist zurück auf die Bestände, die die Vergangenheit aufgehäuft hat. Dieser Bezug auf bestehende Textuniversen, auf kulturelle Bestände und bestehendes ‚Wissen', auf Tradition (wie problematisch diese auch immer ist), und nicht zuletzt auf ***Diskursmacht***, die sich – Statistik hat es immer mit Quantitäten zu tun – immer auch darin spiegelt, in welcher Häufigkeit die Systeme bestimmte Verknüpfungen im Material finden – all dies macht die Frage nach ‚Big Data', Statistik und ‚Bedeutung' nicht leichter.

19.11 Und insgesamt allerdings muss ich zugestehen, dass mein Vorstellungvermögen bei der Frage, wie und inwiefern es tatsächlich ‚Bedeutung' ist, was hier modelliert wird, eindeutig an Grenzen kommt; und dies ist einigermaßen beschämend, weil es sich um den wahrscheinlich wichtigsten Punkt meiner Recherche handelt. Zwischen dem, was über die

---

[74] De Saussure, Ferdinand: Grundfragen der Allgemeinen Sprachwissenschaft [1916]. Berlin: de Gruyter 1967, S. 136ff. (Hervorh. u. Erg. H. W.).

[75] Ebd., S. 140 (Erg. H. W.).

Basistechniken der KI gesagt wurde, und dieser – sicherlich zentralen – Frage, und zwischen Vektortheorie und Saussure, klaffen deutliche Lücken auf.

Kehren wir deshalb zurück zu dem, was zumindest etwas sicherer ist.

20. Schluss

Der Umgang mit KI-Systemen ist binnen weniger Jahre zu einer Alltagerfahrung geworden. Es muss deshalb, denke ich, ein Interesse der Medien- und Kulturwissenschaften sein, diese Technik Schritt für Schritt ***strukturell*** besser zu verstehen, wenn sie sich nicht – aus schlichter Not – darauf beschränken wollen, ausschließlich Gebrauchsweisen zu kommentieren.

Einem solchen Verständnis näher zu kommen, ist nicht einfach. Neben der Möglichkeit, sich technisch zu informieren, wäre es sicher ein Weg, Fragen und Begriffe, die in der Analyse traditioneller Medien entwickelt wurden, an die neue Medientechnik heranzutragen und zu prüfen, ob sie nicht auch an dem neuen Medium Neues erschließen.

Ein solcher Begriff wäre z. B. die ‚Assoziation'. Das Geflecht aus Verweisen, das mit den Vektoren entsteht, erinnert an das Netz der ‚Assoziationen', mit denen Freud den psychischen Apparat, und de Saussure die Sprache beschrieben haben. Und die Assoziation zu den Assoziationen liegt sicherlich näher als die biologische unmittelbar zu den Nerven, die im Begriff der ‚Neuronalen Netze' immer schon mitgedacht ist.

Und umgekehrt verweist das neue Medium zurück auf Fragen, die auch im Feld der traditionellen Medien nach wie vor ungeklärt sind. Hier möchte ich das Problem nennen, wie die unendliche Fläche der Diskurse (Äußerungen, Texte, Bilder, Bewegtbild…) mit dem strukturellen Aspekt der unterschiedlichen symbolischen Systeme (der Sprache, den Bildsystemen..,.) zusammengedacht werden kann. Dass beide dialektisch aufeinander verweisen, ist wenig strittig. Wenn die Neuronalen Netze nun aber in der Lage sind, mit den Mitteln der Vektorrechnung aus Diskursen/Kontexten „Bedeutung" zu extrahieren – haben die Computerleute die Kulturwissenschaften ***beschämt***?

Die Systeme führen vor, ***dass syntagmatische Nähe in paradigmatische Nähe umschlägt***, also Kontiguität in Similarität, Nähe im Diskurs in Semantik**.** Exakt dies habe ich auf dem Terrain der traditionellen Medien verschiedentlich zu zeigen versucht. Aber schöpft die IT-Modellierung das Problem der ‚Bedeutung' tatsächlich aus? Oder gibt es systematische Grenzen? Eine Art systematisches Missverständnis, das die jeweiligen Auffassungen von ‚Bedeutung' schließlich doch voneinander trennt?

Ich möchte es hierbei zunächst belassen.