

Mit präzisen Daten den Tumor gezielter bekämpfen

Jeder Tumor ist einzigartig. Das macht es schwierig, die wirksamste Therapie für dessen Behandlung zu finden. Forschende in Zürich und Basel zeigen nun: Mithilfe modernster molekularbiologischer Verfahren lässt sich innerhalb von vier Wochen ein detailliertes Tumorphil erstellen, das eine massgeschneiderte Therapie ermöglicht. Die Studie ist die weltweit erste dieser Art.

Bei der Behandlung von Krebs stützen sich die Ärztinnen und Ärzte jeweils auf etablierte Leitlinien. Dank diesen wurden etwa bei schwarzem Hautkrebs (Melanom) in den letzten Jahren signifikant bessere Behandlungserfolge erzielt. Allerdings gibt es innerhalb der Standardtherapien oft mehrere Behandlungsoptionen, und es ist nicht immer eindeutig, welche Therapie am ehesten zum Erfolg führen könnte. Noch schwieriger wird es, wenn die Standardtherapien ausgeschöpft sind und es kaum fundierte Hinweise gibt, wie die Behandlung fortgesetzt werden könnte.

Jede einzelne Zelle des Tumors

Welche Therapieform die beste ist wird bisher in erster Linie anhand des Ursprungsgewebes des Tumors sowie seiner genetischen Eigenschaften festgelegt. Im seit 2018 laufenden Tumor Profiler-Projekt wird nun untersucht, wie neue molekularbiologische Methoden helfen können, die Therapiemöglichkeiten zu verbessern und zu erweitern. Dafür machen sich die Forschenden zunutze, dass jeder Tumor bis in die einzelnen Zellen hinein einzigartig ist.

Die Forschenden analysieren dazu mit neun Technologien die Tumore auf Einzelzellebene. Dadurch entsteht ein umfassendes Bild der biologischen Vorgänge im Tumor. Dieses Wissen soll es möglich machen, aus den vorhandenen Therapieformen die individuell wirksamste Behandlung zu bestimmen. Der datenbasierte Ansatz erlaubt es zudem, Medikamente in die Evaluation einzubeziehen, die für die Behandlung anderer Krebsarten eingesetzt werden. Dadurch erweitert sich das Spektrum an Therapiemöglichkeiten.

In einer ersten Phase des Projekts wurde untersucht, welche molekularbiologischen Technologien relevante Informationen liefern. Zudem wurde in dieser Phase gezeigt, dass solche umfassenden

Analysen machbar sind und die enormen Datenmengen auch verarbeitet werden können. In einem weiteren Schritt ging es darum zu prüfen, wie das Tumorphil in der Praxis angewendet werden kann.

Grosse Datenmengen bewältigbar

In einer prospektiven, multizentrischen Beobachtungsstudie untersuchte die Forschungsgruppe aus über 100 Wissenschaftlerinnen und Wissenschaftlern des Universitätsspitals Zürich, der Universität Zürich, der ETH Zürich, des Universitätsspitals Basel und der Firma Hoffmann-La Roche, ob dieser Ansatz in der Klinik machbar ist und ob er Vorteile bietet. Im Fokus stand dabei die Frage, wie lange es dauert, bis die Tumoranalyse vorliegt und wie die behandelnden Ärztinnen und Ärzte die daraus resultierenden Empfehlungen beurteilen – zwei zentrale Faktoren für die erfolgreiche Anwendung in der Praxis. Für die Studie wurden Tumoren von 116 Patientinnen und Patienten analysiert. Aus den resultierenden 43 000 Datenpunkten pro Probe wurden individuelle Behandlungsempfehlungen abgeleitet.

Die aus dem Tumorphil gelieferten Empfehlungen lagen jeweils nach vier Wochen vor und wurden in 75 Prozent der Fälle von den behandelnden Spezialisten als hilfreich beurteilt. Es zeigte sich zudem, dass die Patientinnen und Patienten, deren Behandlung auf Informationen aus den Profiler-Daten beruhte, häufiger auf die Therapie ansprachen als die Patientinnen und Patienten, die nicht am Programm teilnahmen.

Die ermutigenden ersten Resultate müssen nun noch in prospektiven und randomisierten klinischen Studien mit mehr Patienten bestätigt werden. Die beteiligten Forschenden sind aber überzeugt, dass diese Studie ein grosser Schritt in Richtung datenbasierte Medizin ist.

Der Artikel basiert auf einer Medienmitteilung der Universität Zürich

Quelle

Miglino N. et al. 2025. Feasibility of multiomics tumor profiling for guiding treatment of melanoma. *Nature Medicine*. DOI: <https://doi.org/10.1038/s41591-025-03715-6>

Wie KI funktioniert – vom Hopfield-Netzwerk zur Boltzmann-Maschine

Im letzten Heft habe ich das Hopfield-Netzwerk vorgestellt, das Muster – ähnlich wie unser Hirn – verteilt auf viele Neuronen speichert. In diesem Artikel zeige ich, welche Modifikationen Geoffrey E. Hinton vorgenommen hat, um die Nachteile von Hopfield-Netzwerken zu eliminieren und sie wesentlich leistungsfähiger zu machen. Dies führte schliesslich zur Entwicklung von «Transformern» und damit zu den KI-Systemen, die gegenwärtig die Welt erobern.

Wie Geoffrey Hinton das Hopfield-Netzwerk modifizierte

Die seit den 1960er-Jahren versuchte Entwicklung der künstlichen Intelligenz kulminierte mit dem Backpropagation-Mechanismus. Dies war eine mathematische Optimierungsmethode, die interessante Ergebnisse lieferte, aber Ende der 1970er-Jahre in Ermangelung weiterer Entwicklungsmöglichkeiten in eine Krise geriet. John Hopfield suchte einen Neuanfang mit Konzepten, die sich stärker an biologischen Hirnen orientierten, und brachte damit neuen Schwung in die KI-Community. Dazu entwickelte er die sogenannten Hopfield-Netzwerke.

Um zu illustrieren, wie Geoffrey E. Hinton (Abb. 1) das Hopfield-Netzwerk weiterentwickelte, werde ich dieselben Muster verwenden, mit denen ich im vorhergehenden Heft das Hopfield-Netzwerk erklärt habe. Ich bleibe also bei Schriftzeichen auf einem 5×6 Punkte Raster. Jedes Symbol entspricht jeweils einem 30 Bit langen Zeichen-Vektor, wobei 1 für Schwarz und 0 für Weiss steht. So lässt sich die *Boltzmann-Maschine*, wie Hinton sein Modell genannt hat, leichter mit dem Hopfield-Netzwerk vergleichen und auch besser verstehen. Was dieses Modell mit dem berühmten österreichischen Physiker Ludwig Boltzmann (1844-1906) zu tun hat, wird später klar werden.

Im vorhergehenden Artikel habe ich gezeigt, wie das Hopfield-Netzwerk in ungewollten Fixpunkten stecken bleiben kann. Diese ungewollten Fix-



Abb. 1: Geoffrey E. Hinton (geb. 1947) bei einem Vortrag in Toronto 2024. (Bild: Vaughn Ridley, wikimedia commons, CC BY 2.0)

punkte entsprechen Mustern, die nicht einprogrammiert wurden. Ein Beispiel war die Kombination der Zeichen F und O, die fälschlicherweise als stabiler Fixpunkt auftrat.

Hinton griff wie zuvor Hopfield auf das Ising-Modell des Ferromagnetismus zurück. Das Ising-Modell enthält als wichtigen Parameter die Temperatur. Die Bewegungsenergie der Atome in einem Kristallgitter ist proportional zur Temperatur. Obwohl diese an ihren Gitterplätzen festsitzen, können sie sich in allen drei Richtungen etwas hin- und herbewegen, also Schwingungen ausführen. In diesen Schwingungen steckt Energie, die proportional zur absoluten Temperatur (Masseinheit: Kelvin, K) ist. Beim absoluten Nullpunkt ($0 \text{ K} = -273,15^\circ\text{C}$) ist diese Schwingungsenergie praktisch Null (bis auf die sehr kleine Nullpunktsenergie).

Erhitzt man einen Eisenmagneten über 768°C (die sogenannte Curie-Temperatur), verschwindet die Magnetisierung, weil die intensiven Temperatur-

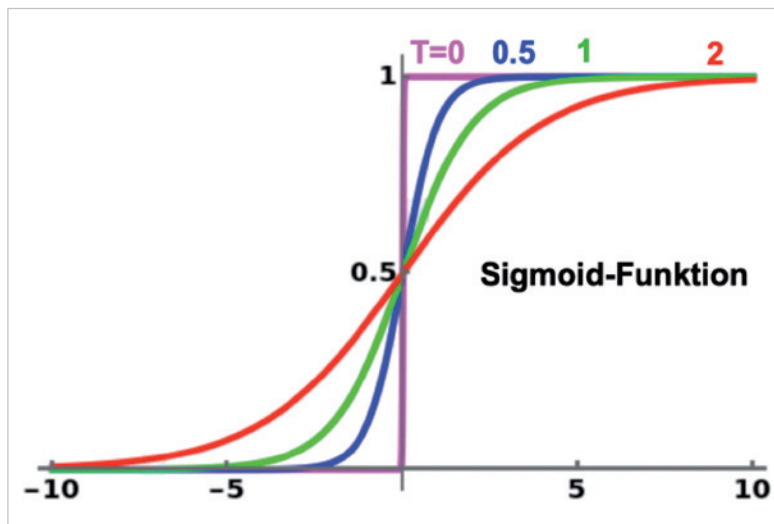


Abb. 2: Sigmoid-Funktion (auch Logistische Funktion genannt) für die «Temperaturen» $T=0, 0,5, 1$ und 2 . Als Beispiel nehmen wir an, dass die Summe, die sich aus der Multiplikation der Gewichtsmatrix mit einem Zeichen-Vektor für eine bestimmte Komponente ergibt, 5 sei (x -Achse).

Die Normierung, also die Umwandlung der Komponente in 1 oder 0 , gestaltet sich nun etwas aufwändiger als bei Hopfield: Mit Hilfe eines Zufallszahlen-Generators wird im Intervall $0 \dots 1$ (y -Achse) zunächst eine zufällige Zahl berechnet. Ist diese Zahl kleiner als der Wert auf der Sigmoid-Kurve der entsprechenden «Temperatur», wird die Komponente auf 1 gesetzt, wenn sie grösser ist auf 0 .

Beträgt die Zufallszahl z.B. $0,6213$, wird die Komponente bei einer «Temperatur» von 2 auf 1 gesetzt, da die Zahl kleiner ist als der Wert auf der roten Funktion. Beträgt die Zufallszahl hingegen $0,9524$, wird die Komponente bei einer «Temperatur» von 2 auf 0 gesetzt, weil sie höher ist als der Wert auf der roten Funktion. Beträgt die «Temperatur» hingegen 1 , wird die Komponente immer noch auf 1 gesetzt, da die Zufallszahl unter der grünen Funktion liegt. Bei einer «Temperatur» von 0 (violette Kurve) wird die Komponente bei positiven Summenwerten immer auf 1 gesetzt und auf 0 bei negativen Summenwerten (das entspricht dem Hopfield-Netzwerk). Die Grafik zeigt, dass selbst bei positiver Summe eine 0 für die Komponente umso wahrscheinlicher wird, je höher die «Temperatur» ist. (Bild Fritz Gassmann)

bewegungen die vorher parallel zueinander ausgerichteten «Elementarmagnetchen» aus der Ordnung bringen. Die Ordnung muss dann bei tieferer Temperatur durch ein äusseres Magnetfeld erst wieder hergestellt werden, um den Magneten zu reaktivieren. Das Ising-Modell kann all diese Vorgänge simulieren.

Hinton übertrug diesen Fluktuationsprozess in sein neuronales Netz, wobei die ursprüngliche Bedeutung der physikalischen Temperatur verloren ging und einfach als Mass für Fluktuationen verstanden werden soll. Um das zu verstehen, stelle man sich elektrische oder chemische Fluktuationen vor, die in einem biologischen Hirn auftreten. Ähnlich wie beim Begriff «Energie», der vom Ising-Modell übernommen wurde, sprechen die Physiker von der «Temperatur» des neuronalen Netzes, um die Fluktuationen zu beschreiben. Um anzudeuten, dass es sich hierbei um eine Übertragung handelt, werde ich im folgenden «Temperatur» wie «Energie» in Anführungszeichen schreiben.

Die «Temperatur» bzw. die Fluktuation bewirkt in einem neuronalen Netz, dass flache relative Minima verlassen werden können, indem die zufälligen Bewegungen das System sozusagen über die Ränder schubsen.

Hinton baut «Temperatur» ein

Hinton hat die «Temperatur» nach bewährter physikalischer Manier ins Hopfield-Netzwerk so eingebaut, dass beim «absoluten Nullpunkt», also bei $T=0$, seine Boltzmann-Maschine identisch mit dem Hopfield-Netzwerk wird. So konnte er erreichen, dass sein Modell die interessanten Eigenschaften des Vorgängermodells behält.

In der Vierteljahrsschrift 1|2025 habe ich detailliert gezeigt, wie das Hopfield-Netzwerk funktioniert. Entscheidend ist dabei, dass ein beliebiger Vektor mit N Komponenten mit Hilfe der quadratischen und symmetrischen $N \times N$ -Gewichtsmatrix w_{ik} in einen neuen Vektor derselben Länge transformiert werden kann. Multipliziert man die Gewichtsmatrix w_{ik} mit dem Vektor, entstehen Summen, die positiv oder negativ sein können. Hopfield hat die einfachste Methode gewählt, um wieder Vektoren mit Komponenten 1 oder -1 zu erhalten: Ist die Summe positiv, wird die entsprechende Komponente auf 1 gesetzt, sonst auf -1 . Eine solche Umrechnung, bei der aus Vektoren mit beliebigen Zahlen x als Komponenten wieder Vektoren mit Komponenten 1 oder -1 entstehen, nennt man *Normierung*.

Hier hat Hinton angesetzt: Er hat die Komponenten mit den Werten -1 durch die Werte 0 ersetzt

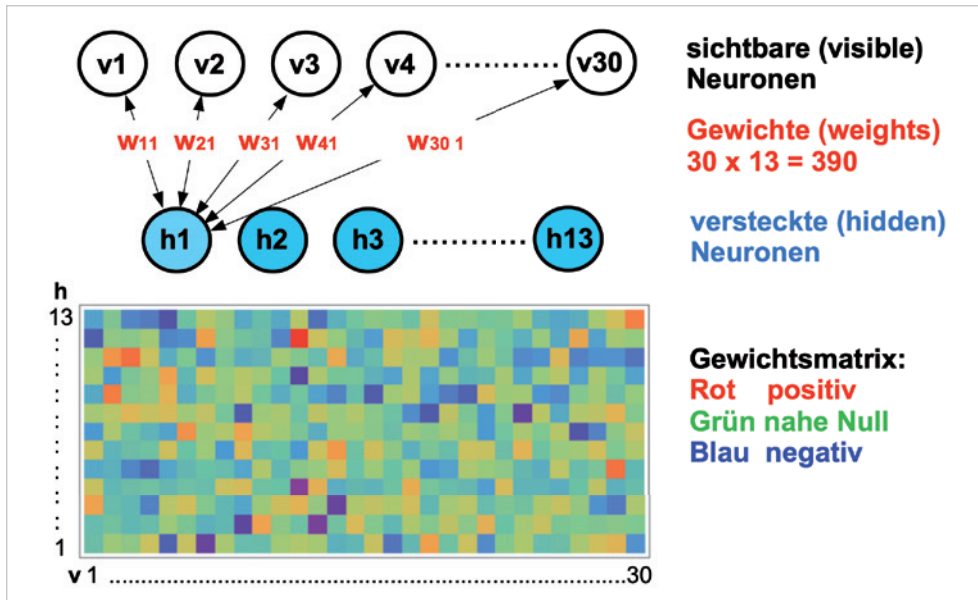


Abb. 3: Struktur der Restricted Boltzmann Machine. Die sichtbaren Neuronen ($v_1 \dots v_{30}$, in der oberen Hälfte der Grafik in Weiss dargestellt) sind untereinander nicht verbunden, sondern ausschliesslich mit den versteckten Neuronen ($h_1 \dots h_{13}$, blau). Diese sind wiederum untereinander nicht verbunden, doch kann jedes versteckte Neuron mit denselben Gewichten auf jedes sichtbare Neuron zurückwirken. Um die Figur nicht zu überladen, sind nur die Verbindungen zwischen dem versteckten Neuron h_1 und einigen sichtbaren Neuronen eingezeichnet sowie deren Gewichte w_{i1} (Rot) angegeben.

In der unteren Hälfte ist ein Beispiel einer Matrix mit 390 Gewichten in Farbcodierung wiedergegeben. Diese Gewichtsmatrix ist nicht mehr quadratisch, sondern rechteckig, weil weniger versteckte als sichtbare Neuronen vorhanden sind. (Bild F. Gassmann)

(dies ist unwesentlich) und die Summe mit Hilfe einer Wahrscheinlichkeit in den neuen Zustand umgerechnet (dies ist wesentlich). Abb. 2 zeigt, wie dies funktioniert und welche Rolle die «Temperatur» dabei übernimmt. Wird $T=0$ gesetzt, entsteht die Sprungfunktion (engl. *step function*), die Hopfield benutzt hat. Je grösser T ist (T kann nur eine positive Zahl sein), desto unsicherer wird der zu berechnende Vektor, d.h. bei mehrmaliger Berechnung entstehen umso unterschiedlichere Resultate.

Hinton bringt Netzwerk «Lernen» bei

Im Hopfield-Netzwerk besteht die Gewichtsmatrix aus starren positiven oder negativen Zahlen, nachdem sie durch einen einmaligen Prägungsprozess erzeugt wurde. Diesen Prägungsprozess habe ich im vorhergehenden Artikel detailliert beschrieben, wobei ich auf die Neuronen fokussiert habe, um die Verwandtschaft mit biologischen neuronalen Netzwerken hervorzuheben.

Fokussieren wir jedoch auf die N -dimensionalen Vektoren und verwenden die Mathematik der Vektorrechnung, erhalten wir ein elegantes Resultat:

Die in der Gewichtsmatrix eingepprägten Vektoren gehen bei der Multiplikation mit der Gewichtsmatrix nach der Normierung in sich selbst über, sie werden also reproduziert. Mathematisch ausgedrückt: $\text{Norm}(M \cdot F) = F$. Dabei bedeutet Norm die Normierung nach Abb. 2, M ist die Gewichtsmatrix und F steht für einen in die Gewichtsmatrix eingepprägten Vektor. Da bei dieser Multiplikation alle N Teilschritte für die N Komponenten nun als einen Rechenschritt betrachtet werden, wird dieser grössere Schritt als *Epoche* bezeichnet. Da die Vektoren in unseren Beispielen immer $N=30$ Komponenten haben, entsprechen 30 Rechenschritte im Hopfield-Modell nur einer Epoche im Hinton-Modell.

Im Hopfield-Modell beginnt man mit einem zufälligen Input-Vektor, multipliziert ihn mit der Matrix und normiert ihn anschliessend. So entsteht ein neuer und gleich langer Vektor, da die Matrix quadratisch ist. In einem Kreisprozess wird dieser neue Vektor wieder mit der Matrix multipliziert und dann normiert. Die Abb. 2 des Artikels im letzten Heft zeigt, wie sich aus diesem Kreisprozess das Muster F entwickelt. Nach dem oben erläuterten

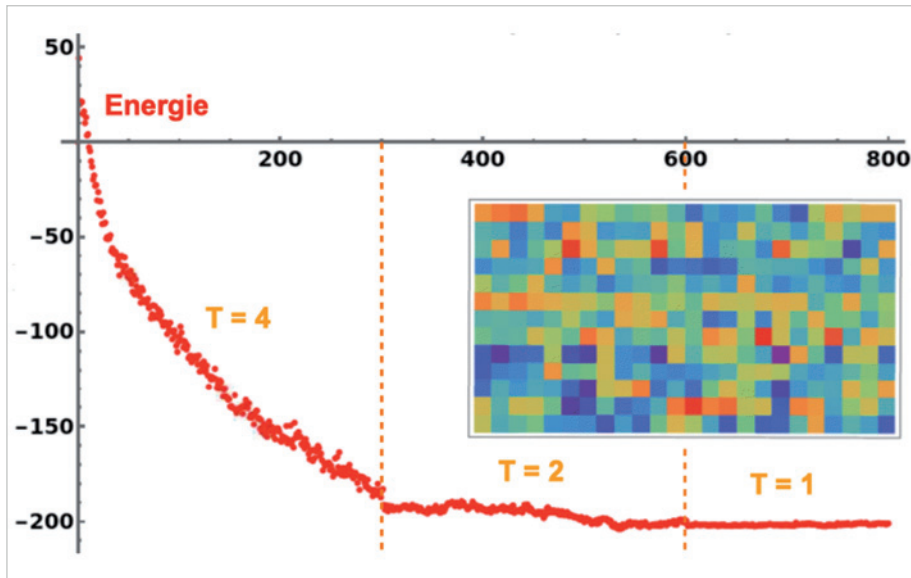


Abb. 4: Training der Hinton-Gewichtsmatrix mit 24 Symbolen in total 800 Epochen. Rote Punkte bedeuten die über die 24 Symbole gemittelte Energie. Der Start war bei 43 und das Energieminimum wurde nach rund 500 Epochen bei etwa -200 erreicht. Um die Rechnung zu beschleunigen, wurde die Temperatur für die ersten 300 Epochen auf 4 gesetzt: Die Fluktuationen sind deutlich zu sehen. Weitere 300 Epochen wurden mit $T=2$ gerechnet, weshalb die Fluktuationen kleiner wurden. Der letzte Teil wurde mit $T=1$ gerechnet und zeigt einen sehr ruhigen beinahe stationären Verlauf. Die trainierte Matrix nach 800 Epochen ist in einer Farbcodierung dargestellt. (Bild und Rechnung F. Gassmann)

eleganten Resultat passiert dann nichts mehr, selbst wenn der Kreisprozess weiterläuft, weil sich F nur noch reproduziert.

Nun kommt Hinton's entscheidende neue Idee dazu: Er hat gesehen, dass dieser Kreisprozess dem Abwärtsfließen von Wasser in einem strukturierten Gelände bis zum tiefsten Punkt entspricht. Eine geografische Geländeform, in der Wasser abwärts fließt, wurde jedoch durch frühere geologische Prozesse geformt und die heutige Form erinnert an diese Prozesse. Will man ein neuronales Netz lernfähig machen, müsste man sich demnach nicht auf das Abwärtsfließen des Wassers fokussieren, sondern auf die Geländeform. Der Lernprozess in einem neuronalen Netz entspricht dann dem geologischen Prozess, der zu einer langsamen Änderung des Geländes führt.

In einem neuronalen Netz entspricht die Geländeform der Gewichtsmatrix, dementsprechend müsste also die Gewichtsmatrix in kleinen Schritten umgeformt werden, um den langsamen geologischen Formationsprozess oder den biologischen Lernprozess zu simulieren. Es sei in diesem Zusammenhang an das französische Wort *formation* für *Bildung* oder *Ausbildung* erinnert.

Hinton hat diese Idee vorerst mit quadratischen Matrizen umgesetzt, aber dann mit Hilfe numerischer Experimente festgestellt, dass seine Boltzmann-Maschine noch viel interessanter wird, wenn er die Neuronen in zwei separate Schichten unterteilt, in sogenannte sichtbare und unsichtbare Neuronen, und gleichzeitig die Anzahl Verbindungen zwischen den Neuronen drastisch reduziert. Abb. 3 zeigt die Struktur seiner erfolgreichen *Restricted Boltzmann-Machine*.

Vergleichen wir nun das Modell von Hopfield mit demjenigen von Hinton etwas genauer: Bei Hopfield haben wir den Input-Layer mit $N=30$ Neuronen, den wir durch Multiplikation mit der quadratischen Matrix Q und anschließender Normierung in den Output-Layer mit wiederum N Neuronen transformieren oder abbilden: $\text{Out} = \text{Norm}(Q \cdot \text{In})$. Um den Kreisprozess zu schließen, müssen wir nur noch den Output-Layer auf den Input-Layer kopieren ($\text{In} = \text{Out}$) und schon ist die Epoche abgeschlossen und der iterative Prozess kann weitergehen. Die benötigte quadratische Matrix Q hat $N^2=900$ Gewichte.

Bei Hinton beginnen wir mit demselben Input-Layer mit $N=30$ Neuronen, multiplizieren diesen aber nun mit einer rechteckigen Matrix R mit

14 FORSCHUNG – PHYSIK IM ALLTAG

$N=30$ Kolonnen und $M=13$ Zeilen (und anschliessender Normierung). Das Resultat dieser Abbildung $H = \text{Norm}(R \cdot \text{In})$ ist ein komprimierter Vektor mit nur 13 Komponenten (H steht für Hidden, versteckt), den wir nicht mehr einfach auf den Input-Layer kopieren können. Wir brauchen also einen zusätzlichen *Expansionsschritt*, um wieder einen Vektor mit N Komponenten zu erzeugen und den Kreisprozess zu schliessen. Diesen Expansionsschritt erzeugen wir mit derselben Matrix, wobei wir allerdings die Kolonnen und die Zeilen vertauschen müssen, damit wir die kürzeren Vektoren H mit der Matrix multiplizieren können. Man nennt die so entstehende Matrix transponiert und bezeichnet sie als R^T : $\text{Out} = \text{Norm}(R^T \cdot H)$. Nun können wir den Kreisprozess wieder schliessen wie vorher: $\text{In} = \text{Out}$. Die für den gesamten zweiteiligen Prozess benötigte Matrix R hat nur $N \times M = 390$ Gewichte. R^T ist keine zusätzliche neue Matrix, sondern nur die anders gelesene alte Matrix R . Für grosse Vektoren mit Millionen von Komponenten, wie sie beispielsweise für Bilder gebraucht werden, sind die Grössen der Matrizen entscheidend für den riesigen Bedarf an Speicherplatz, die Anzahl der Neuronen ist im Vergleich dazu sehr klein.

Das Fazit ist verblüffend: Die in **Abb. 3** wiedergegebene Hinton-Matrix enthält die Muster von 24 Symbolen, im Gegensatz zu nur 4 Symbolen in der grösseren Hopfield-Matrix.

Um die Gewichte in vielen Epochen langsam einzustellen, musste Hinton auch den einfachen Hopfield-Mechanismus anpassen: Mit dem *Kompressionsschritt* wird nach der Hopfield-Vorschrift eine sog. *Aktivierungsmatrix* A und im *Expansionsschritt analog* eine Aktivierungsmatrix A' berechnet.

Eine theoretische Ableitung zeigt, dass ein Lernschritt wie folgt aussehen muss: $w_{\text{neu}} = w + \delta(A-A')$. Die Matrizen A und A' haben beide 30×13 Elemente wie w und das Update w zu w_{neu} muss für jedes der 390 Elemente durchgeführt werden. δ ist der Lernschritt, der ausprobiert werden muss: Ist er zu gross (z.B. 10), wird die Rechnung instabil, ist er zu klein (z.B. 0,001), wird die Rechnung sehr langsam. Optimale Werte bei den hier wiedergegebenen Rechnungen waren $\delta = 0,1 \dots 0,2$.

Abb. 4 zeigt, wie die mittlere «Energie» der Symbole während 800 Epochen des Lerntrainings von 24 Symbolen abnimmt. Sobald die minimale «Energie» erreicht wird, ist der Lernprozess abgeschlossen. Dies dauerte auf meinem Raspberry Pi (Computer der Grösse einer Zündholzschachtel) rund

eine Minute. Man vergleiche diese Zeit mit der Lernzeit eines Primarschülers für das Alphabet.

Was passiert beim Training des neuronalen Netzes?

Vor dem Training wird die Gewichtsmatrix üblicherweise mit Zufallszahlen gefüllt, die schwerpunktmässig um Null herum liegen. Nach einem ersten Durchgang des in **Abb. 5** gezeigten zyklischen Lernprozesses (d.h. nach der ersten Epoche) ist die Rekonstruktion S_v' der Symbole noch sehr schlecht, weil die Matrix noch nicht entsprechend der Trainings-Symbole S_v geformt wurde. S_v' und S_v sind also noch verschieden und die Korrekturmatrizen A und A' , die in den beiden Teilschritten Kompression und Expansion berechnet werden, haben verschiedene Einträge. Deren Differenz multipliziert mit dem Lernschritt $\delta(A-A')$ wird dann benutzt, um die Gewichtsmatrix schrittweise zu verbessern oder zu formen.

Wie das Hopfield-Netzwerk hat auch das Hinton-Netzwerk eine Kapazitätsgrenze, die sich aber nicht so einfach angeben lässt, weil sie von den zu lernenden Symbolen abhängt. Ich habe für mein Beispiel mit 30 sichtbaren und 13 versteckten Neuronen die Kapazitätsgrenze gesucht und sie bei gegen 24 Symbolen gefunden: Die Symbole N und W sind dem Netzwerk nicht mehr beizubringen, obwohl die Fehlleistung bei W noch verzeihlich erscheint. Man beachte aber, dass die geringere Höhe des rekonstruierten W von der ebenfalls geringeren Höhe von N herrührt. Das Netzwerk kann also nur die restlichen 22 Symbole einwandfrei lernen.

Die trainierte Matrix kann neue Symbole generieren

Der Raum der sichtbaren S_v -Vektoren umfasst rund eine Milliarde (2^{30}) verschiedene Symbole. Im Kompressionsschritt, bei dem die Matrix A entsteht, wird jeder S_v -Vektor mit 30 Komponenten auf einen versteckten S_h -Vektor mit nur 13 Komponenten komprimiert. Im Raum dieser S_h -Vektoren gibt es nur rund 8000 (2^{13}) verschiedene Symbole, also etwa 130 000 (2^{17}) mal weniger als im S_v -Raum.

Bei der nachfolgenden Expansion von 13 auf 30 Komponenten, bei der die Matrix A' entsteht, kann die Variabilität der möglichen Vektoren nicht wesentlich grösser werden, d.h. alles, was mit der «trainierten» Matrix später angefangen wird, spielt sich in einem sehr kleinen Teilraum des ursprüng-

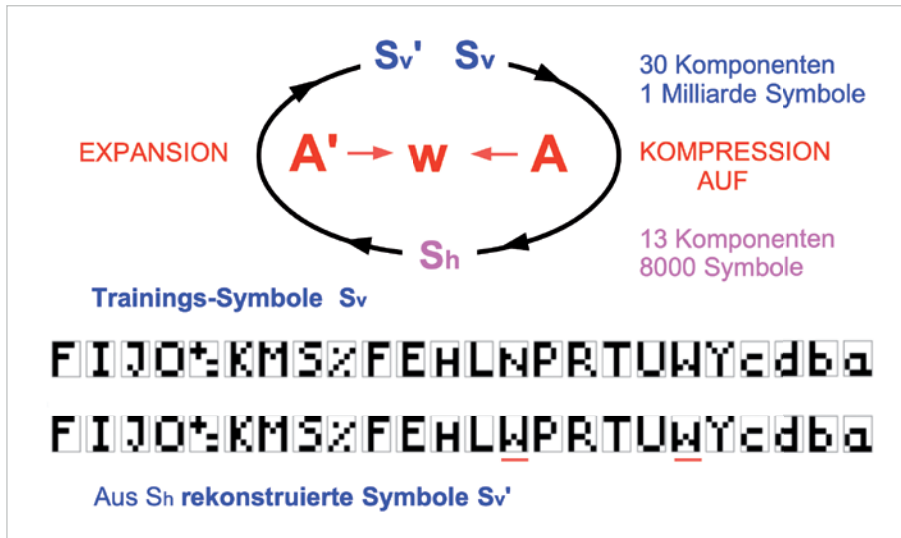


Abb. 5: Der zyklische Trainingsprozess formt die Gewichtsmatrix schrittweise um, bis die Trainings-Symbole S_v perfekt rekonstruiert werden, also $S_v' = S_v$ werden. Dies ist beim Erreichen der minimalen «Energie» der Fall und dann wird auch $A = A'$, d.h. die Matrix w wird nicht mehr weiter umgeformt, weil $\delta(A - A')$ Null wird. Die gezeigte Rekonstruktion ist jedoch nicht ganz perfekt: N und W bereiten Schwierigkeiten (vgl. Erklärungen im Text). (Bild F. Gassmann)

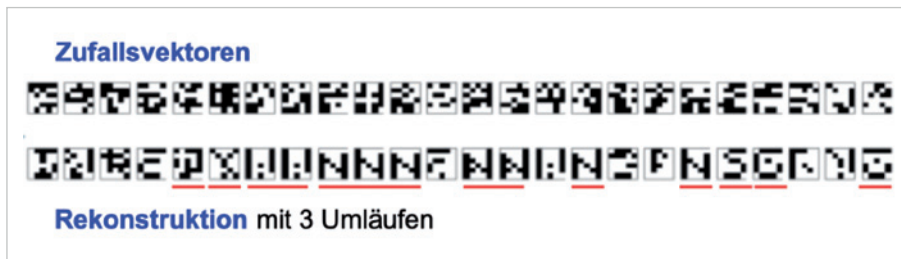


Abb. 6: Rekonstruktion von 24 zufälligen Vektoren mit Hilfe der auf unsere 24 Symbole trainierten Matrix. Der in Abb. 5 dargestellte Zyklus wurde dreimal durchlaufen. Die rot unterstrichenen besonders interessanten Resultate sind im Text erklärt. (Bild F. Gassmann)

lich eine Milliarde Vektoren enthaltenden Raumes ab. Was dies bedeutet, soll am folgenden Beispiel verständlich gemacht werden.

Wir geben dem neuronalen Netz 24 zufällige Vektoren (mit je 30 Komponenten) vor und lassen mit jedem Vektor den in Abb. 5 dargestellten Zyklus dreimal durchlaufen, wobei wir den Lernschritt $\delta = 0$ setzen, damit die Gewichtsmatrix nicht verändert, also nicht weiter geformt wird. Abb. 6 zeigt das Resultat.

Zur Erinnerung: Hätten wir trainierte statt zufällige Vektoren vorgegeben, würden in beiden Zeilen dieselben Symbole stehen (vgl. Abb. 5).

Als erstes kann festgestellt werden, dass alle rekonstruierten Symbole in der unteren Zeile eine gewisse Ähnlichkeit mit dem Trainings-Set in Abb. 5

aufweisen. Die Resultate 9, 11 und 20 sind sogar identisch mit zwei Symbolen im Trainings-Set. Weiter sind die Resultate 10, 13, 14, 16 und 19 sehr nahe beim N im Trainings-Set. Auch 7, 8 und 15 lassen das W erkennen. Interessant und geradezu innovativ ist aber Nr. 21, das ein G sein könnte, obschon dies nicht im Trainings-Set enthalten ist. Ansprechend ist auch Nr. 24 als eine Art Smiley. Nr. 6 scheint eine Kombination von Y und K zu sein und Nr. 5 erinnert an ein menschliches, aber fremdländisches Symbol.

Ein Blick auf die «Energie» zeigt, wie aus Chaos Kreativität wird

Abb. 4 zeigt, wie die mittlere «Energie» der Datenvektoren S_v beim Lernprozess abnimmt. Es lohnt sich jedoch, genauer hinzuschauen und die einzel-

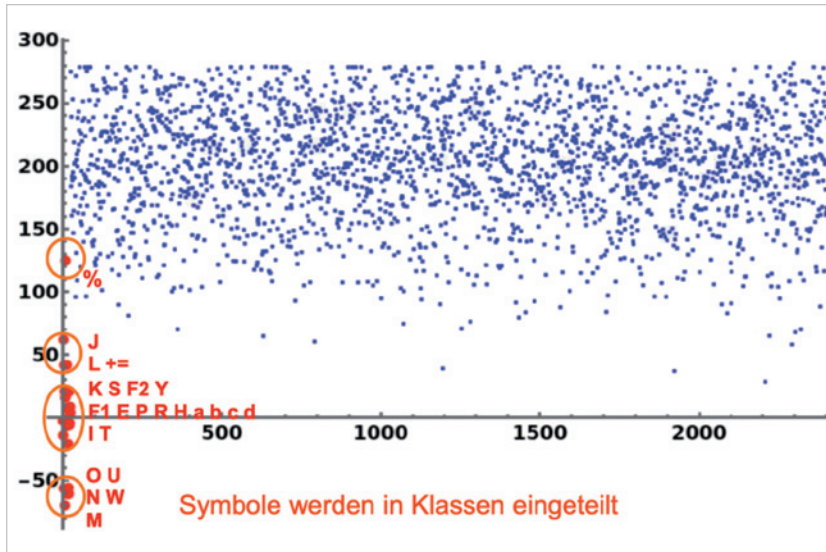


Abb. 7: «Energien» von 2400 zufälligen Vektoren (blau) und «Energien» der 24 trainierten Vektoren (rot). Durch das Training wird die Gewichtsmatrix derart verändert, dass die «Energie» der trainierten Vektoren reduziert wird. Die restlichen Vektoren konvergieren gegen eine Boltzmann-Verteilung (Exponentialverteilung), wenn das thermodynamische Gleichgewicht erreicht wird. Im hier gezeigten Beispiel ist dies der Fall unterhalb etwa 200. Oberhalb ist das Gleichgewicht noch nicht erreicht. Weitere Erklärungen im Text. (Bild und Rechnung F. Gassmann)

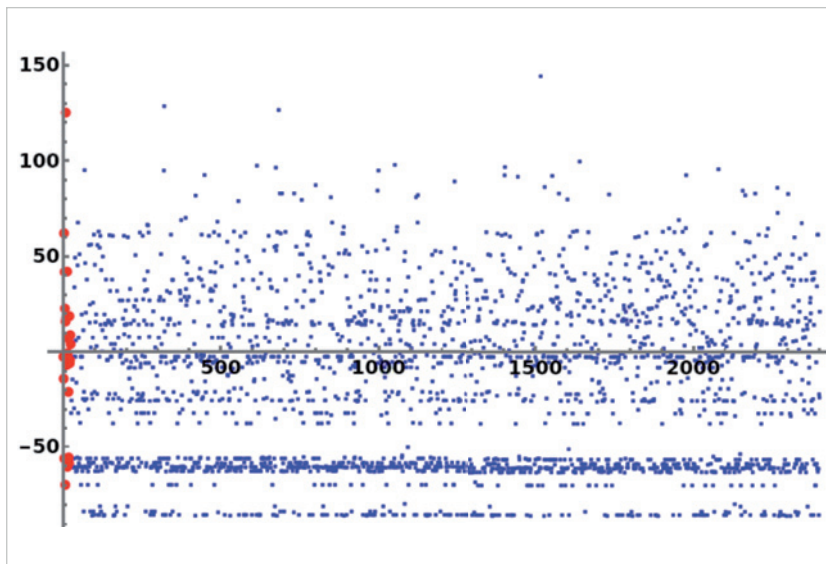


Abb. 8: Blaue Punkte: «Energien» der rekonstruierten 2400 zufälligen Vektoren mit Hilfe der auf unsere 24 Symbole trainierten Gewichtsmatrix. Der in Abb. 5 dargestellte Zyklus wurde je dreimal durchlaufen. Rote Punkte: «Energien» der trainierten Symbole. (Bild F. Gassmann)

nen Vektoren und auch die nicht trainierten Vektoren zu betrachten. Weil sich die «Energie» aus der Kombination der Gewichtsmatrix w mit den Vektoren S_h und S_v ergibt, [mathematisch formuliert ist $E = S_h \cdot (w \cdot S_v)$], wirkt sich das Training auch auf die «Energie» der überwiegenden Mehrheit der Vektoren aus, die nicht beachtet wurden.

In Abb. 7 sind die «Energien» von 2400 der Milliarde nicht trainierter Vektoren als blaue Punkte dargestellt. Es ist klar erkennbar, dass unterhalb etwa der «Energie» 200 die Anzahl Punkte pro «Energie»-Intervall abnimmt. Eine statistische Analyse ergab, dass diese Abnahme regelmässig erfolgt, also beispielsweise für eine geeignete Intervallgrösse um immer den Faktor 2. Eine solche Verteilung nennt man Exponentialverteilung

oder in der Thermodynamik auch Boltzmann-Verteilung. Deshalb nannte Hinton sein neuronales Modell «Boltzmann-Maschine». Auch die Moleküle in der Erdatmosphäre gehorchen annähernd einer Boltzmann-Verteilung, die auch in vielen anderen Gebieten der Physik eine zentrale Rolle spielt.

Die roten Punkte waren vor dem Training auch Teil der blauen «Wolke» und unterschieden sich in keiner Art und Weise von ihnen. Man hätte beliebige andere 24 Punkte auswählen und die Matrix auf diese trainieren können!

Erstaunlicherweise entsteht durch die Aufspaltung nach der «Energie» von selbst eine Klassierung der trainierten Symbole in einzelne Gruppen. Weit weg von allen anderen Symbolen ist das Prozentzeichen %, das von seiner Struktur her am



Abb. 9: «Mona Lisa working with laptop computer» war meine Eingabe auf <http://huggingface.com> beim KI-Computer FLUX-Pro Unlimited (Nihal Gazi). Das Bild wurde in 15 Sekunden erzeugt. Diese Fälschung ist so offensichtlich, dass sie kaum jemand als Beweis für Computertechnologie im 16. Jh. benutzen würde.

wenigsten zu den anderen passt. Im Gegensatz dazu liegen **I** und **T** sehr nahe beieinander (man vergleiche ihre Struktur in Abb. 5). **N** und **W**, die miteinander verwechselt werden, liegen sogar auf demselben «Energieniveau». Das Beispiel legt nahe, wie ein neuronales Netz fähig ist, autonom Gruppierungen zu erzeugen.

Rekonstruiert man die 2400 zufälligen Vektoren mit Hilfe unserer trainierten Gewichtsmatrix mit je 3 Umläufen, erhält man das in Abb. 8 dargestellte Bild. Die vorher in einer Boltzmann-verteilten Wolke angeordneten blauen Punkte befinden sich nun alle nahe bei den «Energieniveaus» der roten Symbole. Dieses Bild macht klar, weshalb ein neuronales Netz nach einem Training beispielsweise mit Portraits von Menschen neue Portraits «kreieren» kann, die so etwas wie Mischungen von allen trainierten Portraits sind.

Damit wird auch deutlich, wie KI arbeitet: Würde man mit Hilfe eines Zufallszahlengenerators farbige Pixel erzeugen, würde mit einer äusserst kleinen Wahrscheinlichkeit (so alle Millionen Jahre einmal) ein ansprechendes menschliches Portrait erscheinen, d.h. es läge praktisch nur Chaos vor. Lässt man jedoch zufällige Pixelvektoren viele Male durch eine mit Portraits trainierte Matrix komprimieren und expandieren, werden die chaotischen Vektoren

den trainierten angeglichen wie in Abb. 8 gezeigt. Je nach der gewählten Anzahl Zyklen gleichen die erzeugten Bilder stärker oder schwächer den Vorbildern, was dann als kleinere oder grössere «Kreativität» interpretiert werden kann.

KI kann Halluzinationen entwickeln

Nachdem man gesehen hat, wie KI aus Zufallsvektoren Portraits erzeugen kann, begreift man auch, wie KI benutzt werden könnte, um UFO-Beobachtungen zu «beweisen». Man würde ein neuronales Netz mit allen zur Verfügung stehenden vermeintlichen UFO-Beobachtungen trainieren und dann an eine Kamera anschliessen, die nachts permanent den Himmel absucht. Mit geeignetem Standort würde es nicht lange dauern, bis die KI die Beobachtung von UFOs meldet und als «Beweis» recht überzeugende Bilder präsentieren würde.

Leute, die den vorangehenden Abschnitt verstanden haben, könnten durch solche Bilder nicht überzeugt werden, andere hingegen schon. Dieses Beispiel soll zeigen, wie wichtig es in Zukunft sein wird, einige Funktionalitäten der KI zu verstehen. Zur Illustration des oben skizzierten Verfahrens zum «Beweis» von UFOs zeige ich als Abb. 9 eine offensichtliche Fälschung.

Umgekehrt könnte man aus Resultaten der KI auch Eigenschaften unserer eigenen Gehirne besser verstehen. Wir wissen, dass Menschen Halluzinationen entwickeln können und die KI-Modelle geben uns eine einfache Erklärung, wie dieses Phänomen in neuronalen Netzen entstehen kann. Vielleicht müssen wir den Standpunkt einnehmen, dass Spukgeschichten, Hexen, Verschwörungstheorien, UFO-Beobachtungen und noch Vieles mehr zu erwartende Fehlleistungen menschlicher Hirne sind.

Fritz Gassmann

Literatur

Hinton G. E. 2002. Training Products of Experts by Minimizing Contrastive Divergence. *Neural Computation* 14: 1771 - 1800.

Hinton G. E. 2014. Boltzmann Machines. *Encyclopedia of Machine Learning and Data Mining*. Springer Science+Business Media New York.