GRUNDLAGEN · AUF DEUTSCH

# Was ist ein Tensor? Vom Kronecker-Produkt zu den Tensor-Netzwerken

*Wer im Netz nach einer Einführung sucht, gerät schnell in die Irre: Der Begriff meint in der Physik etwas anderes als im Quantencomputing — und deutschsprachiges Material, das den Bogen bis zu den Tensor-Netzwerken schlägt, ist kaum zu finden. Dieser Artikel versucht genau das.*

QWeb © Research —Lesezeit ~12 Min.

Vor einiger Zeit habe ich nach einer guten deutschsprachigen Einführung in die Tensor-Mathematik gesucht — von der Frage „Was ist eigentlich ein Tensor?“ über das Tensorprodukt bis hin zu den Tensor-Netzwerken, die heute in der Quantensimulation und im maschinellen Lernen eine wachsende Rolle spielen. Das Ergebnis war ernüchternd. Die erste Trefferseite führte prominent zu einer Wünschelrute; der sogenannte Biotensor ist ein esoterisches Pendel-Instrument und hat mit Mathematik nichts zu tun. Die seriösen Quellen wiederum meinten überwiegend etwas ganz anderes als das, was ich suchte. Und für Tensor-Netzwerke gibt es auf Deutsch praktisch nichts Zusammenhängendes. Dieser Artikel soll die Lücke füllen.

## 01 EIN WORT, ZWEI BEDEUTUNGEN

Die erste Hürde ist keine mathematische, sondern eine sprachliche. Das Wort „Tensor“ bezeichnet in zwei Gebieten zwei durchaus verschiedene Dinge, und wer das nicht weiß, sucht an der falschen Stelle.

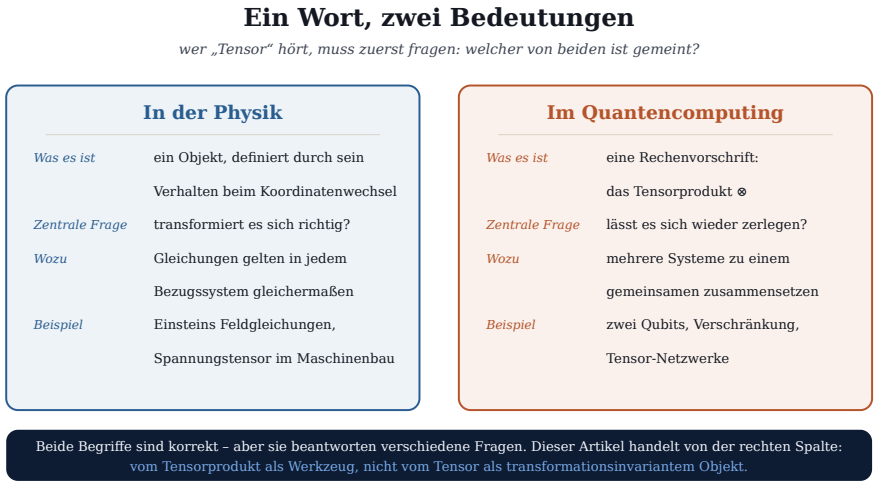


**Abbildung 1.** *Dieselbe Vokabel, zwei Gebiete, zwei Fragen. In der Physik geht es um Transformationsverhalten, im Quantencomputing um eine Rechenvorschrift und ihre Umkehrbarkeit.*

In der **Physik** — besonders in Relativitätstheorie und Kontinuumsmechanik — ist ein Tensor ein Objekt, das durch sein *Transformationsverhalten* definiert ist. Wechselt man das Koordinatensystem, so ändern sich seine Komponenten nach einer festen Regel, und zwar gerade so, dass die durch ihn ausgedrückte physikalische Aussage unverändert bleibt. Ein beliebiges Zahlenschema ist deshalb noch lange kein Tensor. Erst wenn es sich richtig mitdreht, ist es einer. Das ist der Grund, warum Einsteins Feldgleichungen in jedem Bezugssystem gleichermaßen gelten.

Im **Quantencomputing** dagegen ist mit „Tensor“ fast immer das **Tensorprodukt** gemeint — eine ganz konkrete Rechenvorschrift, auch Kronecker-Produkt genannt. Vom Transformationsverhalten ist dabei nirgends die Rede. Die Frage lautet hier nicht „transformiert es sich richtig?“, sondern „lässt es sich wieder zerlegen?“.

Beide Begriffe sind korrekt. Sie beantworten nur verschiedene Fragen. Der Rest dieses Artikels handelt ausschließlich vom zweiten.

---

## 02 DIE STUFEN EINES TENSORS

Bevor es ans Rechnen geht, noch eine Vokabel, die ständig auftaucht: die **Stufe** eines Tensors, auch Ordnung genannt. Sie ist erfreulich einfach definiert — sie ist schlicht die *Anzahl der Indizes*, die man braucht, um ein einzelnes Element zu benennen.

### Die Stufen eines Tensors

*die Stufe ist einfach die Anzahl der Indizes – alles andere folgt daraus*

| Stufe | Indizes | Name | Komponenten im 3D-Raum | Beispiel |
|---|---|---|---|---|
| 0 |  | **Skalar** | $3^0 = 1$ | Temperatur, Masse, Dichte |
| 1 |  | **Vektor** | $3^1 = 3$ | Kraft, Geschwindigkeit, E-Feld |
| 2 |  | **Matrix** | $3^2 = 9$ | Spannungs-, Trägheitstensor |
| 3 |  | — | $3^3 = 27$ | piezoelektrischer Tensor |
| 4 |  | — | $3^4 = 81$ | Elastizitätstensor |

**Nicht verwechseln: Stufe ist nicht Rang**

Eine 2×2-Matrix hat immer die Stufe 2 – sie hat zwei Indizes. Ihr Matrixrang dagegen kann 1 oder 2 sein. Zwei verschiedene Begriffe, die im Deutschen beide gern „Rang“ genannt werden.

**Abbildung 2.** *Die Stufe zählt die Indizes — und damit auch die Beine im Diagramm. Die Komponentenzahl wächst mit jeder Stufe um den Faktor der Raumdimension. Der Warnhinweis unten ist kein Detail: Stufe und Matrixrang sind verschiedene Begriffe.*

Ein **Skalar** hat die Stufe 0: eine einzelne Zahl, kein Index nötig. Temperatur, Masse und Dichte sind solche Größen. Ein **Vektor** hat die Stufe 1 und braucht einen Index — Kraft, Geschwindigkeit, elektrisches Feld. Eine **Matrix** hat die Stufe 2 und zwei Indizes; hierher gehören der Spannungs- und der Trägheitstensor.

Ab der **Stufe 3** hören die vertrauten Namen auf, die Objekte selbst aber nicht. Der piezoelektrische Tensor etwa verknüpft ein elektrisches Feld mit einer mechanischen Verzerrung und braucht dafür drei Indizes. Und die **Stufe 4** begegnet jedem, der schon einmal eine Finite-Elemente-Simulation gerechnet hat: Der Elastizitätstensor verbindet den Spannungs- mit dem Verzerrungstensor, also zwei Objekte der Stufe 2 miteinander — und braucht dafür 2 + 2 = 4 Indizes. In der Relativitätstheorie hat auch der Riemannsche Krümmungstensor die Stufe 4.

Die Komponentenzahl wächst dabei rasch. Im dreidimensionalen Raum sind es 1, 3, 9, 27 und 81 Werte; in der vierdimensionalen Raumzeit 1, 4, 16, 64 und 256. Auch hier also wieder ein exponentielles Wachstum — ein Motiv, das uns in diesem Artikel noch mehrfach begegnen wird.

> EINE FALLE IN DER DEUTSCHEN TERMINOLOGIE
>
> Die Stufe wird oft auch „Rang“ genannt — und genau da lauert eine Verwechslung. Der **Matrixrang** aus der linearen Algebra ist etwas völlig anderes: Er misst, wie viele Zeilen einer Matrix linear unabhängig sind. Eine 2×2-Matrix hat *immer* die Stufe 2, ihr Matrixrang kann aber 1 oder 2 betragen. Dieser Unterschied ist gleich in Abschnitt 04 wichtig: Dort entscheidet ausgerechnet der Matrixrang darüber, ob ein Zustand verschränkt ist.

## 03 DAS TENSORPRODUKT, KONKRET GERECHNET

Die Vorschrift ist erfreulich schlicht: **Jeder Eintrag des ersten Vektors wird mit dem ganzen zweiten Vektor multipliziert**, und die Ergebnisse werden hintereinandergehängt.


**Abbildung 3.** *Das Tensorprodukt zweier Vektoren, vollständig durchgerechnet. Rechts dieselben vier Zahlen als Matrix angeordnet — das äußere Produkt.*

Aus zwei Zahlen und zwei Zahlen werden also vier. Allgemein: aus m und n Einträgen werden m · n. Dieselben vier Produkte lassen sich übrigens auch als 2×2-Matrix anordnen — das nennt man dann das *äußere Produkt*. Es sind dieselben Zahlen in anderer Anordnung; verwirrend wird es nur, wenn man die beiden Formate verwechselt.

Nun das Entscheidende. Ein Qubit wird durch zwei Zahlen beschrieben. Zwei Qubits brauchen 2 · 2 = 4 Zahlen, drei Qubits 8, zehn Qubits bereits 1.024. Bei n Qubits sind es $\mathbf{2^n}$. Das ist keine Redewendung und keine Metapher, sondern schlicht die Dimension des Raumes, in dem der Zustand lebt.

> WARUM DAS EIN PROBLEM IST
>
> Bei 50 Qubits sind das rund $10^{15}$ Zahlen — mehr, als ein großer Rechner speichern kann. Bei 300 Qubits etwa $10^{90}$, also mehr Zahlen, als es Atome im beobachtbaren Universum gibt. Wer ein Quantensystem klassisch simulieren will, stößt hier an eine Wand, die sich nicht durch schnellere Hardware verschieben lässt. In der Datenanalyse heißt dasselbe Phänomen **Fluch der Dimensionalität**.

---

## 04 WANN LÄSST SICH DAS PRODUKT WIEDER AUFLÖSEN?

Das Tensorprodukt setzt zwei Systeme zu einem zusammen. Damit stellt sich sofort die Umkehrfrage: Lässt sich ein zusammengesetzter Zustand wieder in seine Teile zerlegen?

Manchmal ja. Ein Zwei-Qubit-Zustand heißt **separabel**, wenn er sich als A ⊗ B schreiben lässt — dann besitzt jedes der beiden Qubits einen eigenen, wohldefinierten Zustand. Manchmal aber nein, und dann spricht man von **Verschränkung**. Der berühmte Bell-Zustand ist das Standardbeispiel: Es existiert nachweislich kein Paar von Vektoren, dessen Tensorprodukt ihn ergeben würde.

Das ist eine bemerkenswert nüchterne Beschreibung eines Phänomens, das gern geheimnisvoll dargestellt wird. **Verschränkung ist mathematisch nichts anderes als das Scheitern der Tensorfaktorisierung.** Schreibt man die vier Zahlen als 2×2-Matrix, lässt sich das sogar handfest prüfen: Separabel bedeutet, dass diese Matrix den *Matrixrang* 1 hat — im Sinne von Abschnitt 02, nicht im Sinne der Stufe. Beim Bell-Zustand hat sie den Matrixrang 2 — und deshalb gibt es kein solches Vektorpaar.

Dieselbe Struktur begegnet einem übrigens auch in der Wahrscheinlichkeitsrechnung: Die gemeinsame Verteilung *unabhängiger* Zufallsvariablen ist das Tensorprodukt ihrer Einzelverteilungen, und die Abweichung davon misst gerade die Abhängigkeit. Nicht faktorisierbar heißt korreliert — in der Statistik wie in der Quantenmechanik.

---

## 05 EINE SPRACHE AUS KÄSTEN UND LINIEN

Wer mit vielen Tensoren gleichzeitig rechnet, ersäuft rasch in Indizes. Roger Penrose hat dafür eine grafische Notation eingeführt, die heute in der Tensor-Netzwerk-Literatur Standard ist — und die man in fünf Minuten lernt.

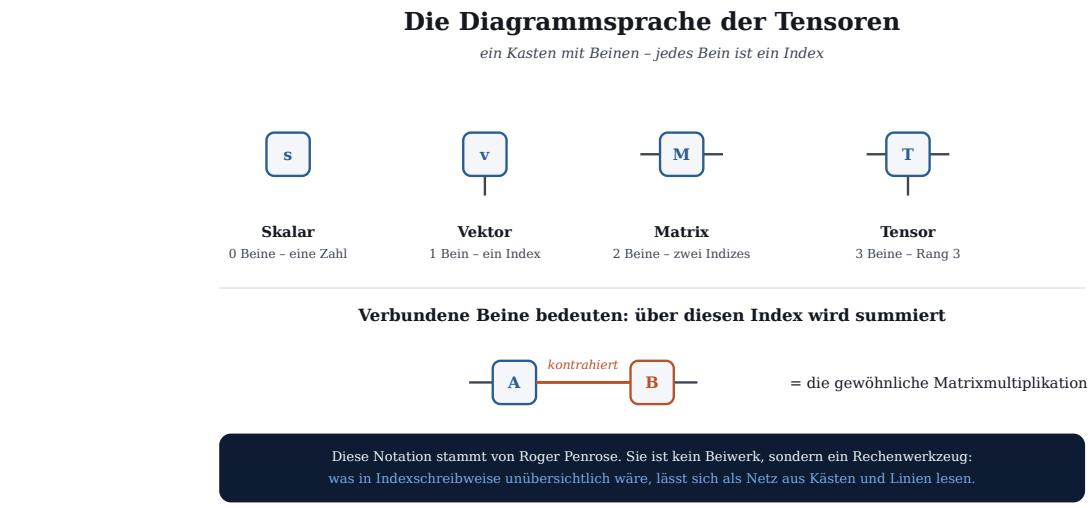


**Abbildung 4.** *Die grafische Notation nach Penrose: Jeder Index wird zu einem Bein, verbundene Beine bedeuten Summation. Die Matrixmultiplikation ist damit ein Bild statt einer Formel.*

Die Regel ist denkbar einfach: Ein Tensor wird als Kasten gezeichnet, und **jeder Index wird zu einem Bein**. Ein Skalar hat kein Bein, ein Vektor eines, eine Matrix zwei, ein Tensor dritter Stufe drei. Verbindet man zwei Beine miteinander, so bedeutet das: über diesen Index wird summiert. Zwei Matrizen mit einem verbundenen Bein sind nichts anderes als die gewöhnliche Matrixmultiplikation.

Das klingt nach Spielerei, ist aber ein echtes Rechenwerkzeug. Ausdrücke, die in Indexschreibweise unübersichtlich wären, lassen sich als Netz aus Kästen und Linien auf einen Blick erfassen — und genau daher hat die ganze Methode ihren Namen.

## 06 TENSOR-NETZWERKE: DEN FLUCH UMGEHEN

Jetzt lässt sich die eigentliche Idee formulieren. Ein Quantenzustand aus n Teilchen ist ein einziger riesiger Tensor mit n Beinen und $2^n$ Einträgen. Der ist nicht speicherbar. Aber muss man ihn überhaupt vollständig speichern?

Die Antwort der Tensor-Netzwerke lautet: häufig nicht. Statt eines großen Tensors schreibt man eine **Kette vieler kleiner**, die jeweils nur mit ihren Nachbarn verbunden sind. Diese Konstruktion heißt **Matrixproduktzustand** (englisch Matrix Product State, MPS) oder auch Tensorzug.

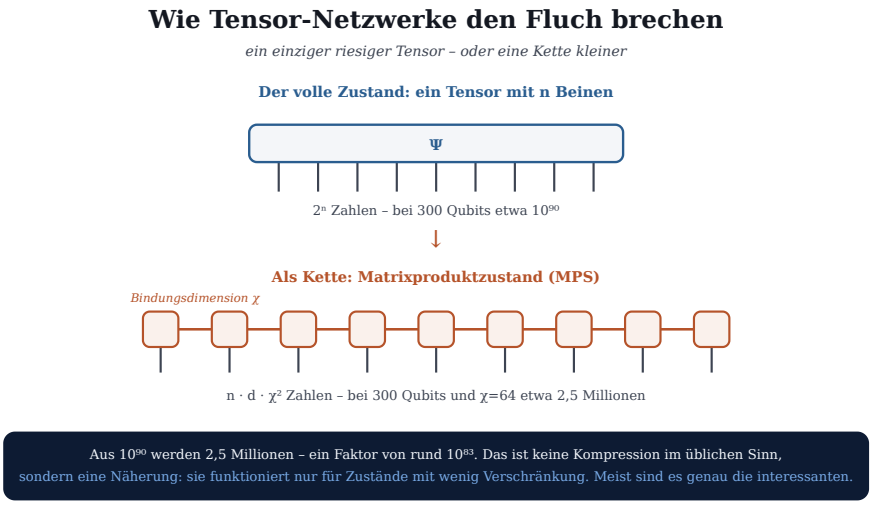


**Abbildung 5.** *Oben der volle Zustand als ein Tensor mit exponentiell vielen Einträgen, unten dieselbe Information als Kette kleiner Tensoren. Die Bindungsdimension χ steuert, wie genau die Näherung ist.*

Der Gewinn ist dramatisch. Die Zahl der benötigten Werte wächst nicht mehr exponentiell mit n, sondern nur noch linear — sie beträgt etwa $n \cdot d \cdot \chi^2$, wobei $\chi$ die sogenannte Bindungsdimension ist, also die Dicke der Verbindungen zwischen den Kettengliedern.

| | |
|---|---|
| `50 Qubits` | voller Zustand ~$10^{15}$ Zahlen — als MPS mit $\chi$=100 rund eine Million. |
| `300 Qubits` | voller Zustand ~$10^{90}$ Zahlen — als MPS mit $\chi$=64 etwa 2,5 Millionen. Ein Faktor von rund $10^{83}$. |
| `Der Preis` | Es ist keine verlustfreie Kompression, sondern eine Näherung. Sie ist gut, solange $\chi$ ausreicht — und schlecht, sobald nicht. |

## 07 WARUM DAS ÜBERHAUPT FUNKTIONIERT

Hier liegt die interessanteste Frage des ganzen Gebiets, und sie blieb lange offen. Der zugrunde liegende Algorithmus, die **Dichtematrix-Renormierungsgruppe** (DMRG), wurde 1992 von Steven White entwickelt und lieferte verblüffend genaue Ergebnisse für eindimensionale Quantensysteme. Warum er so gut funktionierte, verstand zunächst niemand. Erst 1995 zeigten Stellan Östlund und Stefan Rommer, dass DMRG die Quantenzustände im Kern durch eine MPS-Darstellung komprimiert.

Damit war ein Teil des Rätsels gelöst, aber nicht das Ganze: Warum lassen sich offenbar *alle relevanten* Quantensysteme so darstellen? Die Antwort, die sich in den 2000er Jahren herausbildete, ist elegant. Die Wellenfunktionen, die in der Natur tatsächlich vorkommen, sind nicht völlig zufällig. Ihre Verschränkung folgt einem **Flächengesetz**: Sie wächst nicht mit dem Volumen eines Teilgebiets, sondern nur mit dessen Oberfläche. Und genau solche Zustände sind komprimierbar.

Das ergibt eine schöne Pointe. Tensor-Netzwerke funktionieren nicht deshalb, weil Verschränkung unwichtig wäre — sondern weil sie in physikalisch realistischen Zuständen *begrenzt* ist. Zustände mit sehr viel Verschränkung bleiben hart. Es sind allerdings meist nicht die, die man simulieren möchte.

## 08 WO DAS HEUTE EINE ROLLE SPIELT

Das Verfahren hat längst die Physik verlassen. Vier Anwendungsfelder sind erwähnenswert:

| | |
|---|---|
| Quantensimulation | Der ursprüngliche Zweck: Grundzustände eindimensionaler Spinsysteme, Quantenchemie, Materialforschung. Die Verallgemeinerung auf zwei Dimensionen heißt PEPS. |
| Maschinelles Lernen | Modellkompression und überwachtes Lernen — dieselbe Mathematik, angewandt auf hochdimensionale Daten. Google veröffentlichte 2019 die Bibliothek TensorNetwork. |
| Numerische Simulation | Ingenieurprobleme, Finite-Elemente-Verfahren, Darstellung von Funktionen als Tensor-Netzwerk. |
| Quantenähnliche Verfahren | Klassische Algorithmen, die von der Quantenmathematik inspiriert sind, aber auf gewöhnlicher Hardware laufen — im Finanzbereich derzeit der praktisch nutzbringendste Teil des ganzen Feldes. |

Der letzte Punkt führt zu einer Frage, die derzeit ernsthaft diskutiert wird: Wenn klassische Rechner mit Tensor-Netzwerken so weit kommen — wo genau bleibt dann noch Raum für einen echten Quantenvorteil? Eine abschließende Antwort gibt es nicht. Sicher ist nur, dass die Messlatte für Quantenhardware durch diese klassischen Methoden immer wieder höher gelegt wurde.

## 09 WAS ZU BEHALTEN WÄRE

Ein Tensor im Sinne des Quantencomputings ist kein geheimnisvolles Objekt, sondern das Ergebnis einer einfachen Multiplikationsvorschrift. Das Tensorprodukt setzt Systeme zusammen, und dabei wächst die Beschreibung exponentiell — $2^n$ Zahlen für n Qubits.

Ob sich ein zusammengesetzter Zustand wieder zerlegen lässt, ist genau die Frage nach der Verschränkung. Und Tensor-Netzwerke sind die Antwort darauf, was man tut, wenn die exponentielle Beschreibung zu groß wird: Man ersetzt einen unbezahlbaren Tensor durch eine Kette bezahlbarer — eine Näherung, die funktioniert, weil die Verschränkung in realistischen Systemen begrenzt ist.

Wer tiefer einsteigen möchte: Die lineare Algebra dahinter ist in den Vorlesungsvideos von Prof. Edmund Weitz (HAW Hamburg) auf Deutsch hervorragend aufbereitet — insbesondere Matrizenmultiplikation, orthogonale Abbildungen und die Singulärwertzerlegung, die der Schmidt-Zerlegung zugrunde liegt. Für die Tensor-Netzwerke selbst führt dagegen kaum ein Weg an englischsprachiger Literatur vorbei; der etablierte Einstieg ist die Übersichtsarbeit von Roman Orús.

---

QWeb © Research · Reihe Grundlagen

Abbildungen: die zwei Bedeutungen des Wortes Tensor; die Stufen 0 bis 4; das Tensorprodukt Schritt für Schritt; die Diagrammsprache; MPS gegenüber dem vollen Zustand.