# Alles wird zu Zahlen (Tensoren)

Eine Zahl, eine Reihe, eine Tabelle — und schon hast du einen Tensor: die Datenstruktur hinter jeder KI (Skalar, Vektor, Matrix).

**Grundlagen · Kapitel 5 von 50 · ca. 10 Min. · 100 XP**

Kapitel online

**Einfach gesagt:** Ob Foto, Tabelle oder Satz — für den Computer wird daraus immer nur eine Zahlenliste, die er Tensor nennt und mit der er rechnet.

Im letzten Kapitel hast du gesehen: Eine Maschine lernt durch **Raten, Fehler messen, Nachjustieren** — und dafür kann sie nur eines: **rechnen**. Aber womit rechnet sie, wenn du ihr ein Katzenfoto, eine Excel-Tabelle oder einen Satz gibst? Mit **Tensoren** — den Zahlenpaketen, in die ALLES verwandelt wird, bevor KI damit arbeiten kann. Wer Tensoren versteht, hat den Generalschlüssel für jedes weitere Kapitel.

**HerrKI-Tipp:** Nimm dir Zeit für jedes Beispiel. Tensoren sind die Basis von allem, was in KI passiert!

**Was du hier lernst**

- Was sind Tensoren? (Spoiler: Mehrdimensionale Arrays!)
- Welche Dimensionen (Ranks) gibt es?
- Wie werden echte Daten in Tensoren umgewandelt?
- Warum können neuronale Netze nur mit Zahlen arbeiten?

**Für den Unterricht**

**Einstieg (2 Min):** Kapitel gemeinsam lesen, Alltagsbeispiel aufgreifen.

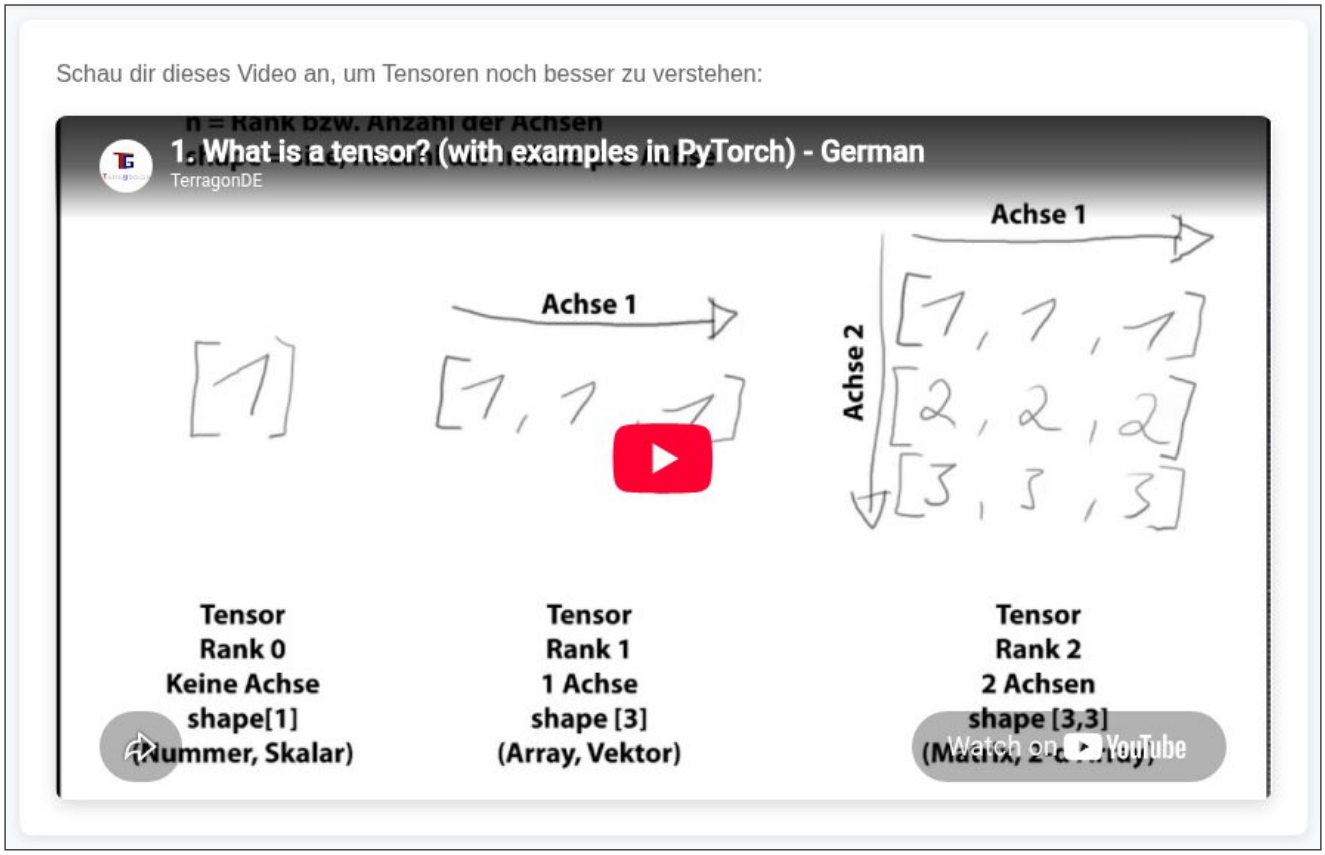
**Erkunden am Gerät (6 Min):** die interaktiven Module auf herrki.de/010_was_sind_tensoren ausprobieren.

**Sichern mit Quiz (2 Min):** „Kurzer Check" besprechen, „Das nimmst du mit" festhalten.

# Video-Erklärung von HerrKI

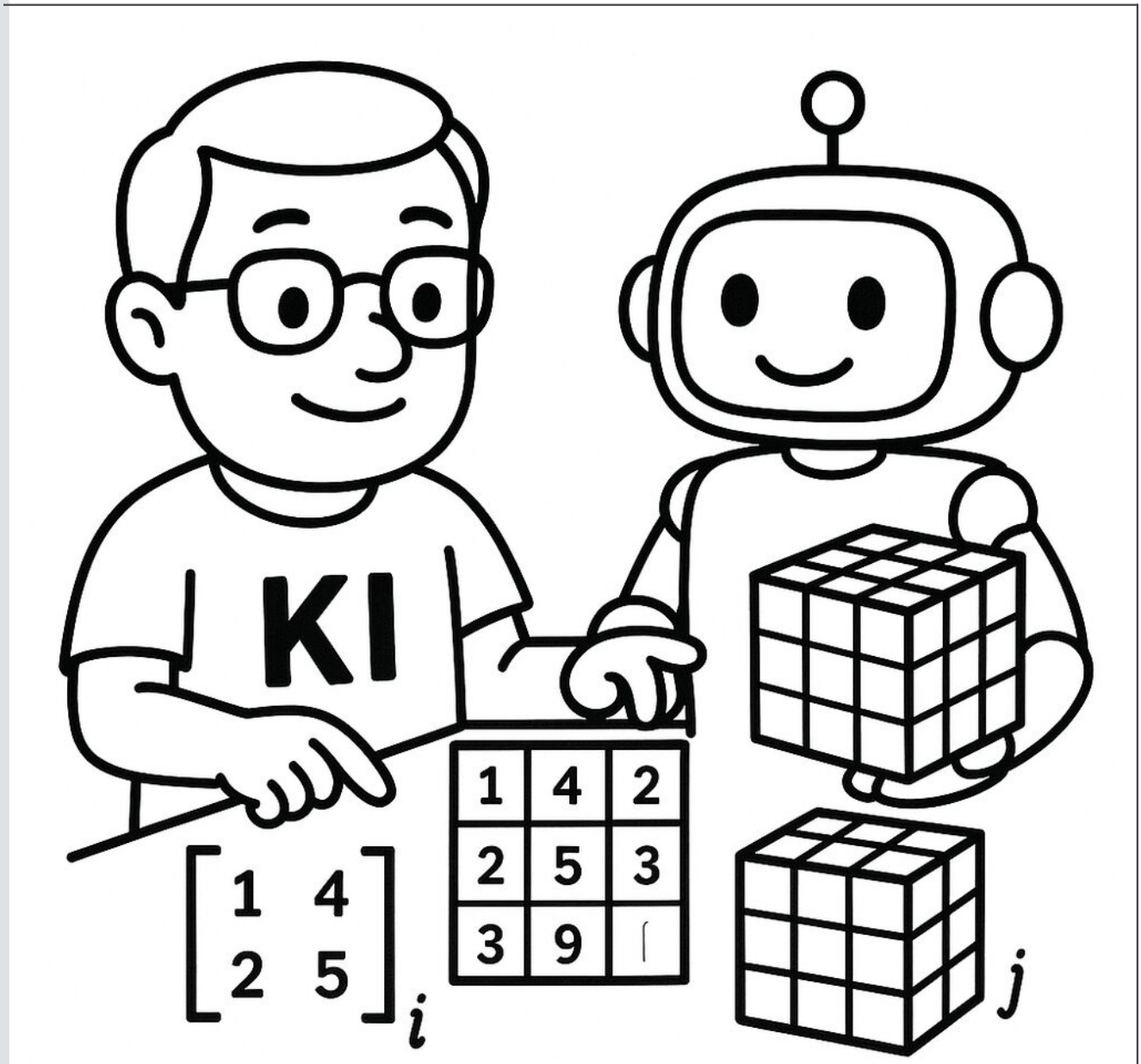
Video ansehen: herrki.de/010_was_sind_tensoren#abschnitt-2

Direkt zum Modul




*Video auf der Kapitelseite – am Gerät ansehen*

# Runter zum Tensor: Was Tensoren sind




**Stell dir vor:** Du zeigst deinem Computer ein Foto von deinem Hund und möchtest, dass er lernt, Hunde zu erkennen. Oder du gibst ihm einen Text und er soll die Stimmung verstehen. **Aber hier ist das Problem:** Computer können weder Bilder "sehen" noch Texte "lesen" - sie verstehen nur eines: Zahlen!

Hier kommen **Tensoren** ins Spiel! Ein Tensor ist nichts anderes als eine **strukturierte Sammlung von Zahlen** - du kannst es dir vorstellen wie:

- Eine **Excel-Tabelle** voller Zahlen (2D)

- Ein **Stapel von Excel-Tabellen** übereinander (3D)
- Sogar **mehrere Stapel nebeneinander** (4D, 5D, ...)

**Das Wichtigste zum Verstehen:**

**Tensoren = Mehrdimensionale Arrays von Zahlen**

Egal was du ins neuronale Netz steckst - Bilder, Text, Audio, Videos - **alles wird erst in Tensoren (Zahlen) umgewandelt**. Nur dann kann das Netz damit rechnen und lernen!

**Warum ist das so wichtig?** Weil neuronale Netze im Kern nichts anderes machen als Mathematik mit Zahlen: Multiplikationen, Additionen, Aktivierungsfunktionen. Ohne Tensoren - keine KI! In den nächsten Kapiteln zeige ich dir genau, wie du Bilder, Excel-Daten, Texte und mehr in Tensoren umwandelst.

**HerrKI erklärt's einfach:** Neuronale Netze können **nur mit Zahlen arbeiten**! Egal ob Bilder, Text, Audio oder Excel-Tabellen - ich zeige dir jetzt, wie alles in Tensoren (Zahlen) umgewandelt wird. Das ist die **fundamentale Grundlage** für alles, was in Machine Learning passiert!

**Die verschiedenen Tensor-Dimensionen (Ranks):**

RANK 0

**Skalar (0D)**

Eine einzelne Zahl

42

Shape: ()

Beispiel: Temperatur, Preis, Alter

RANK 1

**Vektor (1D)**

Eine Liste von Zahlen

| | | | |
|---|---|---|---|
| 1 | 5 | 3 | 8 |

Shape: (4)

Beispiel: Zeitreihe, Audio-Sample

RANK 2

**Matrix (2D)**

Tabelle aus Zeilen und Spalten

| | |
|---|---|
| 1 | 2 |
| 4 | 5 |
| 7 | 8 |

Shape: (3, 3)

Beispiel: Excel-Tabelle, Graustufenbild

RANK 3+

**3D+ Tensor**

Mehrere gestapelte Matrizen

R

R

R

R

G

G

G

G

Shape: (2, 2, 3)

Beispiel: Farbbild (RGB), Video

**Wie werden echte Daten in Tensoren umgewandelt?**

**Einzelne Zahl**

Temperatur: 23.5°C

↓

```
tf.tensor(23.5)
// Shape: ()
```

**→ Rank 0 Tensor (Skalar)**

**Liste von Zahlen**

Preise: [10, 25, 15, 30]

↓

```
tf.tensor([10, 25, 15, 30])
// Shape: (4)
```

**→ Rank 1 Tensor (Vektor)**

### Tabelle (Excel)

Wohnungsdaten: Größe, Zimmer, Preis

↓

```
tf.tensor2d([
  [50, 2, 800],
  [70, 3, 1200],
  [90, 4, 1500]
])
// Shape: (3, 3)
```

**→ Rank 2 Tensor (Matrix)**

### Graustufenbild

28x28 Pixel, Werte 0-255

↓

```
tf.tensor2d(pixelData)
// Shape: (28, 28)
```

**→ Rank 2 Tensor**

### Farbbild (RGB)

64x64 Pixel, RGB-Kanäle

↓

```
tf.tensor3d(imageData)
// Shape: (64, 64, 3)
// 3 = R, G, B
```

**→ Rank 3 Tensor**

### Text / Wörter

"Hallo Welt"

↓

```
// Word Embeddings
tf.tensor([0.2, 0.8, -0.3, 0.5])
// Shape: (4)
```

**→ Rank 1 Tensor (Vektor)**

**Video**

30 Frames, 128x128 RGB

↓

```
tf.tensor4d(videoData)
// Shape: (30, 128, 128, 3)
// 30 Frames
```

**→ Rank 4 Tensor**

**Batch (Mehrere Beispiele)**

10 Wohnungen als Batch

↓

```
tf.tensor2d([
  [50, 2, 800],
  [70, 3, 1200],
  // ... 8 weitere
])
// Shape: (10, 3)
```

**→ Rank 2 Tensor**

**„Müssen die Zahlen nicht zwischen 0 und 1 liegen?"**
Nein! Ein Tensor darf **jede Zahl** enthalten — 23.5 °C, 350 m², 255 als Farbwert: alles völlig korrekt. Das *Umrechnen auf 0–1* (die **Normalisierung**) ist kein Teil des Tensors, sondern ein **Trainings-Trick**: Wenn ein neuronales Netz später aus den Zahlen lernen soll, klappt das stabiler und schneller, wenn alle Werte ähnlich groß sind. Genau das lernst du in „Bilder als Tensoren" (Pixel ÷ 255) und „Excel/CSV → Tensoren" — hier schauen wir uns die Tensoren erst einmal mit ihren echten, vollwertigen Zahlen an.

**Wie kommen Tensoren ins Neuronale Netz?**

Das neuronale Netz erwartet Tensoren in einer bestimmten **Shape**:

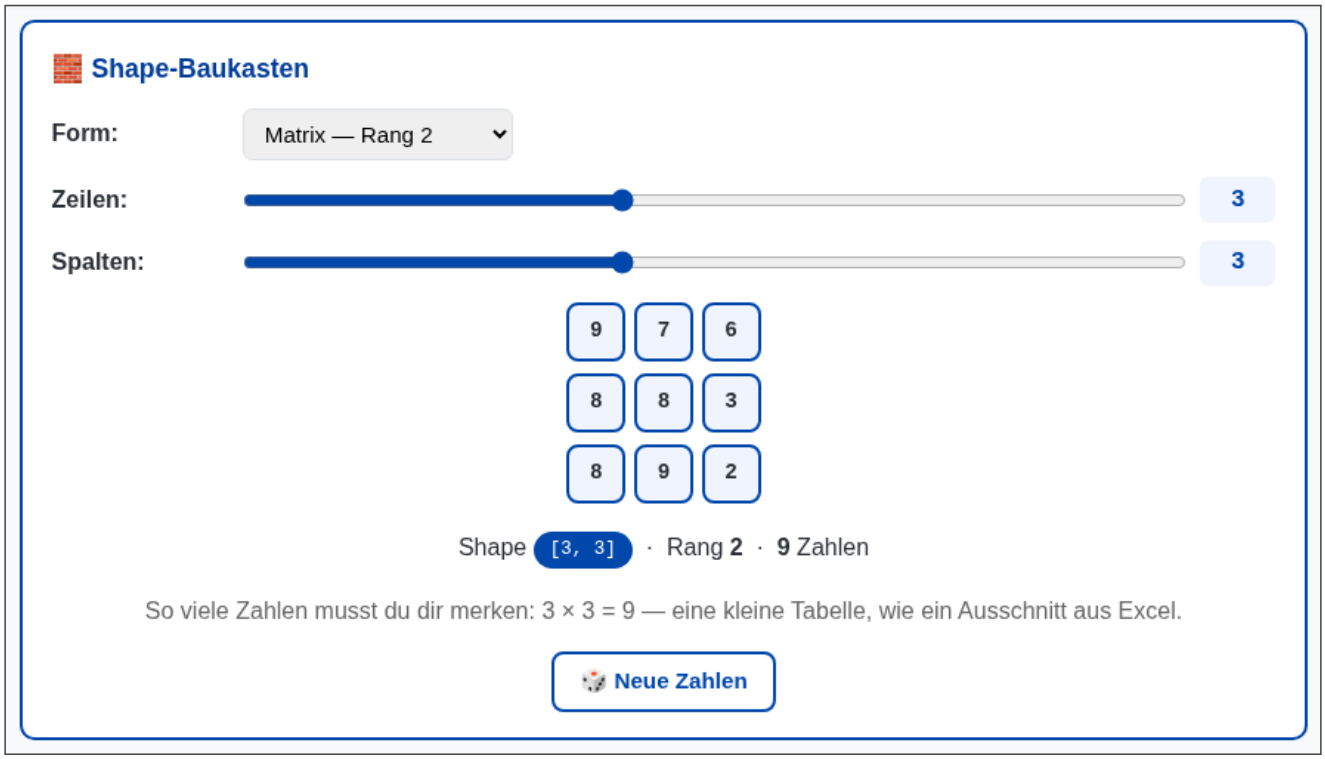
- **Eingabe-Layer:** Die Shape muss mit der Anzahl der Features übereinstimmen
- **Beispiel:** Wohnung mit 3 Features (Größe, Zimmer, Stock) → `inputShape: [3]`
- **Batch Processing:** Mehrere Beispiele werden gleichzeitig verarbeitet → Shape: `(batch_size, features)`
- **Ausgabe:** Das Netz gibt einen Tensor in der Form zurück, die du definiert hast (z.B. 1 Zahl für Regression, 10 Zahlen für 10 Klassen)

# Tensoren zum Anfassen

Im Video und in den vier Karten oben hast du Skalar, Vektor, Matrix und 3D-Tensor gesehen — jetzt baust du sie selbst. Wähl eine Form, zieh an den Reglern und beobachte, wie sich **Shape**, **Rang** und die **Anzahl der Zahlen** mitverändern. Genau diese drei Angaben liest ein KI-Entwickler als Erstes, wenn er neue Daten vor sich hat.

Am Gerät ausprobieren: herrki.de/010_was_sind_tensoren#abschnitt-4

Direkt zum Modul




*Interaktives Modul – am Gerät ausprobieren*

Zwei Dinge fallen dabei schnell auf: Die Shape wird nur *eine Klammer länger*, wenn eine Achse dazukommt — die Anzahl der Zahlen dagegen wird *multipliziert*. Und die Reihenfolge in der Shape ist kein Zufall: erst Zeilen, dann Spalten, bei 3D davor noch der Stapel.

# Zurück zur Anwendung

**Und zurück zur Anwendung:** Wenn dein Handy dein Gesicht erkennt, passiert exakt das hier — Kamera-Bild → Tensor (Höhe × Breite × 3) → Netz rechnet → eine Zahl („du bist es: 0.98“). Kein Hexenwerk, nur Shapes und Mathe. Bei der Wohnungspreis-Schätzung ist es derselbe Ablauf, nur kleiner: [Größe, Zimmer, Stock] → Shape `[3]` → eine Zahl in Euro. Und DESHALB lohnt sich der Aufwand mit Skalar, Vektor, Matrix und Rank 3+: Wer die Shape lesen kann, sieht jeder KI-Anwendung an, was da eigentlich hineinfließt und was herauskommt.

**Das war das wichtigste Kapitel des ganzen Kurses.** Ab jetzt wirst du in JEDEM Kapitel denken: „Okay — welche Shape hat das gerade?“ Genau dieser Blick unterscheidet Anwender von Entwicklern. Weiter geht's mit dem ersten echten Datentyp: Bilder als Tensoren — und wenn du wissen willst, wie man mit Tensoren RECHNET, wartet „Tensor-Operationen“ auf dich.

**Zum Weiterlesen**

- **Zum Ausprobieren:** TensorFlow.js-Guide, *Tensors and operations* — tensorflow.org/js/guide/tensors_operations — genau die Funktionen aus diesem Kapitel (`tf.tensor`, `shape`, `dispose`) im Original
- **Die Mathematik dahinter:** Goodfellow/Bengio/Courville, *Deep Learning*, Kap. 2 „Linear Algebra“ — deeplearningbook.org/contents/linear_algebra.html — dort heißen Skalar, Vektor, Matrix und Tensor genauso, nur mit Formeln (kostenlos lesbar, auf Englisch)
- **Zum Anschauen:** 3Blue1Brown, *Essence of Linear Algebra* — 3blue1brown.com/topics/linear-algebra — falls du sehen willst, was ein Vektor *geometrisch* ist, bevor es in Kapitel 060 ans Rechnen geht

# Kurzer Check – mit Lösung

*Lösungen für Lehrkräfte – für Lernende die Online-Version nutzen.*

Drei schnelle Fragen, dann bist du durch — klick einfach deine Antwort an:

**Frage 1: Ein Tensor hat die Shape [4]. Was steckt dahinter?**

- Eine 4×4-Matrix mit 16 Zahlen
- **✔ Ein Vektor mit genau 4 Zahlen — wie [10, 25, 15, 30]**
- Ein Skalar mit dem Wert 4

> *Die Shape zählt die Einträge je Dimension: [4] heißt eine Dimension, vier Werte — also ein Vektor. Eine 4×4-Matrix hätte die Shape [4, 4].*

**Frage 2: Welche Shape hat die Matrix [[1,2,3],[4,5,6]]?**

- [3, 2] — erst Spalten, dann Zeilen
- [6] — sechs Zahlen sind sechs Zahlen
- **✔ [2, 3] — 2 Zeilen, 3 Spalten**

> *Merkregel: erst Zeilen, dann Spalten. Zwei Zeilen mit je drei Werten → [2, 3]. (Zu [6] käme man erst per flatten() — das lernst du in Kapitel Tensor-Operationen.)*

**Frage 3: Darf ein Tensor „große“ Zahlen wie 350 oder 23,5 enthalten?**

- Nein, Tensoren funktionieren nur mit Werten zwischen 0 und 1
- **✔ Ja — Tensoren dürfen beliebige Zahlen enthalten; auf 0–1 skaliert wird erst später fürs Training**
- Nur ganze Zahlen sind erlaubt

> *Genau! Ein Tensor ist nur der Zahlen-Container. Die Umrechnung auf 0–1 (Normalisierung) ist ein Trainings-Trick, den du bei Bildern (÷255) und Tabellen kennenlernst — keine Eigenschaft des Tensors.*

# Das nimmst du mit

- **Rank 0 — Skalar:** eine einzelne Zahl, Shape `()` — zum Beispiel 23,5 °C.
- **Rank 1 — Vektor:** eine Liste, Shape `(n)` — zum Beispiel `[10, 25, 15, 30]`.
- **Rank 2 — Matrix:** eine Tabelle, Shape `(rows, cols)` — erst Zeilen, dann Spalten.
- **Rank 3+:** Bilder und Videos, Shape `(height, width, channels)` — der Zahlen-Würfel.
- **Alles hat eine Shape** — und das Netz erwartet genau eine bestimmte (`inputShape: [3]` für drei Features). Die Zahlen selbst dürfen dabei beliebig groß sein; das Umrechnen auf 0–1 ist ein Trainings-Trick, keine Eigenschaft des Tensors.

### So zitierst du dieses Kapitel

HerrKI: Alles wird zu Zahlen (Tensoren). herrki.de/010_was_sind_tensoren, ein privates Hobbyprojekt von Terragon.de, abgerufen am 09.09.2026.