Rotary Club Siegen-Schloß

Rotaract Siegen

ROTARY CLUB SIEGEN-SCHLOSS · GASTVORTRAG · 20. OKTOBER 2026 · HAUS DER SIEGERLÄNDER WIRTSCHAFT

Wie rechnet ein Sprachmodell?
Die Mathematik hinter ChatGPT

Von **Backpropagation** und **Matrixmultiplikation** bis zu den Durchbrüchen, die zu ChatGPT geführt haben.

Jan Spörer · Rotaract Club Siegen · Universität St. Gallen (Finance/NLP)

Service Above Self

Inhaltsübersicht

1 **Backpropagation**
Wie ein neuronales Netz überhaupt **lernt**: aus Fehlern die Gewichte anpassen.

2 **Matrixmultiplikation**
Die eine Rechenoperation dahinter. Live-Visualisierung: **bbycroft.net/llm**.

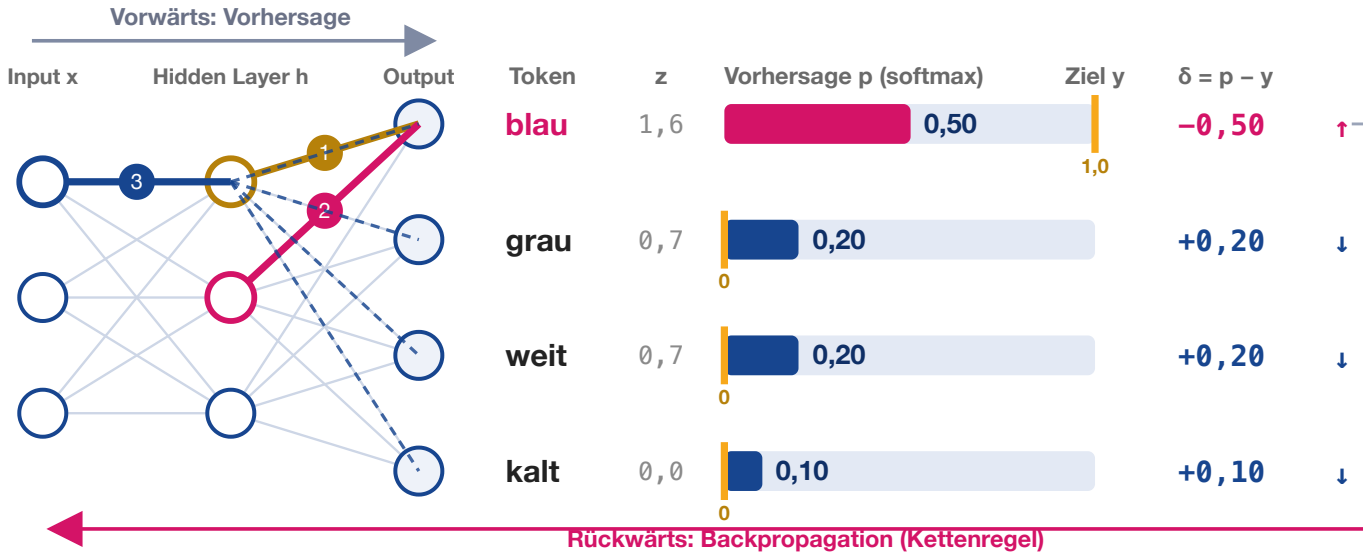
3 **Ein einfaches Netz konkret durchrechnen**
Schritt für Schritt mit echten Zahlen: von der Eingabe bis zur Ausgabe.

4 **Durchbruch-Innovationen zu ChatGPT (November 2022)**
Transformer-Architektur, **Post-Training (RLHF)** und **Geld** (Grafikkarten & Energie).

1 Backpropagation: ein Trainingsschritt konkret

Beispiel: **„Der Himmel ist ___“** — welches Token kommt als Nächstes? (Vokabular: 4 Tokens)

Der Trick: das Fehlersignal ist einfach Vorhersage – Wahrheit (p – y).



Cross-Entropy: wie falsch war es?

Nur das richtige Token zählt:

$L = -\log(0{,}50) = 0{,}69$

je näher p(blau) an 1, desto kleiner L

↻ millionenfach wiederholt → L → 0

Update-Regel (SGD, einfacher Gradientenabstieg): $w \leftarrow w - \eta \cdot \partial L/\partial w$, $\eta = 0{,}1$

① Letzte Schicht: h_1 → „blau“

$\delta = -0{,}50 \qquad h_1 = 0{,}80$

$\partial L/\partial w = \delta \cdot h_1$

$= (-0{,}50) \cdot 0{,}80 = -0{,}40$

$w: 0{,}30 - 0{,}1 \cdot (-0{,}40)$

$= 0{,}34$

② Letzte Schicht: h_2 → „blau“

$\delta = -0{,}50 \qquad h_2 = 0{,}90$

$\partial L/\partial w = \delta \cdot h_2$

$= (-0{,}50) \cdot 0{,}90 = -0{,}45$

$w: 0{,}20 - 0{,}1 \cdot (-0{,}45)$

$= 0{,}245$

③ Erste Schicht: x_1 → h_1 (Kettenregel)

$\delta_{h_1} = (\Sigma_k \delta_k \cdot w_k) \cdot \mathrm{ReLU}'(z_{h_1})$

$\Sigma_k = (-0{,}50) \cdot 0{,}30 + 0{,}20 \cdot 0{,}10$

$+ 0{,}20 \cdot 0{,}20 + 0{,}10 \cdot 0{,}10 = -0{,}08$

$\mathrm{ReLU}'(0{,}80) = 1 \rightarrow \delta_{h_1} = -0{,}08$

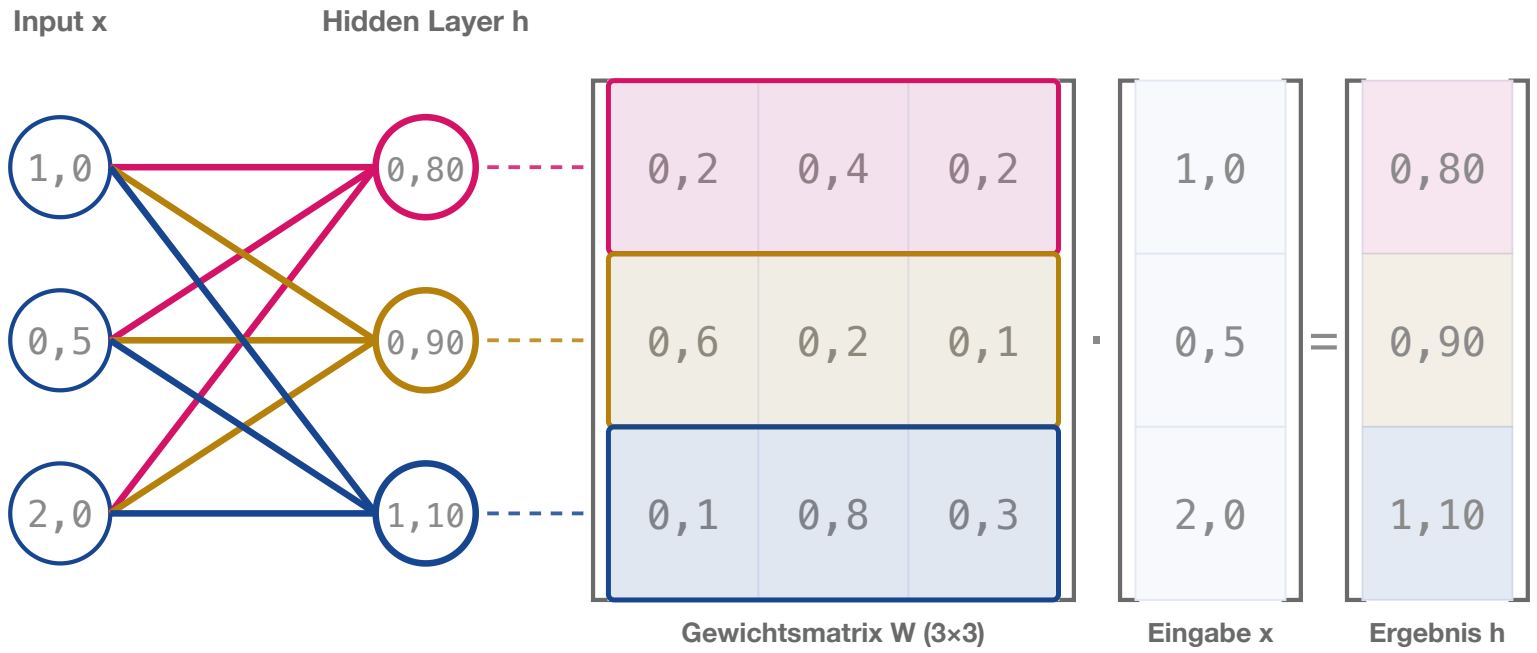
$\partial L/\partial w = \delta_{h_1} \cdot x_1 = (-0{,}08) \cdot 1{,}0 = -0{,}08$

$w: 0{,}20 - 0{,}1 \cdot (-0{,}08) = 0{,}208$

2 Matrixmultiplikation: vom Netz zur Matrix

Die **9 Verbindungen** zwischen Input und Hidden Layer sind **genau die 9 Zahlen** der Gewichtsmatrix W.

Jede Zeile von W = alle eingehenden Gewichte eines Neurons.



3 × 3 = 9 Verbindungen
= 9 Gewichte

$$h_1 = 0{,}2 \cdot 1{,}0 + 0{,}4 \cdot 0{,}5 + 0{,}2 \cdot 2{,}0 = 0{,}80$$
$$h_2 = 0{,}6 \cdot 1{,}0 + 0{,}2 \cdot 0{,}5 + 0{,}1 \cdot 2{,}0 = 0{,}90$$
$$h_3 = 0{,}1 \cdot 1{,}0 + 0{,}8 \cdot 0{,}5 + 0{,}3 \cdot 2{,}0 = 1{,}10$$

Eine ganze Schicht

$$h = \mathrm{ReLU}(W \cdot x + b)$$

① W·x ② + Bias b ③ ReLU

hier b = 0, alle h > 0 → unverändert

Warum überhaupt ReLU?

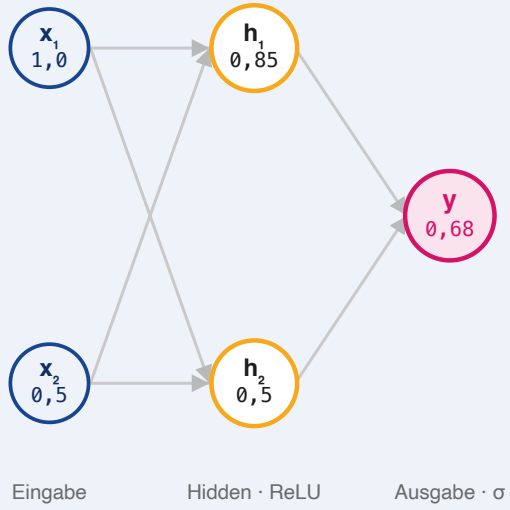
Ohne die Nichtlinearität wäre das ganze Netz nur EINE einzige große Matrix — keine Tiefe.

In GPT: Matrizen mit Millionen Einträgen — genau dafür GPUs.

Live in 3D: bbycroft.net/llm

3 Ein einfaches Netz konkret durchrechnen

- Mini-Netz: **2 Eingaben**, ein Hidden Layer mit **2 Neuronen**, **1 Ausgabe**.
- Echte Zahlen einsetzen und Schritt für Schritt: **multiplizieren, addieren, Aktivierung**.
- Am Ende steht **eine einzige Zahl** (z.B. eine Wahrscheinlichkeit).
- ChatGPT rechnet **genau so**, nur milliardenfach größer.



Pfeile = Gewichte w · f = ReLU (negativ→0) · σ quetscht auf 0…1

$h_1 = f(0{,}9 \cdot 1{,}0 - 0{,}3 \cdot 0{,}5 + 0{,}1)$
$= f(0{,}85) = \mathbf{0{,}85}$

$h_2 = f(0{,}2 \cdot 1{,}0 + 0{,}8 \cdot 0{,}5 - 0{,}1)$
$= f(0{,}5) = \mathbf{0{,}5}$

$y = \sigma(1{,}0 \cdot 0{,}85 - 0{,}6 \cdot 0{,}5 + 0{,}2)$
$= \sigma(0{,}75) = \mathbf{0{,}68}$

→ ≈ **68%** Wahrscheinlichkeit

Kleines Netz, echte Zahlen: aus x_1, x_2 wird über zwei Hidden-Neuronen eine Wahrscheinlichkeit. ChatGPT rechnet genauso, nur milliardenfach größer.

4 Durchbruch-Innovationen zu ChatGPT (November 2022)

- **Transformer-Architektur (2017):** die „Attention" lässt jedes Wort den ganzen Kontext gewichten. Ermöglicht lange Texte und paralleles, schnelles Training.
- **Post-Training / RLHF:** Reinforcement Learning from Human Feedback macht aus einem reinen „Wort-Vorhersager" einen hilfreichen, höflichen Assistenten.
- **Geld & Rechenleistung:** GPT-3, die Basis von ChatGPT, lief auf rund 10.000 GPUs (Azure-Supercomputer, NVIDIA V100); heutige Spitzenmodelle nutzen zehntausende. Enormer Energie- und Datenaufwand, Skalierung ist der eigentliche Hebel.

Der Punkt: Nicht eine einzige Idee, sondern das Zusammenspiel: Architektur × Training × schiere Rechenleistung.

Zum Selber-Entdecken

- **bbycroft.net/llm**: mit Abstand die beste interaktive 3D-Visualisierung eines echten Modells (jeder Rechenschritt zum Anfassen).
- **ig.ft.com/generative-ai**: Financial Times, sehr anschauliche journalistische Erklärung „wie generative KI funktioniert".
- **machinelearningmastery.com**: „The Transformer Attention Mechanism", kompakter technischer Einstieg zum Nachlesen.
- **Sonstige spannende Konzepte**: Model Context Protocol (MCP), Loop Engineering

Rotary Club Siegen-Schloß

Rotaract Siegen

Fragen & Diskussion.

Ein Beitrag aus dem **Rotaract Club Siegen**.

Jan Spörer · Präsident 2026/27 Rotaract Club Siegen · jan.spoerer@whu.edu · +49 171 5395666

Service Above Self