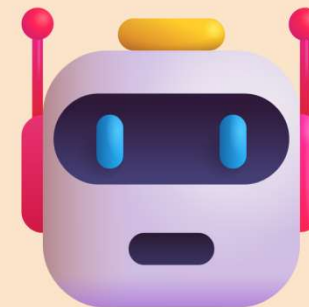


# AI Insights

**LOKALE** Sprachmodelle  
LLMs



## Worum geht's?

💡 Lokale Sprachmodelle  
...und wie ich sie auf meiner  
(eigenen) Hardware laufen lasse



## Was sind diese LLMs?

**Achtung:** es wird *technisch*...  
...wir gehen gleich mal in die *Vollen*

Ein **LLM** ist ein **KI-Modell basierend auf der Transformer Architektur**, das mit großen Mengen an Text trainiert wird.

→ Ein **Transformer** ist eine von Google entwickelte Deep-Learning-Architektur, die einen Aufmerksamkeits-mechanismus (Attention) integriert. [Wiki](#)

→ **Hauptkomponenten:**

**Encoder:** Verarbeitet die Eingabesequenz und wandelt sie in eine abstrakte Repräsentation um.

**Decoder:** Nutzt die Encoder-Ausgabe, um die Zielsequenz zu generieren.

→ **Aufmerksamkeitsmechanismus:**

**Self-Attention:** Ermöglicht es dem Modell, unterschiedliche Teile der Eingabesequenz zu gewichten, um kontextrelevante Informationen zu extrahieren.

**Multi-Head Attention:** Erlaubt es, verschiedene Aspekte der Sequenz gleichzeitig zu berücksichtigen, indem mehrere Aufmerksamkeitseinheiten parallel arbeiten.



## Was sind diese LLMs?

...und nun?

### Funktionsweise:

- **Eingabe:** Ein Text- oder Dateninput (z. B. eine Frage oder ein Satzanfang).
- **Verarbeitung:** Das Modell berechnet Wahrscheinlichkeiten für mögliche nächste Wörter oder Sätze.
- **Ausgabe:** Die wahrscheinlichste Wortfolge wird generiert.

**Datenquellen:** LLMs können mit öffentlichen Internetdaten, Unternehmensinformationen oder spezialisierten Textkorpora trainiert werden.

- Bspw. „internet-scale Data“

## Beispielhafte Interaktion

...was leitet sich daraus ab?

→ **Eingabe:** „Was gab es heute zum Mittagessen?“

Mögliche Antworten: „Nudeln“, „Reis“, „Steak“ – basierend auf Wahrscheinlichkeiten aus den Trainingsdaten.

### **Kein echtes Wissen:**

LLMs haben keine „Fakten“, sondern arbeiten mit statistischen Wahrscheinlichkeiten.

**Problem:** „Garbage in, Garbage out“ – schlechte Trainingsdaten führen zu fehlerhaften oder voreingenommenen Antworten.

**Halluzinationen:** LLMs können erfundene oder falsche Informationen ausgeben.

### **Bias & ethische Herausforderungen:**

Ein *Modell*, das mit voreingenommenen Daten trainiert wird, kann diskriminierende Ergebnisse liefern.

Beispiel: Geschlechtsspezifische Verzerrungen bei LLM-basierten Bewerbungsprozessen.

# Open Source holt auf...

...Welche *open source* Modelle  
gibt's denn jetzt.

...ein paar Beispiele:

- Meta LLaMA Modelle (llama3.1, llama3.2 vision, llama3.3 ...) [LLaMA-Huggingface](#)
- Qwen: Alibaba Cloud's general-purpose AI models [Qwen-Huggingface](#)
- DeepSeek: V3, R1, etc. [DeepSeek-Huggingface](#)
- Microsoft: z.B. Phi4-multimodal (Text, Audio, Bild) [Msft-Huggingface](#)

## Open Source holt auf...

...testen testen & Lizenzen 

Testet die Modelle:

<https://huggingface.co/spaces>

Vergleicht sie:

[https://huggingface.co/spaces/open-llm-leaderboard/open\\_llm\\_leaderboard](https://huggingface.co/spaces/open-llm-leaderboard/open_llm_leaderboard)



Achtet unbedingt auf die Lizenzen!

- für kommerzielle Nutzung sollte es z.B. unter **Apache 2.0** lizenziert sein
- Meta, Qwen u.a. haben meist ihre eigenen Lizenzen  
z.B. kommerzielle Lizenz nötig ab
  - x MAUs
  - x€ Umsatz ...

Prüft das am besten, bevor ihr ein Modell einsetzen wollt.

# Herausforderungen für den Betrieb lokaler LLMs...

...Hard- & Software 

- **Hoher Hardware-Bedarf für große Modelle** → Starke GPUs erforderlich.
- **Komplexe Einrichtung & Wartung** → Technisches Know-how nötig.
- **Skalierung schwieriger als in der Cloud** → Begrenzung durch lokale Ressourcen.



## Gründe & Vorteile lokaler LLMs

### **Datenschutz & Sicherheit**

- **Keine Cloud-Abhängigkeit** → Sensible Daten bleiben im Unternehmen oder auf dem eigenen Gerät.
- **Bessere Compliance** → Erfüllt strenge Datenschutzrichtlinien (z. B. DSGVO, HIPAA).
- **Schutz vor Datenlecks** → Kein Risiko, dass Nutzerdaten an externe Anbieter gelangen.

### **Geschwindigkeit & Verfügbarkeit**

- **Schnellere Reaktionszeiten** → Keine Latenz durch Internetanfragen an externe Server.
- **Offline-Nutzung** → LLMs funktionieren auch ohne Internetverbindung.
- **Unabhängigkeit von Cloud-Ausfällen** → Der Betrieb bleibt stabil, auch wenn Cloud-Services ausfallen.

## Gründe & Vorteile lokaler LLMs



### Anpassung & Kontrolle

- **Modellanpassung möglich** → Feinabstimmung (Fine-Tuning) mit unternehmenseigenen Daten.
- **Prompt Engineering & Optimierung** → Lokale Steuerung für spezifische Anwendungsfälle.
- **Kontrolle über die Modellversion** → Keine unerwarteten Updates oder API-Änderungen.



### Kostenersparnis? (langfristig)

- **Keine API-Kosten** → Lizenzen und Abfragen externer APIs können teuer sein.
- **Einmalige Hardware-Investition** → Statt wiederkehrender Cloud-Gebühren.
- **Effizienter für regelmäßige Nutzung** → Besonders vorteilhaft für Unternehmen mit hohem Anfragevolumen.

# Gründe & Vorteile lokaler LLMs



## Unternehmensintegration

- **Interne Systeme & Software** → Direkte Integration in bestehende IT-Strukturen.
- **On-Premise-Betrieb für Branchen mit hohen Sicherheitsanforderungen** → Z. B. Gesundheitswesen, Banken, Militär.
- **Keine Vendor-Lock-in-Problematik** → Unabhängigkeit von spezifischen Cloud-Anbietern.



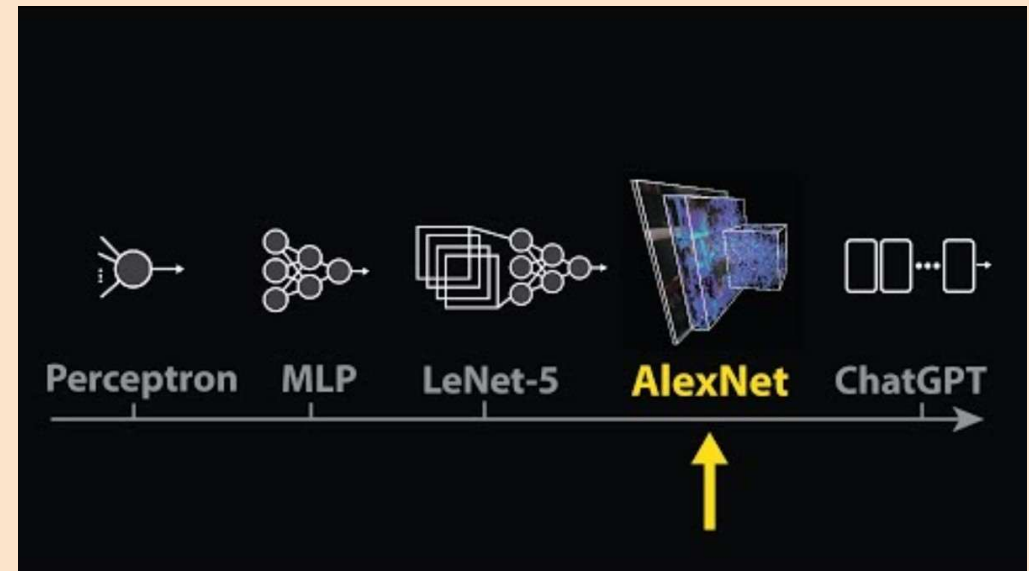
## Technische Möglichkeiten & Open-Source-Nutzung

- **Nutzung optimierter Modelle** → Open-Source-LLMs wie Mistral, LLaMA oder Falcon können lokal effizient laufen.
- **Hardware-Flexibilität** → Betrieb auf GPUs, TPUs oder quantisierten Modellen für CPUs.
- **Experimentierfreiheit** → Entwickler können Modelle modifizieren und erforschen.

# Goody zum Abschluss 📺

“The moment we stopped understanding AI [AlexNet] ”

© Welch Labs auf YouTube



<https://www.youtube.com/watch?v=UZDiGooFs54>