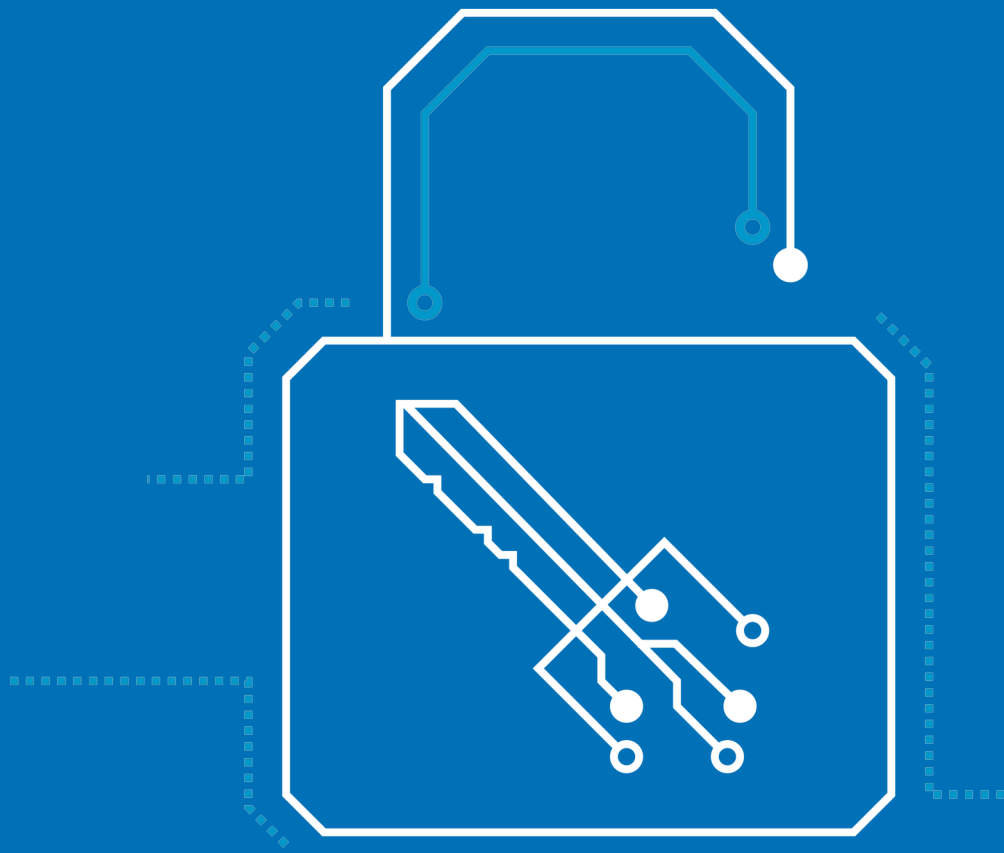
DFN-CERT

Grundlagen der LLM: Embeddings

84. DFN Betriebstagung, Jan Kohlrausch, DFN-CERT

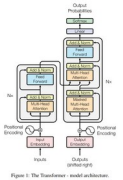
Übersicht

- Einführung und Motivation: Technische Herausforderungen
- Token und Embeddings
- Wie funktioniert ein LLM (Transformer)
- Training und Fine-Tuning
- Was fehlt noch: Ausblick

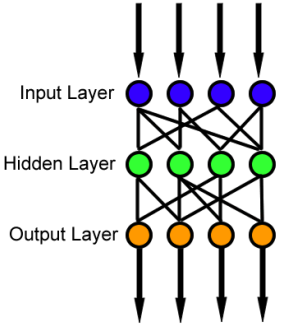


Einführung und Motivation

- Architektur eines LLM (Transformers) [2]:



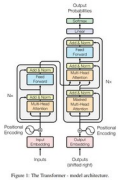
- Starke Vereinfachung: Architektur eines Feed-Forward neuronales Netz ([1])
  - Vorgabe von Kommandos und Daten über die Eingabeschicht (Input Layer) → Prompt.
  - Wissen über Sprache und Regeln ist in den Schichten (Layer) des LLM repräsentiert (Milliarden / Billionen von Parametern).
  - Parameter werden beim Training und Fine-Tuning bestimmt.
  - In den internen Schichten wird die Antwort berechnet.
  - Ausgabe der Antwort erfolgt über die Ausgabeschicht (Output Layer).



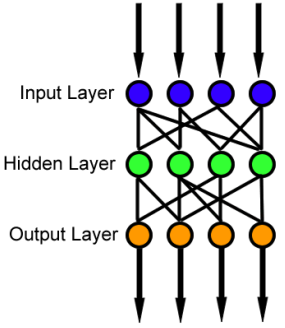
DFN
DEUTSCHES FORSCHUNGSNETZ

Einführung und Motivation

- Architektur eines LLM (Transformers) [2]:



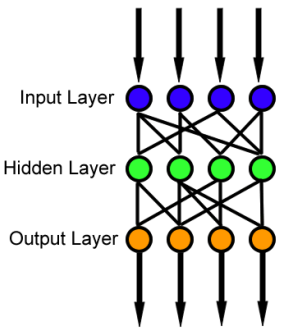
- Starke Vereinfachung: Architektur eines Feed-Forward neuronales Netz ([1])
  - Vorgabe von Kommandos und Daten über die Eingabeschicht (Input Layer) → Prompt.
  - Wissen über Sprache und Regeln ist in den Schichten (Layer) des LLM repräsentiert (Milliarden / Billionen von Parametern).
  - Parameter werden beim Training und Fine-Tuning bestimmt.
  - In den internen Schichten wird die Antwort berechnet.
  - Ausgabe der Antwort erfolgt über die Ausgabeschicht (Output Layer).
  - ➔ **y = f(x)**
    x: Input Layer, f: Hidden Layer, y: Output Layer



DFN
DEUTSCHES FORSCHUNGSNETZ

Herausforderung Nr. 1: numerischer Input

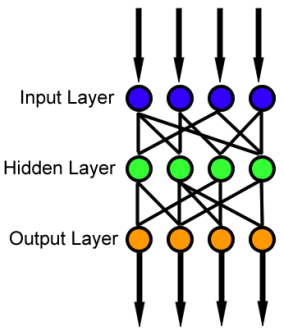
- Input muss numerisch sein



DFN
DEUTSCHES FORSCHUNGSNETZ

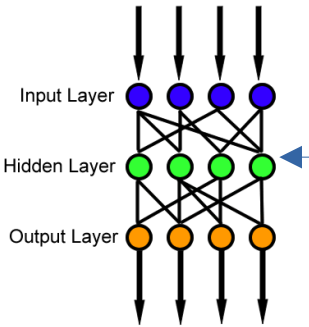
Herausforderung Nr. 1: numerischer Input

- Input muss numerisch sein
- ✔ Text → numerische Repräsentierung (Token)



DFN
DEUTSCHES FORSCHUNGSNETZ

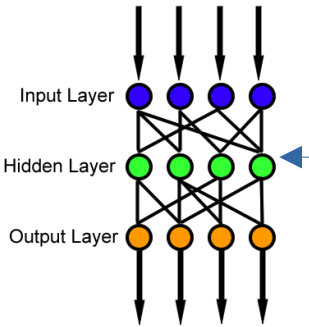
Herausforderung Nr. 2: Determinismus



- Output ist fest vom Input abhängig

DFN
DEUTSCHES FORSCHUNGSNETZ

Herausforderung Nr. 2: Determinismus

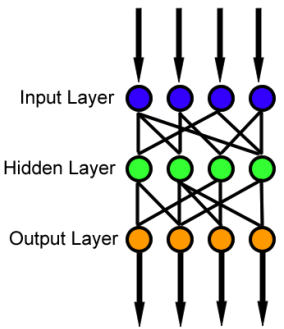


- Output ist fest vom Input abhängig
- ✔ Injektion einer Zufallskomponente (Temperatur)

DFN
DEUTSCHES FORSCHUNGSNETZ

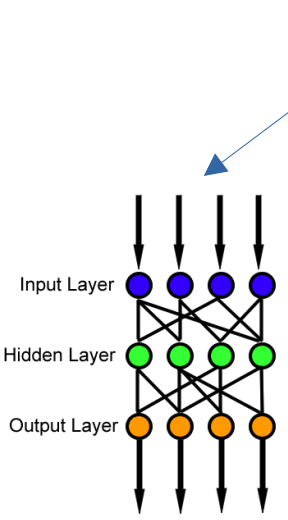
Herausforderung Nr. 3: Feste Eingabestruktur

- Größe der Input Layer ist statisch und begrenzt
- Keine beliebig große Eingabe von Anweisungen und Daten



DFN
DEUTSCHES FORSCHUNGSNETZ

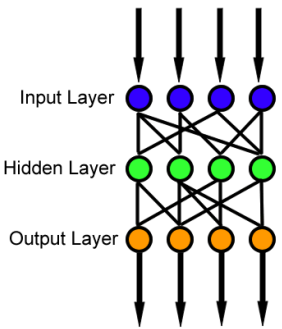
Herausforderung Nr. 3: Feste Eingabestruktur



- Größe der Input Layer ist statisch und begrenzt
- Keine beliebig große Eingabe von Anweisungen und Daten
- ✗ Erhöhung der Anzahl der Elemente („Context Window“)
- ✗ Aktuelle Foundation Modelle verarbeiten eine Größe von ca 1M Token parallel (Buch: 70.000 – 120.000 Wörter/Token)
- ➔ Fenster ist aber begrenzt!
- ➔ Verarbeitung großer Textmengen durch das LLM?

DFN
DEUTSCHES FORSCHUNGSNETZ

Herausforderung Nr. 3: Feste Eingabestruktur

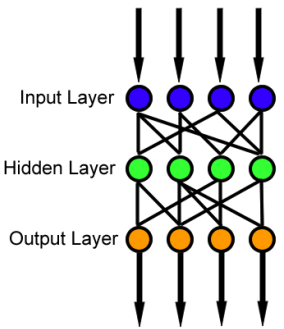


- Größe der Input Layer ist statisch und begrenzt
- Keine beliebig große Eingabe von Anweisungen und Daten
- ✗ Erhöhung der Anzahl der Elemente („Context Window“)
- ✗ Aktuelle Foundation Modelle verarbeiten eine Größe von ca 1M Token parallel (Buch: 70.000 – 120.000 Wörter/Token)
- ➔ Fenster ist aber begrenzt!
- ➔ Verarbeitung großer Textmengen durch das LLM?

DFN
DEUTSCHES FORSCHUNGSNETZ

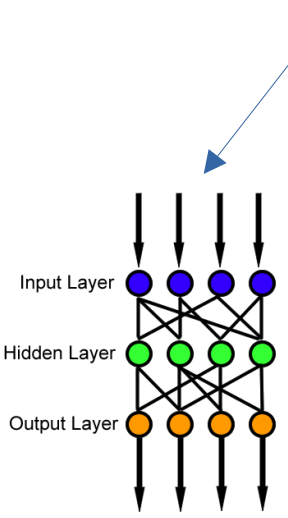
Herausforderung Nr. 4: Dynamisches Lernen

- Wissen (Daten und Regeln) wird in Layer des LLM gespeichert – dynamische Erweiterung des Wissen und wo ist interner Speicherbereich oder Gedächtnis?



DFN
DEUTSCHES FORSCHUNGSNETZ

Herausforderung Nr. 4: Dynamisches Lernen



- Wissen (Daten und Regeln) wird in Layer des LLM gespeichert – dynamische Erweiterung des Wissen und wo ist interner Speicherbereich oder Gedächtnis?
  - ✗ Zustand über mehrere Anfragen merkt sich in das LLM in den Eingabedaten (Context im Prompt)
  - ✗ Zwischenergebnisse beim Reasoning werden auch im Prompt eingefügt
  - ✗ Dynamische Erweiterung durch externes Wissen (RAG und Agenten)
  - ✗ Permanente Erweiterung durch Fine-Tuning des LLMs
  - ➔ Entweder nicht persistent oder extrem kostenintensiv!
  - ➔ Bedeutende Abhängigkeiten von/zu Herausforderung 3!

DFN
DEUTSCHES FORSCHUNGSNETZ

Zusammenfassung: LLM Herausforderungen

- LLMs bieten erstaunliche Leistungen beim Text-Verständnis und Generierung:
  - Zusammenfassung von Texten
  - Generierung von Programmcode
  - …
- Es sind aber immer noch Limitierungen vorhanden:
  - Begrenztes Context Window
  - Kein direktes Äquivalent zum Gedächtnis oder Computer-Speicher:
    - Computer: Giga/Terabytes
    - LLM: Megabytes

DFN
DEUTSCHES FORSCHUNGSNETZ

Motivation: Embeddings

- Zentrale Datenstruktur von KI-Modellen für die Verarbeitung natürlichsprachlicher Inhalte (natural language processing - NLP)
- Grundlage auch für Large-Language-Modelle (LLM)
- Ermöglichen ein intuitives Verständnis, wie diese Modelle Texte verstehen
- Zentrale Bedeutung bei Retrieval-Augmented Generation (RAG)

DFN
DEUTSCHES FORSCHUNGSNETZ

Embeddings und die Semantik von Wörtern

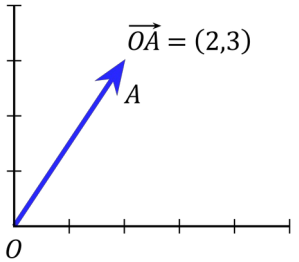
- Jedem Wort (genauer Token) wird jeweils ein Embedding zugeordnet[1]:
  - Wort ↔ Embedding
  - Embeddings gibt es aber auch für Text-Fragmente und Bilder
- Token vs. Embedding:
  - Token ist beliebige numerische Repräsentierung eines Wortes
  - Embedding repräsentiert zusätzlich dessen semantische Bedeutung
- Mathematische Repräsentierung durch Vektor (hohe Dimensionalität)
- Werden durch KI-Modell auf der Basis von unüberwachtem Lernen ermittelt (Prädiktion von Wörtern)

DFN
DEUTSCHES FORSCHUNGSNETZ

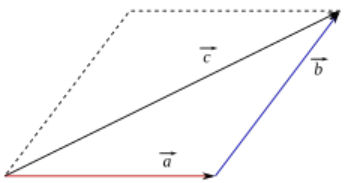
[1] genauer: textuelle Eingaben werden in Token aufgegliedert, die jeweils ein Wort oder ein Teil davon numerisch repräsentieren

Wiederholung Vektor

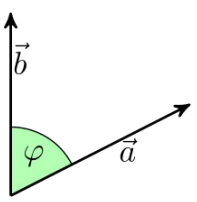
- Beispiel für ein Vektor:



- Mit Vektoren rechnen:



- Ähnlichkeit von Vektoren: Winkel Phi



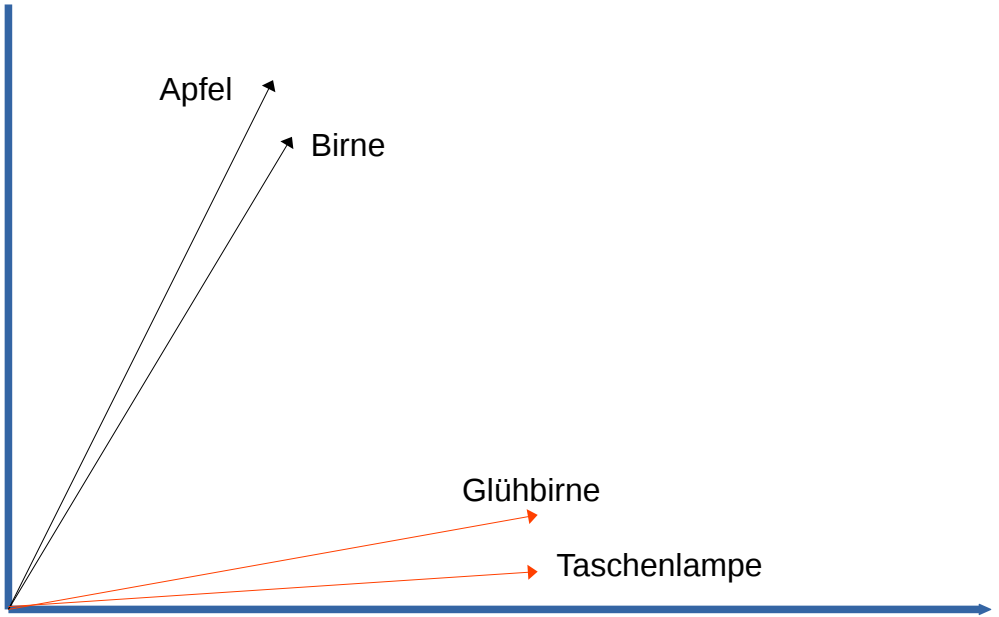
DFN
DEUTSCHES FORSCHUNGSNETZ

Embeddings: Wichtige Eigenschaften

1) Ermöglicht die Erfassung der semantische Bedeutung von Wörtern
2) Messung der semantische Ähnlichkeit von Wörtern
3) Abstraktion für Semantik

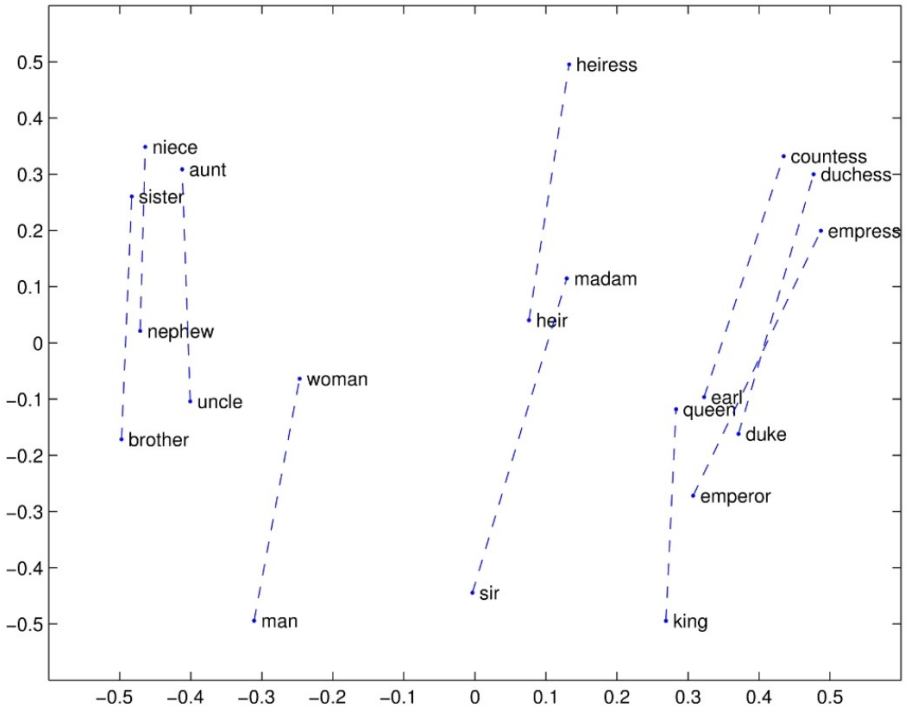
DFN
DEUTSCHES FORSCHUNGSNETZ

Embeddings in der Praxis: Ähnlichkeit



DFN
DEUTSCHES FORSCHUNGSNETZ

Embeddings in der Praxis: Abstraktion Geschlecht



Aus Stanford: Natural Language Processing with Deep Learning CS224N/Ling284

DFN
DEUTSCHES FORSCHUNGSNETZ

Embeddings in der Praxis: Abstraktion Geschlecht

- Vektoren repräsentieren semantische Konzepte
- Konzept des Geschlechtes:
  - ➔ King – Man + Woman ≈ Queen
  - ➔ Schwester – Frau + Mann ≈ Bruder
- Weitere Beispiele:
  - ➔ Michael Jordan – Basketball + Tennis ≈ ?
  - ➔ Eiffelturm - Frankreich + USA ≈ ?

DFN
DEUTSCHES FORSCHUNGSNETZ

Embeddings in der Praxis: Limitierungen

- Embeddings für ein Wort sind statisch: Beispiel „Bank“
  - Bank = Sitzbank?
  - Bank = Bankfiliale?
  - ➔Statisches Embedding ermöglicht keine Unterscheidung der konkreten Bedeutung!
- Exakte Bedeutung geht nicht immer aus dem Wort selbst hervor
  - ➔Kontext wird gebraucht
- Kein umfassendes Verständnis eines Textes

DFN
DEUTSCHES FORSCHUNGSNETZ

Embeddings in der Praxis: Limitierungen

- Embeddings für ein Wort sind statisch: Beispiel „Bank“
  - Bank = Sitzbank?
  - Bank = Bankfiliale?
  - ➔Statisches Embedding ermöglicht keine Unterscheidung der konkreten Bedeutung!
- Exakte Bedeutung geht nicht immer aus dem Wort selbst hervor
  - ➔Kontext wird gebraucht
- Kein umfassendes Verständnis eines Textes

➢ Spezialaufgabe für LLMs:
Einbeziehung des Kontextes für dynamische Embeddings

DFN
DEUTSCHES FORSCHUNGSNETZ

Dynamische kontext-basierte Embeddings

- LLM berechnet aus statischen Embeddings dynamische kontext-basierte Embeddings („Dynamic Contextual Representation“):
  - Kontext und gelernte Zusammenhänge zwischen Wörtern werden einbezogen: Embedding ist nicht mehr eindeutig für ein Wort
  - Unterscheidung zwischen Bank (Sitzbank) und Bank (Bankfiliale)
    - Er/sie setzte sich auf die *Bank*, um sich auszuruhen.
    - Sie/er ging zur *Bank*, um Geld abzuheben.
  - Abstrakte Bedeutungen können erkannt und angewendet werden: zum Beispiel die Unterscheidung zwischen Daten und Instruktionen
- Ermöglicht wird dies durch die Attention-Schichten des LLM

DFN
DEUTSCHES FORSCHUNGSNETZ

Embeddings in der Praxis: LLM Arbeitsweise

Aufgliedern des textuellen Prompts in Token:
→ “Tokenization helps models understand text.“ → ["Token", "ization", " helps", " models", " understand", " text", "."]

DFN
DEUTSCHES FORSCHUNGSNETZ

Embeddings in der Praxis: LLM Arbeitsweise

Aufgliedern des textuellen Prompts in Token:
→ “Tokenization helps models understand text.“ → ["Token", "ization", " helps", " models", " understand", " text", "."]

Transformation der Token in Embeddings
→ numerischer Input mit semantische Bedeutung der Wörter

DFN
DEUTSCHES FORSCHUNGSNETZ

Embeddings in der Praxis: LLM Arbeitsweise

Aufgliedern des textuellen Prompts in Token:
→ “Tokenization helps models understand text.“ → ["Token", "ization", " helps", " models", " understand", " text", "."]

Transformation der Token in Embeddings
→ numerischer Input mit semantische Bedeutung der Wörter

Berechnung dynamischer Embeddings (Dynamic Contextual Representation)
→ LLM lernt konkrete Semantik des Wortes „models“ aus dem Kontext

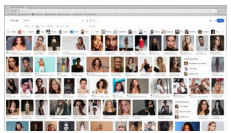
DFN
DEUTSCHES FORSCHUNGSNETZ

Embeddings in der Praxis: LLM Arbeitsweise

Aufgliedern des textuellen Prompts in Token:
→ “Tokenization helps models understand text.“ → ["Token", "ization", " helps", " models", " understand", " text", "."]

Transformation der Token in Embeddings
→ numerischer Input mit semantische Bedeutung der Wörter

Berechnung dynamischer Embeddings (Dynamic Contextual Representation)
→ LLM lernt konkrete Semantik des Wortes „models“ aus dem Kontext

→  ?

DFN
DEUTSCHES FORSCHUNGSNETZ

Embeddings in der Praxis: LLM Arbeitsweise

Aufgliedern des textuellen Prompts in Token:
→ “Tokenization helps models understand text.“ → ["Token", "ization", " helps", " models", " understand", " text", "."]

Transformation der Token in Embeddings
→ numerischer Input mit semantische Bedeutung der Wörter

Berechnung dynamischer Embeddings (Dynamic Contextual Representation)
→ LLM lernt konkrete Semantik des Wortes „models“ aus dem Kontext
→ ChatGPT: „In the sentence “tokenization helps models understand text”, the word “models” refers to machine learning models designed for natural language processing (NLP).“

DFN
DEUTSCHES FORSCHUNGSNETZ

Embeddings in der Praxis: LLM Arbeitsweise

Aufgliedern des textuellen Prompts in Token:
→ “Tokenization helps models understand text.“ → ["Token", "ization", " helps", " models", " understand", " text", "."]

Transformation der Token in Embeddings
→ numerischer Input mit semantische Bedeutung der Wörter

Berechnung dynamischer Embeddings (Dynamic Contextual Representation)
→ LLM lernt konkrete Semantik des Wortes „models“ aus dem Kontext
→ ChatGPT: „In the sentence “tokenization helps models understand text”, the word “models” refers to machine learning models designed for natural language processing (NLP).“

Schrittweise Berechnung der Antwort aus den dynamischen Embeddings
→ nächstes Token für die Antwort (das Wahrscheinlichste)
→ Token wird in Input übernommen (auto-regressive): „Tokenization is the step...“
→ Vorgang wird bei „End-Token“ abgebrochen

DFN
DEUTSCHES FORSCHUNGSNETZ

Embeddings in der Praxis: LLM Arbeitsweise

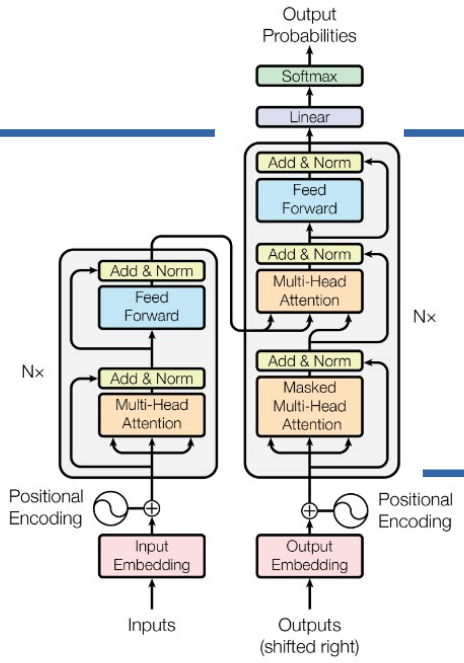


Figure 1: The Transformer - model architecture.

Berechnung des nächsten Token

Dynamische kontext-basierte Embeddings

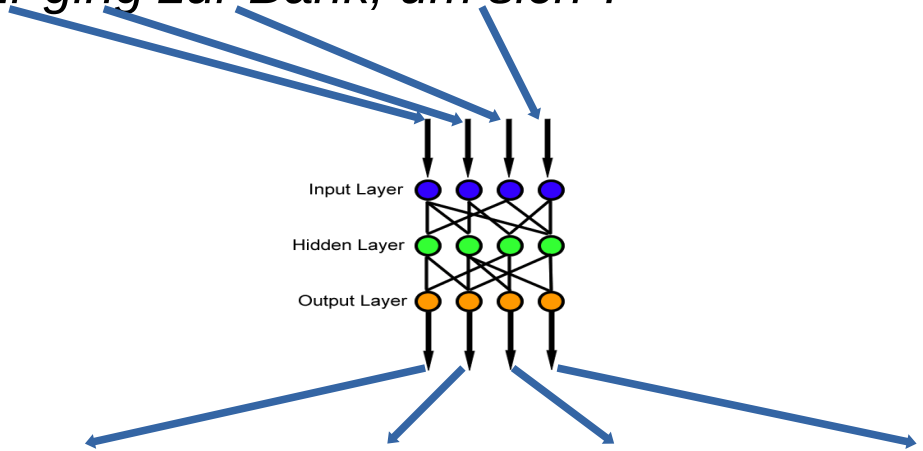
Statische Embeddings

DFN
DEUTSCHES FORSCHUNGSNETZ

Beispiel: Ausgabe des LLM – nächstes Wort

Fortsetzung des Satzes „Er ging zur Bank, um sich“:
Wahrscheinlichkeiten für „auszuruhen“, „Geld“, „zu beschweren“, „lesen“, …

Er ging zur Bank, um sich ?



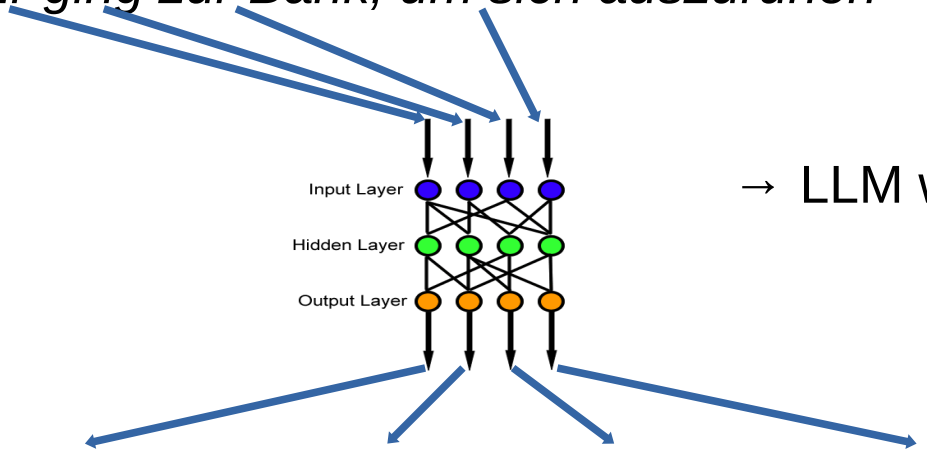
auszuruhen	Geld	zu beschweren	lesen
0.41	0.33	0.25	0.01

DFN
DEUTSCHES FORSCHUNGSNETZ

Beispiel: Ausgabe des LLM – nächstes Wort

Fortsetzung des Satzes „Er ging zur Bank, um sich“:
Wahrscheinlichkeiten für „auszuruhen“, „Geld“, „zu beschweren“, „lesen“, …

Er ging zur Bank, um sich auszuruhen



→ LLM würde mit „auszuruhen“ fortsetzen

auszuruhen	Geld	zu beschweren	lesen
0.41	0.33	0.25	0.01

DFN
DEUTSCHES FORSCHUNGSNETZ

Beispiel: Nächstes Wort - ChatGPTs Einschätzung

Frage an ChatGPT nach den Wahrscheinlichkeiten:

Er ging zur Bank, um sich ?

Bewertung der vier Optionen

Fortsetzung	Beispiel kompletter Satz	Plausibilität	Geschätzte relative Wahrscheinlichkeit
Geld	„Er ging zur Bank, um sich **Geld zu holen**."	Sehr typisch für „Bank"	~70%
beschweren	„Er ging zur Bank, um sich **zu beschweren**."	Möglich, aber weniger häufig	~20%
auszuruhen	„Er ging zur Bank, um sich **auszuruhen**."	Nur sinnvoll, wenn „Bank" = Sitzbank	~7%
lesen	„Er ging zur Bank, um sich **lesen** ..."	Grammatisch unvollständig („um zu lesen" nötig)	~3%

DFN
DEUTSCHES FORSCHUNGSNETZ

Beispiel: Auswechseln eines Wortes

Was passiert, wenn wir „beschweren“ durch „hinzusetzen“ austauschen?

Er ging zur Bank, um sich ?

Einschätzung der Fortsetzungen

Wort	Grammatik	Plausibilität	Grobe relative Wahrscheinlichkeit
auszuruhen	korrekt reflexiv („sich auszuruhen“)	sehr natürlich	~40–45 %
hinzusetzen	korrekt reflexiv („sich hinzusetzen“)	ebenfalls sehr natürlich	~35–40 %
lesen	grammatisch möglich („um sich zu lesen“ aber untypisch; eher „um zu lesen“)	deutlich weniger plausibel	~10–15 %
Geld	grammatisch falsch nach „um sich“	praktisch 0 %	

DFN
DEUTSCHES FORSCHUNGSNETZ

LLM: Training und Fine-Tuning

	Methodik	Ergebnis	Aufwand
Training	• Unüberwachtes Lernen auf umfangreichen Texten • Keine Vorgabe von Regeln oder Anweisungen ➔ Initiale Optimierung der Parameter		
Fine-Tuning			

DFN
DEUTSCHES FORSCHUNGSNETZ

LLM: Training und Fine-Tuning

	Methodik	Ergebnis	Aufwand
Training	• Unüberwachtes Lernen auf umfangreichen Texten • Keine Vorgabe von Regeln oder Anweisungen ➔ Initiale Optimierung der Parameter	• LLM lernt Struktur und Grammatik der Sprachen. • LLM kann fehlerfreie Texte produzieren • Antwort wird aus Text-Wahrscheinlichkeiten generiert	
Fine-Tuning			

DFN
DEUTSCHES FORSCHUNGSNETZ

LLM: Training und Fine-Tuning

	Methodik	Ergebnis	Aufwand
Training	• Unüberwachtes Lernen auf umfangreichen Texten • Keine Vorgabe von Regeln oder Anweisungen ➔ Initiale Optimierung der Parameter	• LLM lernt Struktur und Grammatik der Sprachen. • LLM kann fehlerfreie Texte produzieren • Antwort wird aus Text-Wahrscheinlichkeiten generiert	• Extrem hoch: Rechenzentrum • Extrem komplexe numerische Optimierung der Parameter
Fine-Tuning			

DFN
DEUTSCHES FORSCHUNGSNETZ

LLM: Training und Fine-Tuning

	Methodik	Ergebnis	Aufwand
Training	• Unüberwachtes Lernen auf umfangreichen Texten • Keine Vorgabe von Regeln oder Anweisungen → Initiale Optimierung der Parameter	• LLM lernt Struktur und Grammatik der Sprachen. • LLM kann fehlerfreie Texte produzieren • Antwort wird aus Text-Wahrscheinlichkeiten generiert	• Extrem hoch: Rechenzentrum • Extrem komplexe numerische Optimierung der Parameter
Fine-Tuning	• Vorgabe von Beispielen zum gewünschten Verhalten (Höflichkeit, Nützlichkeit und Persönlichkeit) • Spezifische Beispiele für Anwendungen (Text zusammenfassen) • Domänenspezifisches Wissen (Mathematik) → Verfeinerung der Parameter		

DFN
DEUTSCHES FORSCHUNGSNETZ

LLM: Training und Fine-Tuning

	Methodik	Ergebnis	Aufwand
Training	• Unüberwachtes Lernen auf umfangreichen Texten • Keine Vorgabe von Regeln oder Anweisungen ➔ Initiale Optimierung der Parameter	• LLM lernt Struktur und Grammatik der Sprachen. • LLM kann fehlerfreie Texte produzieren • Antwort wird aus Text-Wahrscheinlichkeiten generiert	• Extrem hoch: Rechenzentrum • Extrem komplexe numerische Optimierung der Parameter
Fine-Tuning	• Vorgabe von Beispielen zum gewünschten Verhalten (Höflichkeit, Nützlichkeit und Persönlichkeit) • Spezifische Beispiele für Anwendungen (Text zusammenfassen) • Domänenspezifisches Wissen (Mathematik) ➔ Verfeinerung der Parameter	• LLM kann Antworten den Wünschen der Anwender anpassen • Bessere Performanz bei häufigen Aufgaben • Anpassung an spezielle Aufgaben • Reasoning • Bessere Erkennung von Angriffen	

DFN
DEUTSCHES FORSCHUNGSNETZ

LLM: Training und Fine-Tuning

	Methodik	Ergebnis	Aufwand
Training	• Unüberwachtes Lernen auf umfangreichen Texten • Keine Vorgabe von Regeln oder Anweisungen ➔ Initiale Optimierung der Parameter	• LLM lernt Struktur und Grammatik der Sprachen. • LLM kann fehlerfreie Texte produzieren • Antwort wird aus Text-Wahrscheinlichkeiten generiert	• Extrem hoch: Rechenzentrum • Extrem komplexe numerische Optimierung der Parameter
Fine-Tuning	• Vorgabe von Beispielen zum gewünschten Verhalten (Höflichkeit, Nützlichkeit und Persönlichkeit) • Spezifische Beispiele für Anwendungen (Text zusammenfassen) • Domänenspezifisches Wissen (Mathematik) ➔ Verfeinerung der Parameter	• LLM kann Antworten den Wünschen der Anwender anpassen • Bessere Performanz bei häufigen Aufgaben • Anpassung an spezielle Aufgaben • Reasoning • Bessere Erkennung von Angriffen	• Sehr hoch: sehr leistungsfähige Workstation • Parameter müssen nur lokal optimiert werden

DFN
DEUTSCHES FORSCHUNGSNETZ

Fine-Tuning und LoRa

- Vorhandene Parameter sind Ausgangspunkt für die weitere Optimierung:
  - Neue Daten
  - Neue Anforderungen und Regeln
  - Spezifische Aufgaben
  - Aber: extrem viele Parameter
- Low-Rank Adaptation (LoRa):
  - Sind wirklich alle Parameter relevant (Low Rank Matrix)?
  - Idee: Addition einer deutlich kleineren Anzahl von Parametern, die optimiert werden
  - Anpassung von deutlich weniger Parametern:
    - Zum Beispiel: 2*n Parameter anstelle n*n Parameter-Matrix

DFN
DEUTSCHES FORSCHUNGSNETZ

Mensch vs LLM: Lernen

	
• Lernt dynamisch • Kann direkt aus Fehlern lernen • Kann Inhalte selbst bestimmen • Gehirn entwickelt sich kontinuierlich weiter	• Initiales Training • Begrenztes, externes Wissen durch RAG und Agenten • Erweiterung durch Fine-Tuning möglich (isolierte Phase; kein dynamisches Lernen) • Externe Vorgabe von Lern-Beispielen

DFN
DEUTSCHES FORSCHUNGSNETZ

Mensch vs LLM: Lernen

	
• Lernt dynamisch • Kann direkt aus Fehlern lernen • Kann Inhalte selbst bestimmen • Gehirn entwickelt sich kontinuierlich weiter ➔ Denken = Lernen ➔ Selbstständige und autonome Weiterentwicklung	• Initiales Training • Begrenztes, externes Wissen durch RAG und Agenten • Erweiterung durch Fine-Tuning möglich (isolierte Phase; kein dynamisches Lernen) • Externe Vorgabe von Lern-Beispielen ➔ Entweder Denken (Inferenz) oder Lernen (Fine-Tuning) ➔ Keine autonome Weiterentwicklung

DFN
DEUTSCHES FORSCHUNGSNETZ

Mensch vs LLM: Lernen

	
• Lernt dynamisch • Kann direkt aus Fehlern lernen • Kann Inhalte selbst bestimmen • Gehirn entwickelt sich kontinuierlich weiter ➔ Denken = Lernen ➔ Selbstständige und autonome Weiterentwicklung	• Initiales Training • Begrenztes, externes Wissen durch RAG und Agenten • Erweiterung durch Fine-Tuning möglich (isolierte Phase; kein dynamisches Lernen) • Externe Vorgabe von Lern-Beispielen ➔ Entweder Denken (Inferenz) oder Lernen (Fine-Tuning) ➔ Keine autonome Weiterentwicklung

DFN
DEUTSCHES FORSCHUNGSNETZ

➢ Skynet muss noch warten! (?)

Zusammenfassung

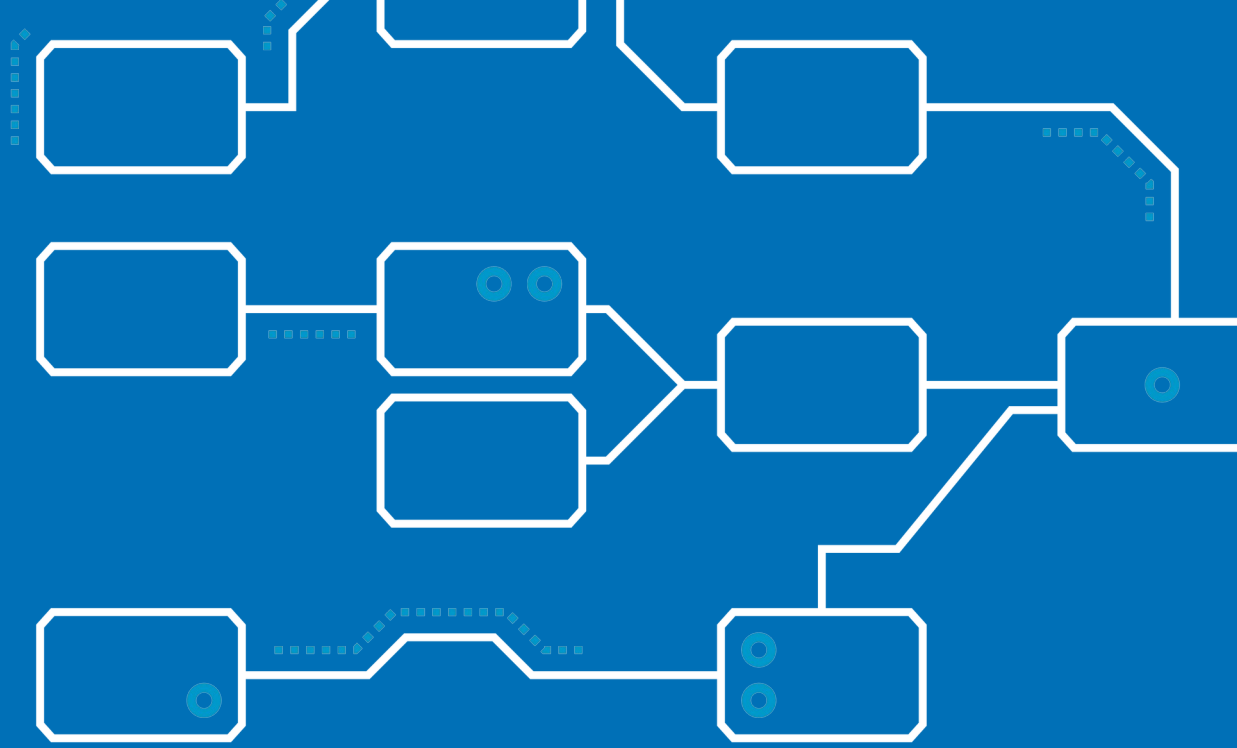
- Es gibt Gemeinsamkeiten, aber auch fundamentale Unterschiede zwischen Menschen und LLMs
- Es gibt aktuelle Herausforderungen durch begrenztes Kontext-Fenster und Lern-Fähigkeiten
- Embeddings sind zentrale Datenstruktur
- Dynamische Embeddings ermöglichen kontext-basierte Unterscheidung von Wörtern und Text-Verständnis
- Initiales Training ist extrem aufwendig
- Fine-Tuning ist auch sehr aufwendig, ermöglicht aber permanente Anpassungen und Verbesserungen
- Weitere Optimierung des Fine-Tunings durch LoRa

DFN
DEUTSCHES FORSCHUNGSNETZ

Ausblick

- Kern-Spezialität der LLMs sind die dynamischen Embeddings
- Technische Realisierung durch Attention-Schichten
- Wie funktionieren diese?
- Wie funktioniert Retrieval Augmented Generation (RAG) – Spoiler: was haben Embeddings damit zu tun?

DFN
DEUTSCHES FORSCHUNGSNETZ



Many Thanks / Vielen Dank

DFN-CERT

Referenzen

- [1] Image: Typical CNN architecture by "Aphex34" licensed under Creative Commons Attribution-Share Alike 4.0 International (https://creativecommons.org/licenses/by-sa/4.0/deed.en)
- [2] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS'17). Curran Associates Inc., Red Hook, NY, USA, 6000–6010.
- [3] GÉANT SIG-AI, https://community.geant.org/sig-ai/

DFN
DEUTSCHES FORSCHUNGSNETZ