Wie funktioniert ChatGPT?

KI-basierte Schreibwerkzeuge aus technischer Sicht

Andreas Hildebrandt

Institute for Computer Science, Johannes Gutenberg University Mainz

27.02.23

Was ist ChatGPT?

- ChatGPT ist ein statistisches Modell aus dem *NLP*-Bereich (Natural Language Processing), das auf die Simulation natürlicher Dialoge optimiert wurde
- Basiert auf GPT-3.5 (Generative Pre-Trained Transformer)
- Maschinelles/Statistisches Lernmodell mit ca. 175 Milliarden Parametern
- Trainiert auf ca. 45 TB Textdaten

Sprachmodelle

- GPT-n sind sogenannte *Sprachmodelle* (*language models*)
- Ein Sprachmodell ist eine Wahrscheinlichkeitsverteilung über Textsequenzen
- Genauer: ein Sprachmodell weist jeder Sequenz $w_1, \ldots, w_n$ von Worten mit beliebiger Sequenzlänge n die gemeinsame Wahrscheinlichkeitsverteilung $P(w_1, \ldots, w_n)$ zu
- Erinnerung: $P(w_1, \ldots, w_n)$ ist W'keit, dass in einem Text aus n Worten das erste Wort w_1 ist **und gleichzeitig** das zweite Wort w_2 ist **und gleichzeitig** ...

Sprachmodelle

- Noch genauer: GPT approximiert *bedingte W'keit* $P(w_n|w_1, \ldots, w_{n-1})$: W'keit, dass n-tes Wort w_n ist, wenn wir schon wissen, dass das erste Wort w_1 war **und gleichzeitig** das zweite Wort w_2 war **und gleichzeitig** ...
- Diese Wahrscheinlichkeit ist der methodische Kern von ChatGPT

Textgenerierung mit ChatGPT

- Was passiert bei Unterhaltung mit ChatGPT?
- Nutzer übermittelt **Anfrage (prompt / query)** als Text $w_1, \ldots, w_k$
- ChatGPT bestimmt bedingte Wahrscheinlichkeit $P(s|w_1, \ldots, w_k)$ für **alle möglichen Fortsetzungen** s[1]
- **Beispiel:**

The best thing about AI is its ability to

learn	4.5%
predict	3.5%
make	3.2%
understand	3.1%
do	2.9%

[1] GPT hat einen endlichen Wortschatz - betrachten wir gleich noch genauer

Textgenerierung mit ChatGPT

- Was passiert bei Unterhaltung mit ChatGPT?
- Nutzer übermittelt **Anfrage (prompt / query)** als Text $w_1, \ldots, w_k$
- ChatGPT bestimmt bedingte Wahrscheinlichkeit $P(s|w_1, \ldots, w_k)$ für **alle möglichen Fortsetzungen** s
- ChatGPT wählt aus allen möglichen Fortsetzungen eine aus ($\hat{s}$) und setzt $w_{k+1} = \hat{s}$
- Eines der "Worte", die ChatGPT kennt, steht für "Beende die Ausgabe"; wurde dieses gewählt, ist der Text fertig
- Ansonsten fahre wie oben fort mit $P(s|w_1, \ldots, w_{k+1})$

Textgenerierung mit ChatGPT

- Wie wählt man die Fortsetzung aus?
- Naiv: wähle jedesmal das Wort mit höchster W'keit
- Empirisch stellt man fest, dass die Texte dann repetitiv und oberflächlich werden
- Besser: erlaube auch nicht-optimale Worte (exponentialverteiltes Samplen mit Temperaturparameter, meist auf 0.8 festgelegt)

Textgenerierung mit ChatGPT

Beispiel:

- Temperatur 0 (immer wahrscheinlichstes Wort):
  The best thing about AI is its ability to automate processes and make decisions quickly and accurately. AI can be used to automate mundane tasks, such as data entry, and can also be used to make complex decisions, such as predicting customer behavior or analyzing large datasets. AI can also be used to improve customer service, as it can quickly and accurately respond to customer inquiries. AI can also be used to improve the accuracy of medical diagnoses and to automate the process of drug discovery.
- Temperatur 0.8:
  The best thing about AI is its ability to learn and develop over time, allowing it to continually improve its performance and be more efficient at tasks. AI can also be used to automate mundane tasks, allowing humans to focus on more important tasks. AI can also be used to make decisions and provide insights that would otherwise be impossible for humans to figure out.

Der Wortschatz von GPT

- GPT bricht Worte zunächst in Token auf
- Beispiel: “hamburger” $\Rightarrow$ “ham”, “bur”, “ger”
- Hilft bei seltenen, zusammengesetzten oder Fremdworten sowie bei Neologismen
- Können wir erstmal ignorieren. . .

Der Wortschatz von GPT

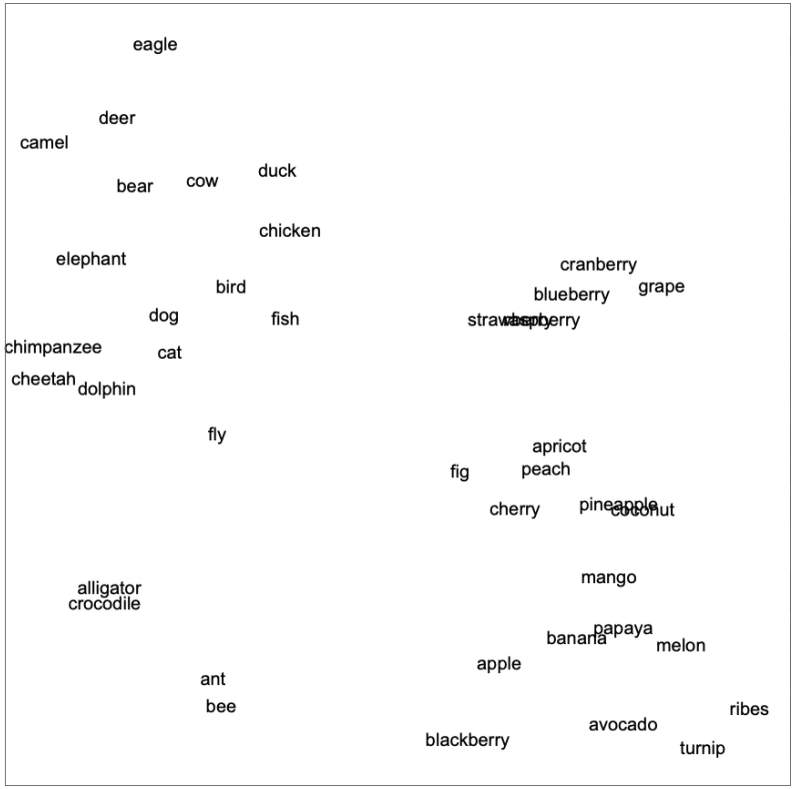
- Welche Worte kennt GPT?
- ChatGPT hat einen Wortschatz von 50.257 Token
- Jedes Token entspricht dann einer Zahl zwischen 1 und 50.257
- Für den weiteren Ablauf benötigen wir geeignetes Distanzmaß zwischen Wörtern (eigentlich: Token)
- Nicht geeignet: textueller Unterschied
  - gleich geschriebene Worte können unterschiedliches bedeuten
  - sehr unterschiedliche Worte können das gleiche bedeuten
- Statt dessen: lerne Word-Embedding

Word-Embeddings

- Word-Embedding: betrachte jedes Wort (eigentlich: Token) in n -dimensionalen Euklidischen Raum $\mathbb{R}^n$ ein
- Euklidischer Abstand der Embeddings dient dann als Abstand zwischen Wörtern
- Einbettung ist Abbildung der Token-Menge $[1, \ldots, 50.257] \to \mathbb{R}^n$
- Bei ChatGPT: $n = 12.288$
- Diese Abbildung wird aus den Trainingsdaten erlernt
- Bei GPT aus “word prediction” - Modell abgeleitet:
  - Gegeben: Tripel der Art “the _ cat” oder “_ black _” mit Leerstellen
  - Gesucht: Wahrscheinlichkeiten für Füllwörter
- Bestimmt aus großem Textkorpus
- Worte, die oft die gleiche Leerstelle ausfüllen können, sind ähnlich

Word-Embeddings

- **Beispiel:** 2d-Projektion (2 Hauptkomponenten einer PCA) eines einfachen Wort-Embeddings



https://writings.stephenwolfram.com/2023/02/what-is-chatgpt-doing-and-why-does-it-work/

Statistische Lernverfahren

- Wie berechnet ChatGPT die gesuchten Wahrscheinlichkeiten oder die genauen Word-Embeddings?
- Grundlage ist ein statistisches Lernverfahren

Statistische Lernverfahren – Wie "lernen" Computer?

> *"A computer program is said to learn from experience E with respect to some task T and some performance measure P, if its performance on T, as measured by P, improves with experience E."*
> *—Tom Mitchell, 1997*

- "Erfahrung" E durch Trainigsdaten realisiert
- Kann auf verschiedene Weise verwendet werden
  - **Instanzbasiert:** merkt sich die Trainigsbeispiele, vergleicht Anfrageinstanz mit diesen
  - **Modellbasiert:** verwendet mathematisches / statistisches Modell mit *Parametern*, die aus den Trainigsdaten abgeleitet werden, indem *Zielfunktion* auf Trainigsdatensatz optimiert wird

Statistische Lernverfahren – Wie “lernen” Computer?

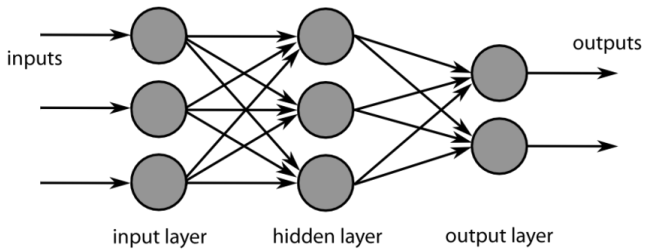
- GPT ist modellbasiertes Lernverfahren
- Implementiert als (sehr große) mathematische Funktion mit (ca. 175 Milliarden) Parametern
- Zielfunktion bewertet, wie gut Programm gestellte Aufgabe löst
- **Wichtige Varianten:**
  - **Supervised Learning:** Training auf annotierten Daten, d.h. Eingaben, für die eine korrekte Lösung bekannt ist und dem Training zur Verfügung steht
  - **Unsupervised Learning**: Training basiert nur auf Korrelationen in den Eingabedaten (Bsp.: Clustering)
- Training von GPT zum größten Teil **unsupervised**
- Liefert **vortrainiertes** Modell (*pre-trained*, das P in GPT), das später verfeinert wird

Tiefe Neuronale Netze

- Populäre Klasse maschineller Lernverfahren, inspiriert von Funktionsweise des Gehirns
- besteht aus Schichten von Neuronen, die Eingaben akzeptieren und Ausgaben berechnen
- jede Eingabe wird gewichtet, die gewichteten Eingaben addiert und dann in Aktivierungsfunktion gegeben

Tiefe Neuronale Netzwerke

- **Beispiel:**



Einfaches Feed-Forward Netzwerk

- Ausgabe y_k von Neuron k mit Eingaben $x_1, x_2, \ldots, x_n$:

$$y_k = \varphi\left(\sum_{j=0}^{n} w_{kj} x_j\right) = \varphi\left(w_{k1}x_1 + w_{k2}x_2 + \cdots + w_{kn}x_n\right)$$

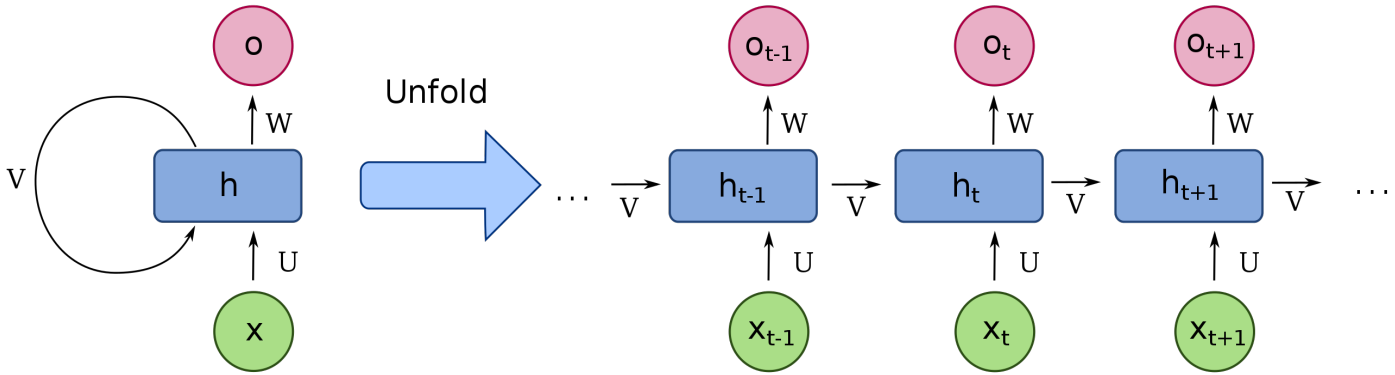
- φ “Aktivierungsfuntion”, heute oft Rectified Linear Units ($\varphi(x) = max(x, 0)$)
- w_{ij} Parameter, die gelernt werden sollen

Tiefe Neuronale Netzwerke

- Verschiedene Netzwerkarchitekturen für verschiedene Problemklassen gut geeignet
- **Beispiel:** Convolutional Neural Networks (CNN) hervorragend geeignet, um auf Bildern zu lernen (Bilderkennung, Bildgenerierung, . . .)
- Lange keine vergleichbar gute Architektur zur Verarbeitung von Textsequenzen bekannt

Tiefe Neuronale Netzwerke

- Klassische Standardarchitektur zur Textverarbeitung: Formen Rekurrenter Neuronaler Netzwerke (RNN)
- Verarbeiten Text als Sequenz oder Zeitreihe, ein Token nach dem anderen



- Information aus letztem Zustand in aktuellen Zustand mitübertragen (Kurzzeitgedächtnis)
- Architektur hat mehrere Probleme
  - schwer zu trainieren
  - vergesslich - am Ende langer Sätze erinnert sich Netzwerk kaum

Transformer

- Transformer haben RNNs (wie z.B. LTSM) als Architektur in vielen Bereichen abgelöst
- **Grundidee:** betrachte Text nicht sequentiell (ein Wort/Token nach dem anderen), sondern alle Worte/Token simultan
- Erreicht durch sogenannte **Attention**
  - Gegeben zwei Token w_i, w_j
  - a_{ij} misst den Einfluß, den der Wert von Token j auf das Token i hat
- **Self-attention**: Zusammenhänge zwischen Bestandteilen des gleichen Texts
- **Attention:** Zusammenhang zwischen Texten (z.B. Anfrage und Antwort oder Satz in Sprache 1 und Satz in Sprache 2)

Transformer

- Architektur von Transformern nicht ganz trivial zu erklären. . .

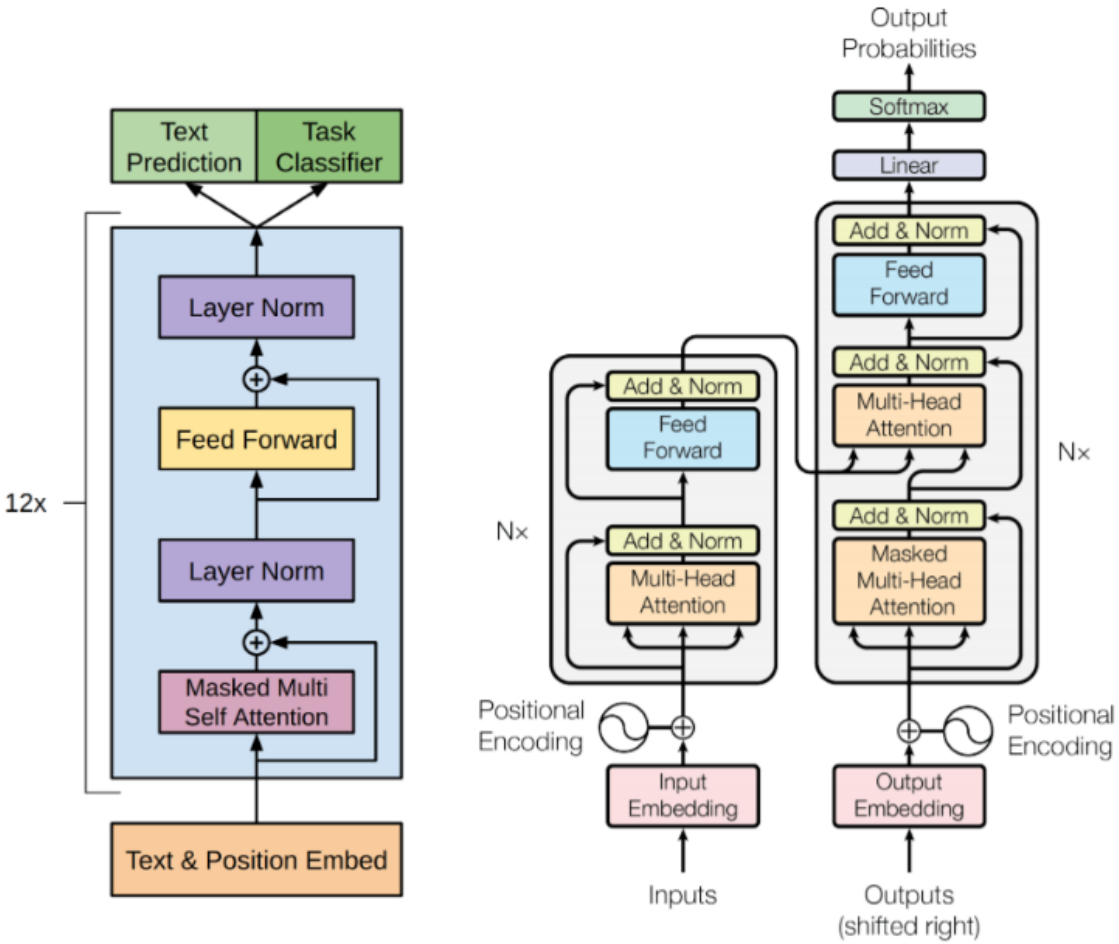
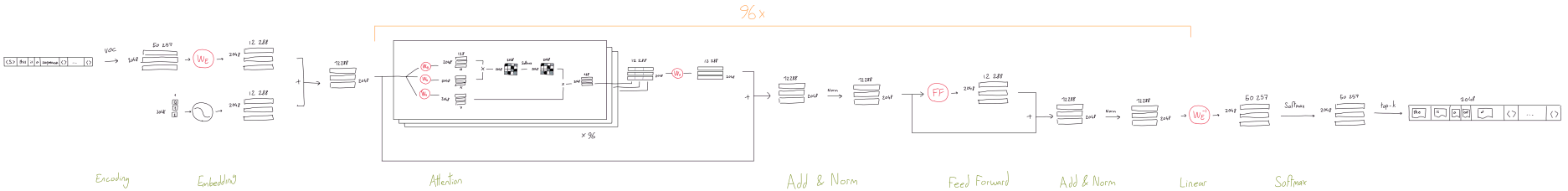


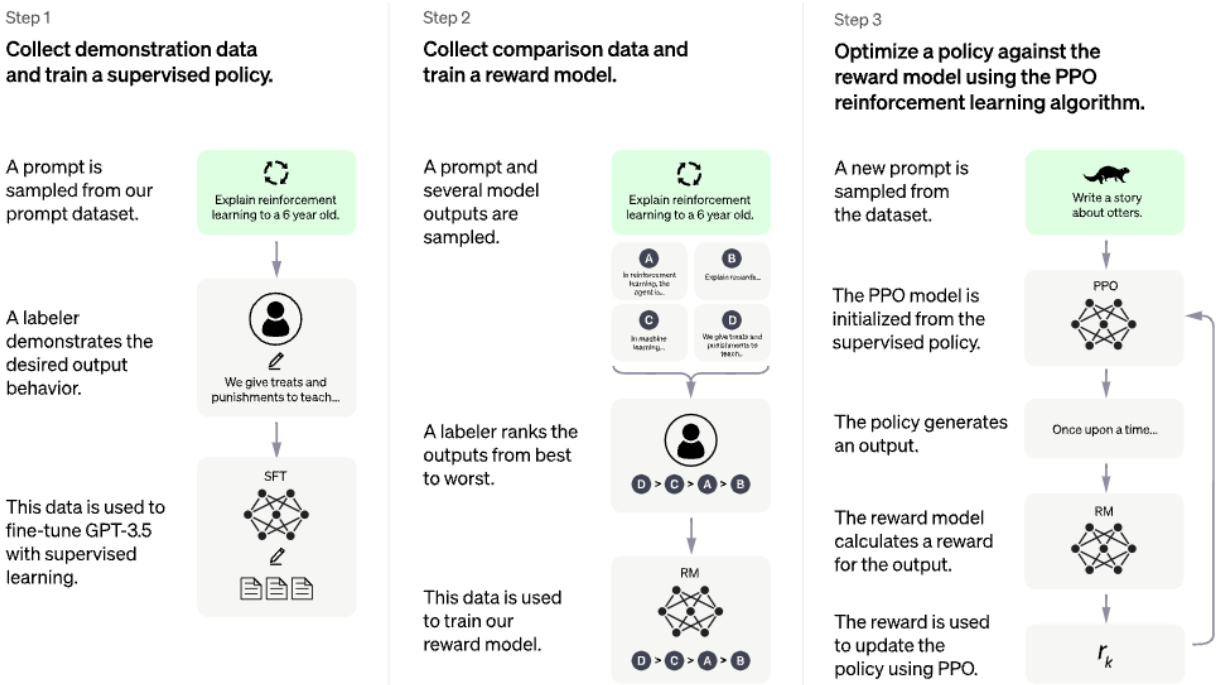
Figure 1: The Transformer - model architecture.

Transformer



GPT: Training

- GPT wurde semi-supervised trainiert:
  - Language Model zunächst aus nicht-annotierten Text-Daten (ca. 45 TB) gelernt
  - Dann mit menschlicher Unterstützung verfeinert



https://cdn.openai.com/chatgpt/draft-20221129c/ChatGPT_Diagram.svg