

KI im Alltag – sehen und verstehen

Prof. Dr. Matthias Wendlandt

Institut für Informatik
Justus-Liebig-Universität Gießen

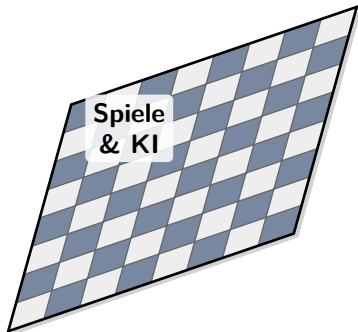
Übersicht

- Was ist KI?
- Neuronale Netze
- Klassische KI
- Machine Learning
- Was bedeutet das für uns als Gesellschaft?
- Large Language Modells

Künstliche Intelligenz

- Automatisierung von intelligentem Verhalten
- Maschinelles Lernen

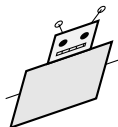
Künstliche Intelligenz – Entwicklung



Expertensysteme



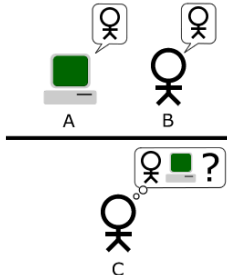
Robotik



Wann ist ein Computer intelligent?

Der Turing Test

- Der Turing Test wurde 1950 von Alan Turing vorgestellt. Das Ziel des Tests ist es, die Intelligenz einer Maschine festzustellen.
- Ein Fragesteller interagiert per Chat sowohl mit einem Computer als auch mit einer Person.
- Kann der Fragesteller nicht unterscheiden, wer Computer und wer Mensch ist, so gilt der Computer als intelligent.



Turin Test – Chat GPT

Stell Dir vor, Du bist ein Kind was 11 Jahre alt ist und beantworte mir folgende Fragen. Antworte kurz.

1. Was ist $17 * 21$?
2. Welchen Fussballer magst Du am liebsten?
3. Magst Du Schule?
4. Welche Musik magst Du gerne?
5. Was ist die Wurzel aus 2?
6. Was ist Dein Lieblingsgetränk?
7. Was ist Dein Lieblingsessen?
8. Was denkst Du über Krieg?
9. Was ist Dein Lieblingsfilm?
10. Was machst Du gerne in Deiner Freizeit?
11. Magst Du Harry Potter?
12. Welche Tiere gibt es bei euch zu Hause?
13. Welches Spiel spielst Du gerne auf der Playstation?
14. Erzähl mal einen kurzen Witz.
15. Was bedeutet Nachhaltigkeit?

Künstliche Intelligenz – Hauptströmungen

Klassische KI

- Suche (BFS/DFS/ A^*)
- Wissensrepräsentation & Logik
- Expertensysteme
- Regelbasierte Sys.
- ...

Probabilistische KI

- Bayes-Netze
- Markov-Modelle (HMM)
- MDPs/POMDPs
- Probabilistische Inferenz
- Graphical Models

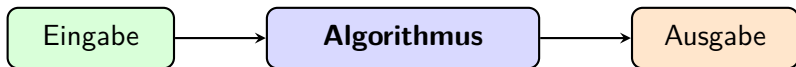
(oft Brücke angesehen)

Statistische KI

- Maschinelles Lernen (ML)
 - Deep Learning, LLMs
 - Optimierung, Regularisierung
 - ...
-

Was ist ein Algorithmus?

- Eine präzise Abfolge von Handlungsschritten zur Lösung eines Problems.
- Bei gleicher Eingabe liefert ein Algorithmus stets die gleiche Ausgabe (Determinismus).



Deterministisch: gleiche Eingabe \Rightarrow gleiche Ausgabe

Neuronale Netze

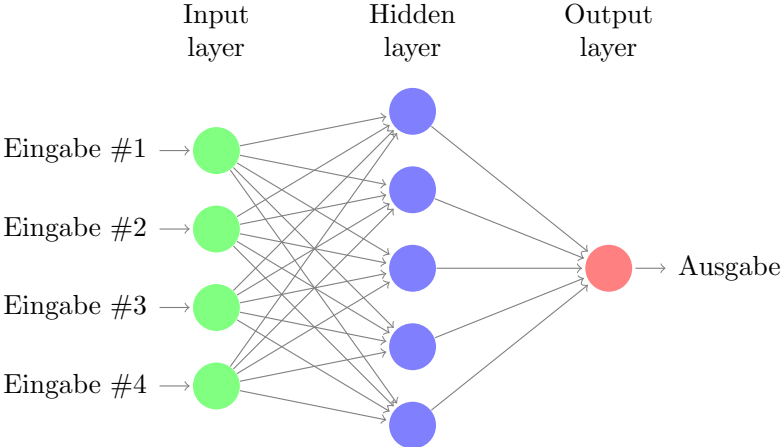
- Neuronale Netze sind ein Konzept der Informatik, die das Verhalten und die Arbeitsweise des menschlichen Gehirns versuchen zu simulieren.
- Im Wesentlichen bestehen neuronale Netze aus einer großen Anzahl von Neuronen, dessen Verhalten durch Lernalgorithmen trainiert werden.

Schichten eines neuronalen Netzes

Sei $N = (U, C)$ ein neuronales Netz. So gibt es drei Arten von Schichten:

- *Die Eingabeschicht* (input units, $U_{in} \subseteq U$) - Die Schicht empfängt Signale von der Außenwelt.
- *Die verborgenen Schichten* (hidden units, $U_{hidden} \subset U$) - Diese Art von Schicht empfängt keine Signale von außen und gibt auch keine Signale an die Außenwelt ab.
- *Die Ausgabeschicht* (output units, $U_{out} \subseteq U$) - Die Schicht gibt Signale an die Außenwelt ab.

Künstliches Neuron



Propagierungsfunktion

Definition

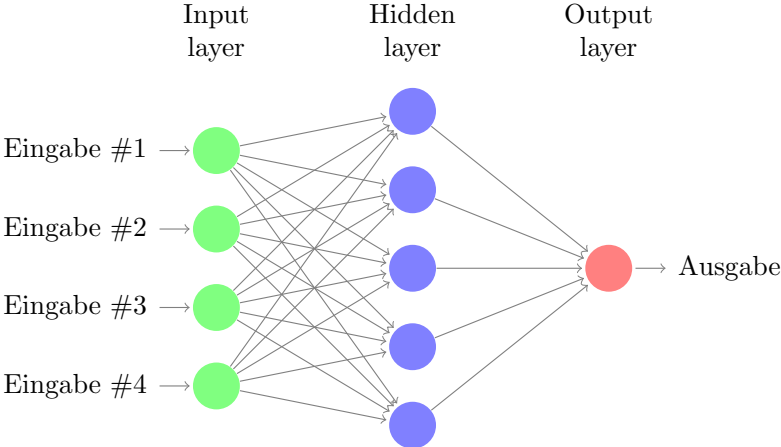
Ein Neuron n besteht aus:

- **Eingabeinformationen** $i_0 \dots i_k$ sind die Erregungen, die von den Vorgängerneuronen empfangen werden. Es sei n ein Neuron, dann wird mit $|n|$ die Anzahl der Eingabeinformationen bezeichnet.
- **Propagierungsfunktion** net ist die Funktion, die auf die Eingabeinformationen $i_0 \dots i_k$ Gewichte $w_0 \dots w_k$ anwendet:

$$net(n) = \sum_{j=0}^{|n|} w_j i_j$$

wird Propagierungsfunktion genannt. Die Ausgabe der Funktion net für ein Neuron n wird häufig als net_n geschrieben.

Künstliches Neuron



Aktivierungsfunktion

Definition

Ein Neuron n besteht aus:

- **Aktivierungsgrad** - Der Aktivierungsgrad a_n des Neurons n spiegelt die Erregung des Neurons wieder.
- **Aktivierungsfunktion** f_{act} - Eine Funktion, die den Aktivierungsgrad des Neurons bestimmt:

$$f_{act}(net_n) = a_n$$

- **Ausgabefunktion** f_{out} - Die Ausgabefunktion ermittelt aus dem Aktivierungsgrad a_n den Ausgabewert o_n (der für alle Nachfolger gleich ist) und gibt ihn an die nachfolgenden Neuronen weiter.

$$f_{out}(a_n) = o_n$$

Ausgabe und Aktivierungsfunktionen

- Die Aktivierungsfunktion und auch die Ausgabefunktion können beliebig komplexe Funktionen sein, was jedoch im Grunde nicht gewollt ist.
- Die neuronalen Netze bekommen ihre Stärke dadurch, dass sie sehr einfach aufgebaut sind, dafür aber aus einer Vielzahl solcher einfachen Einheiten bestehen.
- Für die Ausgabefunktion wird meistens die Identitätsfunktion gewählt, für die Aktivierungsfunktion die logistische Funktion oder die binäre Funktion.

Ausgabe und Aktivierungsfunktionen

→ Die Identitätsfunktion

$$f_{out}(a_n) = a_n$$

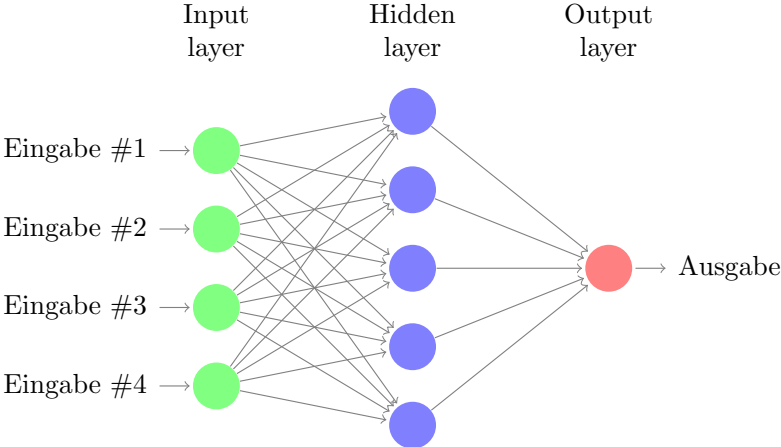
→ Lineare Aktivierungsfunktion mit Schwelle

$$f_{act}(net_n) = \begin{cases} m \cdot net_n + b & \text{falls } net_n \geq 0, \\ 0 & \text{sonst} \end{cases}$$

→ binäre Funktion (gegebenenfalls auch mit einer anderen Schwelle)

$$f_{act}(net_n) = \begin{cases} 1 & \text{falls } net_n \geq 0, \\ 0 & \text{sonst} \end{cases}$$

Künstliches Neuron



Was ist künstliche Intelligenz – Ein kleines Experiment

Was ist künstliche Intelligenz – Ein kleines Experiment

Aktivitäten zu künstlicher Intelligenz ohne Strom (Lindner Seegerer 2019)

Was ist künstliche Intelligenz – Ein kleines Experiment

	Dreieckige Form?	Viereckige Form?	Runde Form?
<i>Haus</i>	Ja	Ja	Nein
<i>Auto</i>	Nein	Ja	Ja
<i>Katze</i>	Ja	Nein	Ja

Aktivitäten und Unterrichtsmaterial zu künstlicher Intelligenz ohne Strom (Lindner Seegerer 2019)

Klassische KI

Problemlösen durch Suchen

- Viele Probleme in der künstlichen Intelligenz können auf eine Suche in einem Zustandsraum zurückgeführt werden.
- Ganz ähnlich führt auch der Mensch selbst eine Problemlösung durch.
- Beispielsweise denkt ein Schachspieler einige Züge voraus.
- Dabei erzeugt er sich in Gedanken mögliche zukünftige Spielformationen, bewertet diese und sucht sich die geeignetste Lösung heraus.
- Zur Lösung eines Problems mittels Zuständen müssen die Zustände zuerst definiert werden.
- Dann folgt die Definition von Anfangszuständen und die Definition von Endzuständen.

Adversariale Suche

- Suchprobleme, in denen sich die Ziele von verschiedenen Parteien konkurrierend gegenüber stehen, werden unter dem Namen *adversariale Suche* zusammengefasst.
- Betrachten wir zuerst Spiele mit zwei Spielern, die als *min* und *max* bezeichnet werden.
- *max* beginnt immer das Spiel.
- Daraufhin ziehen die beiden Parteien abwechselnd.
- Am Spielende bekommt der Spieler, der gewonnen hat, eine Belohnung und der andere eine Bestrafung.

Adversariale Suche

Schauen wir uns zunächst ein Beispielspiel an.

Der Minimax Algorithmus

- Der Minimax Algorithmus durchläuft einen Suchbaum bis in die Blätter.
- Dass Blätter erreicht worden sind, bedeutet nicht zwingend, dass eine Spielendeposition erreicht worden ist.
- Eine Spielendeposition zu erreichen, erfordert bei den meisten Spielen eine sehr große Suchtiefe.
- So werden in der Regel nur die nächsten n Züge vorausberechnet.
- Hier wird eine Bewertung der durch diesen Pfad entstehenden Spielsituation getroffen.
- Beispielsweise die oben beschriebene $\{1, -1, 0\}$ Bewertung für gewonnen, verloren oder unentschieden.

Der Minimax Algorithmus

- Die Bewertungen können aber auch ganz anders getroffen werden.
- So könnte beim Schach (hierbei ist es im Allgemeinen zu aufwändig in Suchtiefen zu kommen, die keinen Sieger hervorbringen) Spielpositionen bewertet werden.
- Die Ebenen des Baums werden abwechselnd in *max* und *min* Ebenen eingeteilt.
- So wird die Wurzel Ebene 0 *max* zugeordnet und die Ebene 1 *min* zugewiesen.
- Im Allgemeinen ist somit die Ebene k mit *max* beschriftet, falls $k \bmod 2 == 0$, andernfalls ist sie mit *min* beschriftet.

Der Minimax Algorithmus

Computerprogramme KI – Klassisch und Modern

Klassisch

→ Input → Programm → Output

Modern

→ Sample → ML-Algorithmus → Modell

→ Input → Modell → Output

Maschinelles Lernen

- überwachtes Lernen (Supervised Learning)
- unüberwachtes Lernen (Unsupervised Learning)
- verstärkendes Lernen (Reinforcement Learning)

Überwachtes Lernen – Beispiel Entscheidungsbaum

Verstärkendes Lernen – Beispiel Spiel

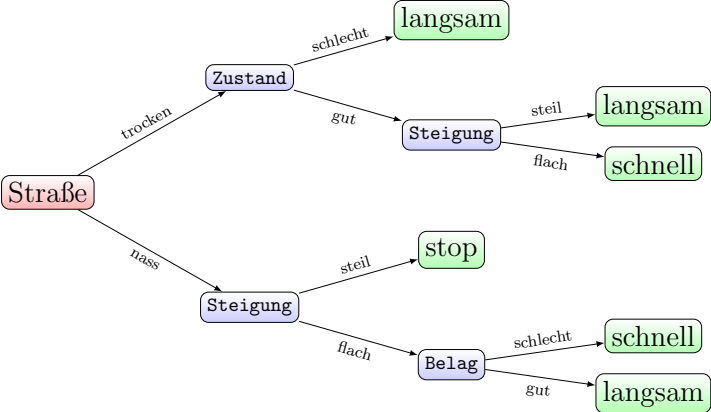
<https://www.youtube.com/watch?v=qv6UVOQ0F44>

<https://www.i-am.ai/build-your-own-ai.html>

Entscheidungsbäume

- Ein Entscheidungsbaum besteht aus einem Baum im mathematischen Sinne, also einem gerichteten, schwach zusammenhängenden und kreisfreien Graphen.
- Jeder Knoten des Baumes ist entweder ein Ergebnis oder stellt eine Frage, wobei die von ihm ausgehenden Kanten mögliche Antworten repräsentieren.
- Die Fragen müssen nicht binär sein, aber zu jeder Frage muss es immer genau eine Antwort, also eine ausgewählte Kante geben.
- In diesem Sinne müssen die Ausgangskanten eine Partitionierung des Antwortraums der gestellten Frage sein.
- Die Anwendung von Entscheidungsbäumen funktioniert so, dass man beim Wurzelknoten anfängt und sich Frage für Frage durch den Baum arbeitet.

Entscheidungsbäume



Was wird bei KI *trainiert* – und was passiert später bei der Nutzung?

Kernidee

- Beim **Training** lernt ein Modell aus **Trainingsdaten** Muster und Zusammenhänge.
- Die Daten sind zunächst „nur“ Beispiele — sie werden *nicht* wörtlich gespeichert, sondern statistisch verallgemeinert.

Spätere Verwendung (Inference)

- Bei der Nutzung können ganz **neue Fragestellungen/Szenarien** gestellt werden, die im Training so nicht vorkamen.
- Das Modell **generalisiert** das Gelernte auf diese neuen Eingaben.

Was wird bei KI *trainiert* – und was passiert später bei der Nutzung?

Kernidee

- Beim **Training** lernt ein Modell aus **Trainingsdaten** Muster und Zusammenhänge.
- Die Daten sind zunächst „nur“ Beispiele — sie werden *nicht* wörtlich gespeichert, sondern statistisch verallgemeinert.

Wichtige Einschränkungen

- **Bias und Verzerrungen**: Trainingsdaten können gesellschaftliche/strukturelle **Vorurteile** enthalten.
- **Datenqualität & Abdeckung**: Rauschen, Lücken oder einseitige Quellen führen zu fehlerhaften oder unfairen Ergebnissen.
- **Domänenverschiebung**: Wenn die realen Anfragen stark vom Training abweichen, sinkt die Zuverlässigkeit.

Was wird bei KI *trainiert* – und was passiert später bei der Nutzung?

Kernidee

- Beim **Training** lernt ein Modell aus **Trainingsdaten** Muster und Zusammenhänge.
- Die Daten sind zunächst „nur“ Beispiele — sie werden *nicht* wörtlich gespeichert, sondern statistisch verallgemeinert.

Gegenmaßnahmen (kurz)

- Kuratierte/ausgewogene Datensätze, **Bias-Audits**, Evaluation auf vielfältigen Tests,
- Techniken wie **Debiasing**, **Re-Weighting**, **Human-in-the-loop** und Richtlinien bei der Nutzung.

Robotik, künstliche Intelligenz und Ethik

- Autonomes Fahren ist eines der Hype-Themen.
- Doch welche Probleme erwarten uns damit?

Robotik, künstliche Intelligenz und Ethik

- Wer hat Schuld, wenn ein Unfall passiert?
- Wie reagiert das Fahrzeug, wenn es keinen guten Ausweg gibt?

Was würdest du machen?

KI – Gefahren und Probleme

- Massenüberwachung: Es braucht keine Menschen mehr um Chats, Internetaktivitäten, Kameras, Buchungen etc. zu überwachen.
- (Wahl-)Manipulation: Perfekt auf die eigenen Weltanschauungen zugeschnittene Wahlwerbung.
- Autonome Waffen: Entscheidung zu töten wird automatisch getroffen.
- Monopolbildung: Nur wenige Unternehmen (Google, Meta, Microsoft, etc.) besitzen die Daten für Entwicklung leistungsfähiger KI-Systeme

KI – Chancen und/oder Probleme

- Routine und regelbasierte Aufgaben werden verstärkt von KI übernommen.
- Der Mensch als Prüfer und nicht mehr Erzeuger von Inhalten. Generieren kreativer Vorschläge durch maschinelles Lernen.
- Was bleibt: soziale Arbeit, Entscheidungen treffen, investigative Recherche/Forschung, Handwerk, Hilfsarbeiten

Ethik der KI – Grundfragen

- Welche Auswirkungen haben KI-Systeme auf Menschen und Gesellschaft?
- Wer trägt Verantwortung für Entscheidungen automatisierter Systeme?
- Wie gehen wir mit möglichen Risiken um?



Ethik der KI – Herausforderungen

- **Bias & Diskriminierung:** Verzerrte Daten führen zu unfairen Entscheidungen.
- **Transparenz:** Black-Box-Modelle erschweren Nachvollziehbarkeit.
- **Datenschutz:** Umgang mit sensiblen Informationen.
- **Autonomie:** Wie viel Entscheidungsfreiheit darf man an Maschinen abgeben?

Ethik der KI – Leitlinien

- **Fairness:** Gleichbehandlung und Vermeidung von Diskriminierung.
- **Transparenz & Erklärbarkeit:** Modelle sollten nachvollziehbar sein.
- **Verantwortung:** Klare Zuständigkeiten bei Fehlern und Schäden.
- **Nachhaltigkeit:** Ressourcenschonender Einsatz von KI.

„Ethik ist kein Hindernis,
sondern ein Rahmen für vertrauenswürdige KI.“

Word2Vec – Auf dem Weg zum Transformer

Ein grundlegendes Problem der natürlichen Sprachverarbeitung:

Unterschiedliche Formulierungen können dieselbe Bedeutung transportieren. So verweisen etwa die folgenden Überschriften mit hoher Wahrscheinlichkeit auf dasselbe Ereignis – obwohl die Wortwahl völlig verschieden ist:

Beispiel 1

„Merz besucht Macron in Paris“

Beispiel 2

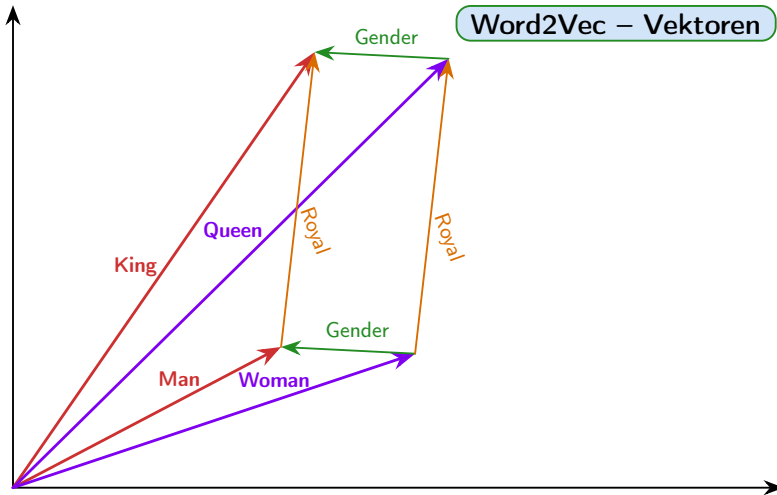
„Der Bundeskanzler zu Besuch in Frankreich“

Semantische Merkmale von Konzepten

	King	Queen	Man	Woman	Horse	Castle	Crown
human	1.0	1.0	1.0	1.0	0.0	0.0	0.0
masculinity	0.95	0.05	0.95	0.05	0.0	0.0	0.0
femininity	0.05	0.95	0.05	0.95	0.0	0.0	0.0
royal_role	1.0	1.0	0.0	0.0	0.0	0.0	0.0
wears_crown	0.9	0.9	0.0	0.0	0.0	0.0	0.0
royal	1.0	1.0	0.1	0.1	0.3	0.8	1.0
animacy	1.0	1.0	1.0	1.0	1.0	0.0	0.0
agency	0.9	0.9	0.7	0.7	0.4	0.0	0.0
concreteness	0.7	0.7	0.9	0.9	1.0	1.0	1.0
size	0.6	0.55	0.6	0.55	0.8	1.0	0.2
mobility	0.8	0.8	0.9	0.9	1.0	0.0	0.0
objectness	0.0	0.0	0.0	0.0	0.0	1.0	1.0

Word2Vec – Auf dem Weg zum Transformer

King-Man+Woman=Queen



Automatische Ermittlung von Wortvektoren

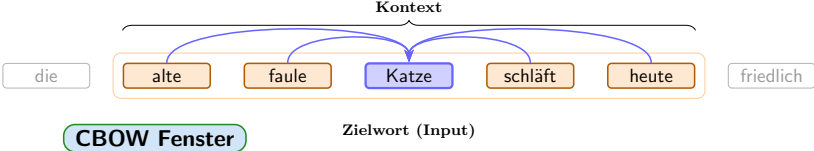
- In der Praxis ist es kaum möglich, für alle Wörter eines Textkorpus manuell ein einheitliches und konsistentes Set vordefinierter Eigenschaften zu erstellen.
 - ▶ Der Aufwand wäre enorm.
 - ▶ Eigenschaften müssten für unterschiedliche Korpora jeweils neu angepasst werden.

- Stattdessen werden Wortvektoren automatisch mithilfe von **maschinellern Lernen**
 - genauer: **neuronalen Netzen** – ermittelt.

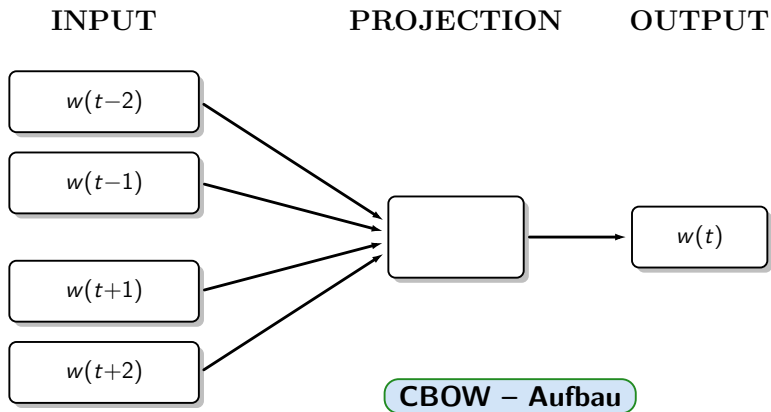
- Das Modell „lernt“, wie Wörter im Kontext anderer Wörter auftreten, und leitet daraus ihre Vektorrepräsentationen ab.

- Zwei der bekanntesten Ansätze: **CBOW** (Continuous Bag of Words) und **Skip-Gram**.

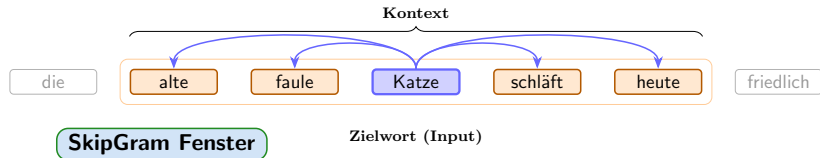
Idee CBOW



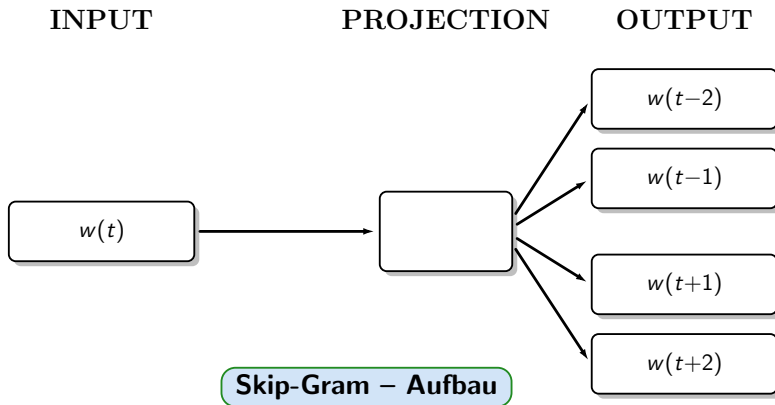
CBOW Neuronales Netz



Idee Skip Gram



Skip Gram Neuronales Netz

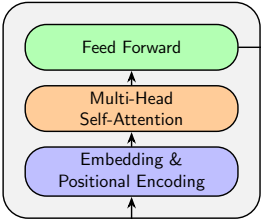


Transformer – Kurze Einführung

- **Seit 2017** das zentrale Architekturprinzip moderner Sprachmodelle (z. B. BERT, GPT, T5).
- Kernidee: **Self-Attention** anstelle rekurrenter oder konvolutionaler Strukturen.

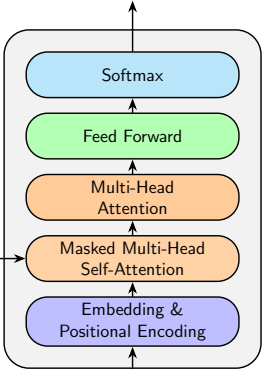
Das Transformer Modell

Transformer Modell



Inputs

Encoders



Targets

Decoders

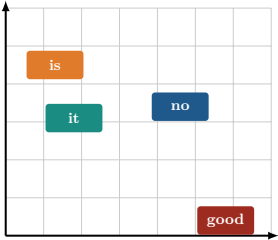
Positional Encoding

- Transformer haben **keine eingebaute Reihenfolge-Information** (im Gegensatz zu RNNs).
- Lösung: **Positional Encodings** – Vektoren, die die Position eines Tokens in der Sequenz codieren und zu den Wortvektoren addiert werden.
- Eigenschaften:
- Effekt: Modell „weiß“, welches Wort am Anfang, in der Mitte oder am Ende der Sequenz steht.

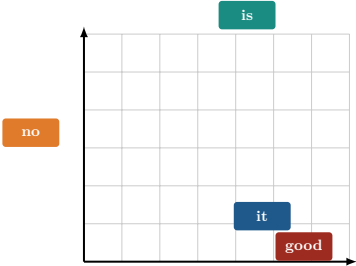
Positional Encoding – Beispiel

Positional Encoding

no it is good



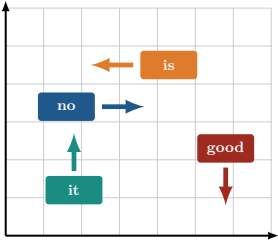
it is no good



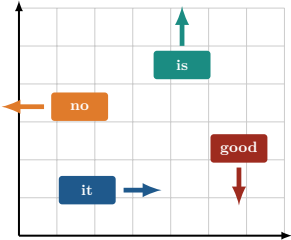
Positional Encoding – Beispiel

Positional Encoding

no it is good



it is no good



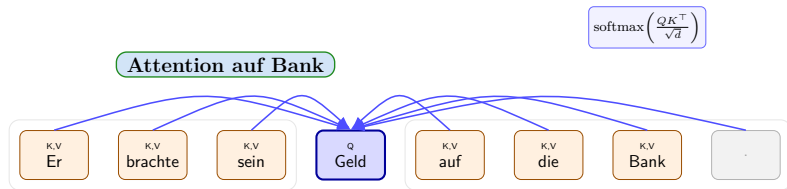
Self-Attention

Satz: Er brachte sein Geld auf die Bank.

Self-Attention – Einführung

- Kernbaustein des **Transformers**.
- Idee: Jedes Wort einer Sequenz kann sich flexibel auf andere Wörter „konzentrieren“.
- Mechanismus:
 1. Jedes Eingabe-Token wird in drei Vektoren projiziert: **Query (Q), Key (K), Value (V)**.
 2. Für jedes Token wird die **Ähnlichkeit** zwischen Query und allen Keys berechnet (QK^T).
 3. Softmax normalisiert diese Scores zu **Aufmerksamkeitsgewichten**.
 4. Ausgabe = gewichtete Summe der Values.
- Vorteile:
 - ▶ Erfasst **Beziehungen** zwischen allen Wörtern (z. B. lange Abhängigkeiten).
 - ▶ Parallelisierbar und effizienter als rekurrente Modelle.

Attention – Idee

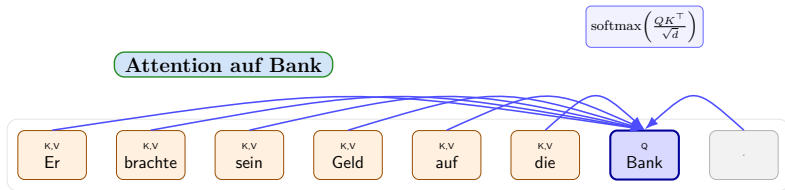


Self-Attention (Single Head)

Query: „Bank“

Satz: *Er brachte sein Geld auf die Bank.*

Attention – Idee



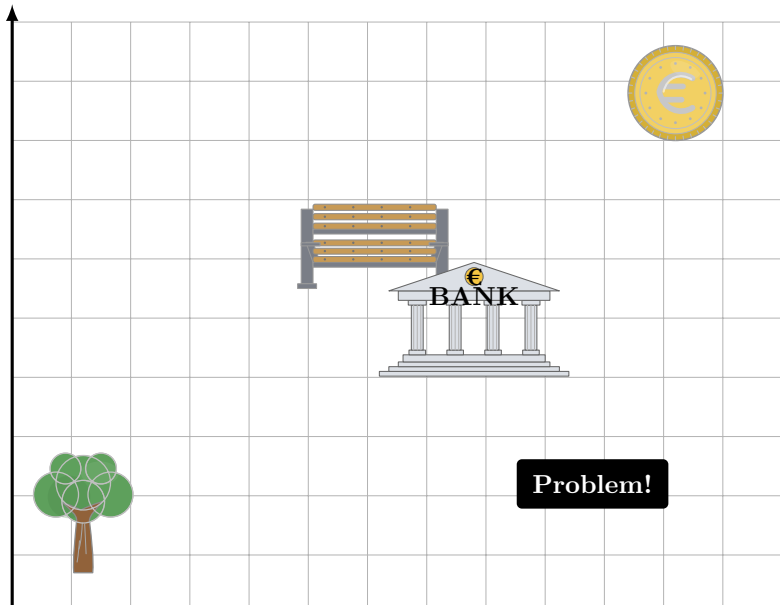
Self-Attention (Single Head)

Query: „Bank“

Satz: *Er brachte sein Geld auf die Bank.*

Attention – Wirkung

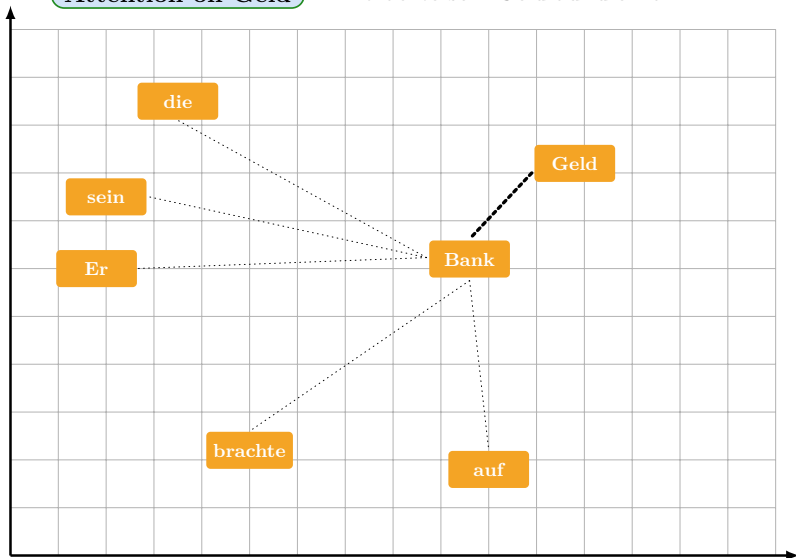
Embeddings



Attention – Wirkung

Attention on Geld

Er brachte sein Geld auf die Bank.



Self-Attention – Beispiel 1

Satz: The pizza is made in the oven, and it tastes good.

- Das Pronomen `it` richtet starke Aufmerksamkeit auf `pizza`.
- Das Adjektiv `good` bezieht sich über `tastes` auf das Subjekt.
- \Rightarrow Der Transformer lernt: `it = pizza`.

Self-Attention – Beispiel 2

Satz: The guy is going to the bank and collects money.

- bank ist mehrdeutig (Flussufer vs. Finanzinstitut).
- Der Kontext collects money erhält hohe Aufmerksamkeit in Richtung bank.
- ⇒ Auflösung: bank = Finanzinstitut.

Feed-Forward-Netze im Transformer

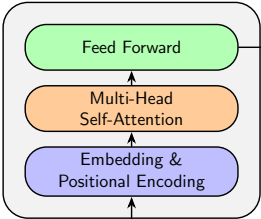
- Nach der **Self-Attention** besitzt jedes Token eine neue Kontextrepräsentation.
- Ein **Feed-Forward-Netzwerk** (2 lineare Schichten + Aktivierung) wird auf jedes Token separat angewendet:

$$\text{FFN}(x) = W_2 \sigma(W_1 x + b_1) + b_2$$

- Funktion:
 - ▶ transformiert die Information nicht nur linear, sondern nichtlinear
 - ▶ erlaubt es, komplexere Muster zu modellieren
 - ▶ wirkt wie eine „Veredelung“ der durch Attention gewonnenen Repräsentationen

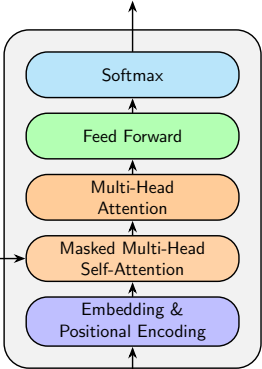
Transformer Modell

Transformer Modell



Inputs

Encoders



Targets

Decoders

Danke für Ihre Aufmerksamkeit!