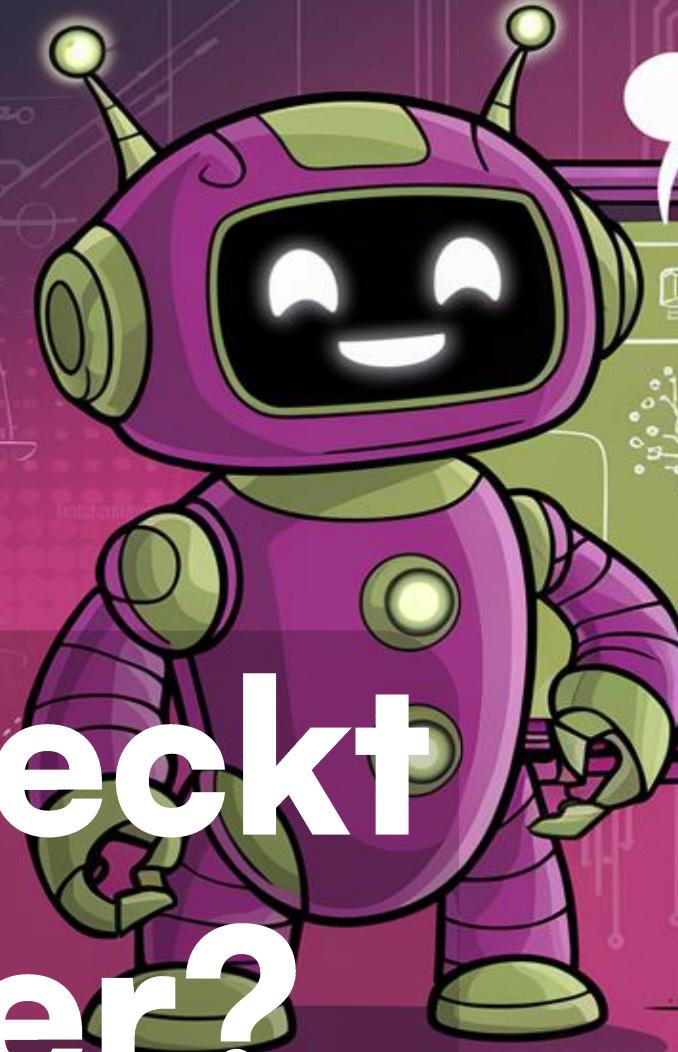
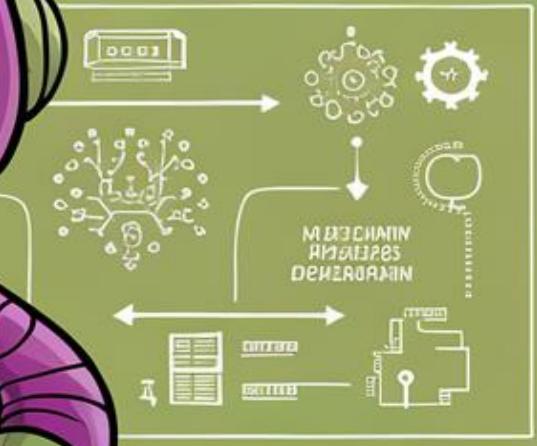


Was steckt dahinter?



GRUNDLAGEN DER KI





Jakow Smirin
Chief AI & Operations Officer
STARTPLATZ AI Hub Duesseldorf

STARTPLATZ
AI HUB



Lukas Stratmann
Chief AI & Operations Officer
STARTPLATZ AI Hub Koeln

AlexNet @ IMAGENET



ImageNet Classification with Deep Convolutional Neural Networks

By Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton

Abstract

We trained a large, deep convolutional neural network to classify images in the ImageNet LSVRC-2010 contest into the 1000 different classes. On the test data, we achieved top-1 and top-5 error rates of 15.3% and 5.1%, which are 14.1% and 11.8% better than the previous state-of-the-art. The neural network, which has 60 million parameters and 650,000 neurons, consists of five layers of feature detectors followed by max pooling layers, and three fully connected layers with a final 1000-way softmax. To make training faster, we used non-uniform learning rates and a hierarchical structure of the convolution operation. To reduce overfitting in the fully connected layers we employed a recently developed regularization method called “dropout” that proved very effective. We also entered a variant of this model in the ILSVRC-2012 competition and achieved a winning top-5 test error rate of 15.3%, compared to 26.2% achieved by the second-place team.

1. PROLOGUE

that were widely investigated in the 1980s. These networks used multiple layers of feature detectors that were all learned from the same input. In the early 1990s, LeCun et al. had hypothesized that a hierarchy of such feature detectors would provide a robust way to recognize objects but they had no idea how to implement it. This was a major problem at the time because several different research groups discovered that multiple layers of feature detectors could be easily trained using a simple gradient descent algorithm called backpropagation^{1,2,3,4,5,6} to compute, for each image, how the classification performance of the whole network depended on the values of the weights in each layer.

Backpropagation worked well for a variety of tasks, but in the 1990s it did not live up to the very high expectations of its proponents. In particular, it failed to learn to classify thin networks with many layers and these were precisely the networks that should have given the most impressive results.

It was only in the late 1990s that it was realized that a deep neural network from random initial weights was just too difficult. Twenty years later, we know what went wrong: for deep neural networks to shine, they needed far more labeled data and longer more computation.



Im Jahr 2012 erzielte das Team um **Geoffrey Hinton** und **Ilya Sutskever** einen bedeutenden Durchbruch im Bereich des Maschine Learning.

Sie entwickelten ein tiefes **neuronales Netzwerk**, bekannt als **AlexNet**, das die Bildklassifizierungsaufgabe des **ImageNet-Wettbewerbs** mit beeindruckender Genauigkeit löste.

Was ist AlexNet ?

Ein bahnbrechendes Convolutional Neural Network (CNN), das 2012 von Geoffrey Hinton und seinem Team entwickelt wurde. Mit AlexNet begann die Ära moderner Deep Learning-Modelle.

Die Zauberformel

ReLU (Rectified Linear Unit)
Beschleunigte das Training
durch einfache
Aktivierungsfunktion.

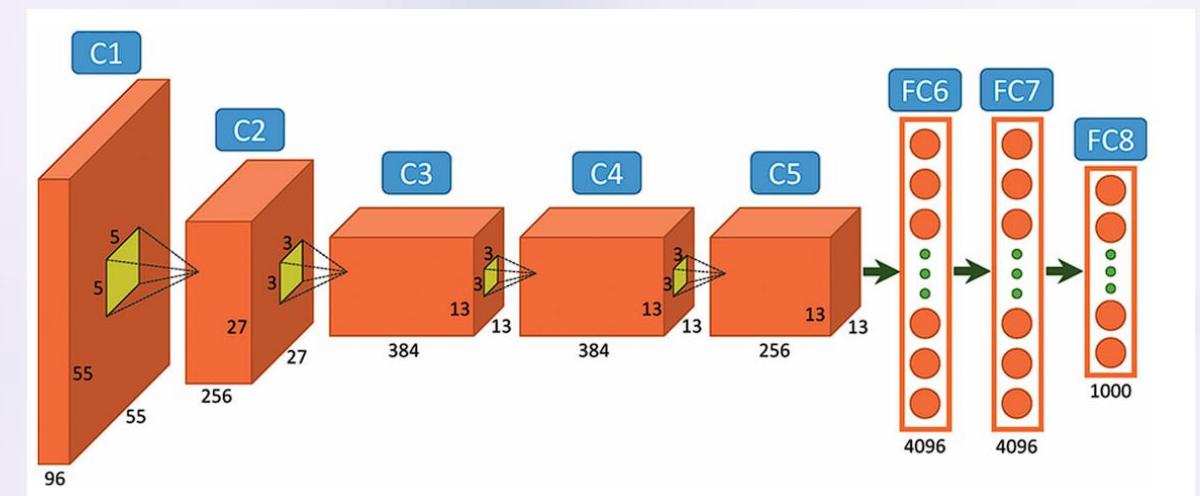
GPU-Nutzung
Ermöglichte superschnelles
Training auf großen
Datensätzen.

Overlapping Pooling
Reduzierte Fehler durch
überlappende Bereiche.

Dropout
Verhinderte Überanpassung
des Modells (Overfitting).

8 Schichten Architektur

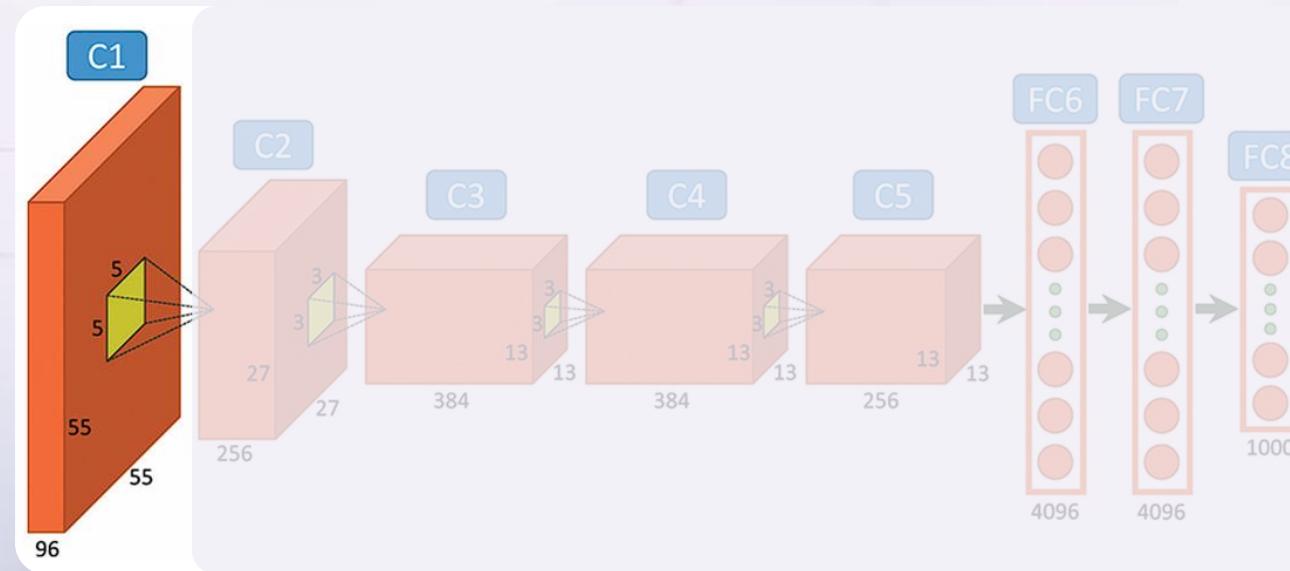
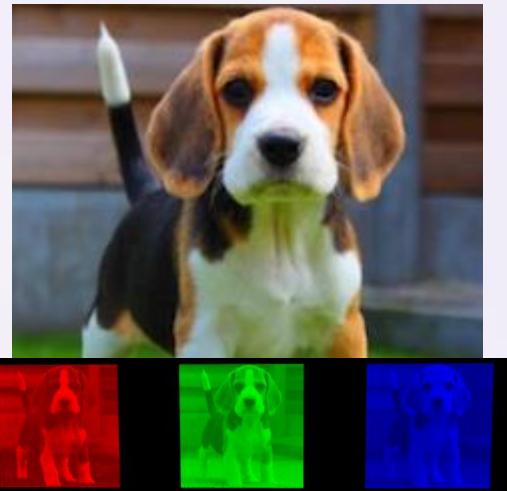
5 Convolutional Layers (C1—C5).
3 Fully Connected Layers (FC6—FC8).



Wie funktioniert AlexNet?

AlexNet nutzt Convolutional Layers, Fully Connected Layers und Aktivierungsfunktionen, um Bilder Schritt für Schritt zu analysieren. Es erkennt Muster wie Kanten, Formen und komplexe Objekte.

Schritt-für-Schritt-Erklärung:



Layer 1 (C1): Erkennung von Kanten und Farben

Kernels: 96 Filter (11x11 Pixel groß).

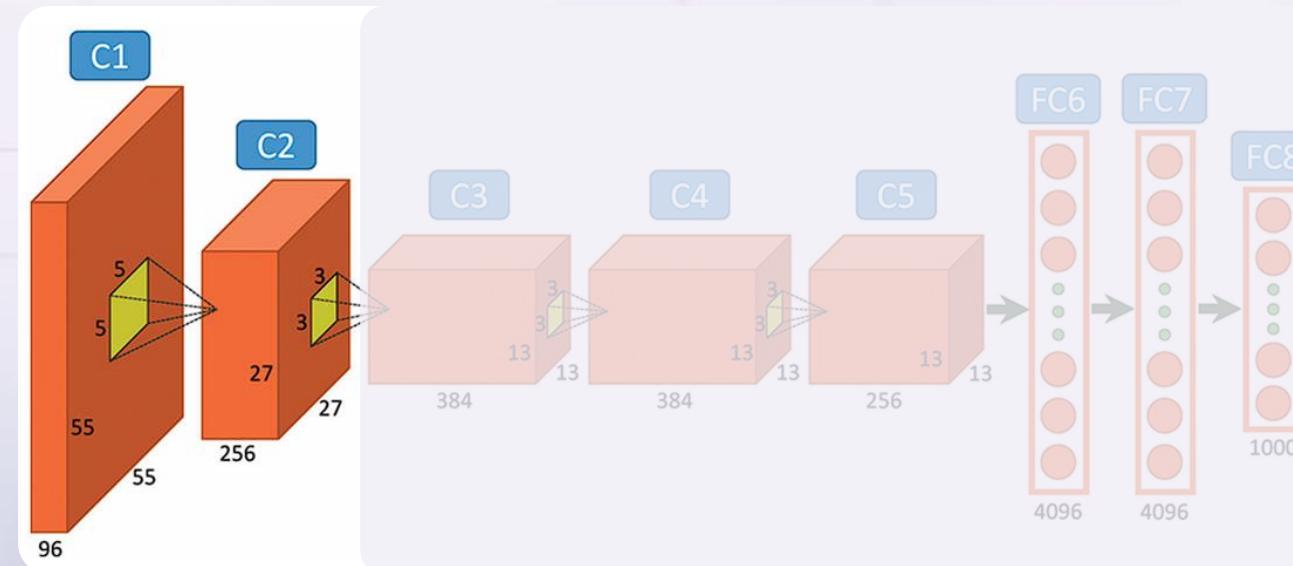
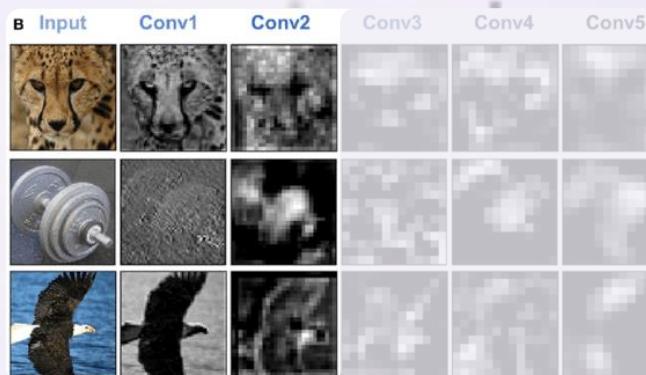
Ergebnis: Kanten und Farbkontraste werden erkannt (z. B. Umrisse der Ohren).

ReLU: Aktiviert nur positive Werte, ignoriert irrelevante Daten.

Wie funktioniert AlexNet?

AlexNet nutzt Convolutional Layers, Fully Connected Layers und Aktivierungsfunktionen, um Bilder Schritt für Schritt zu analysieren. Es erkennt Muster wie Kanten, Formen und komplexe Objekte.

Schritt-für-Schritt-Erklärung:



Layer 2 (C2): Kombination von Mustern

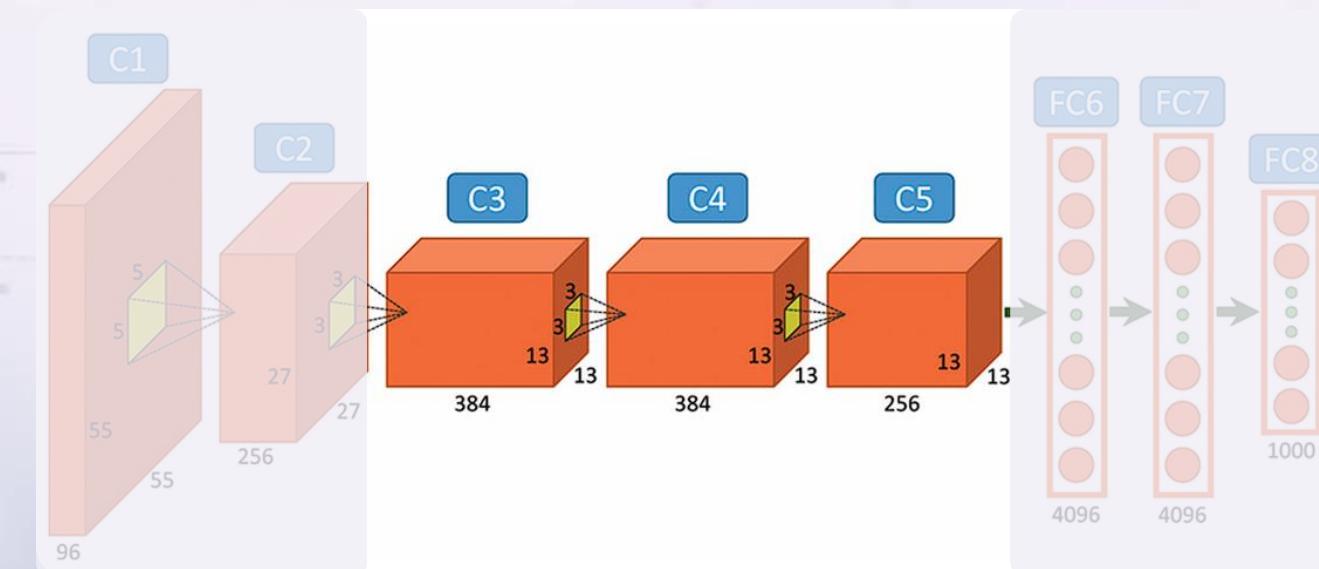
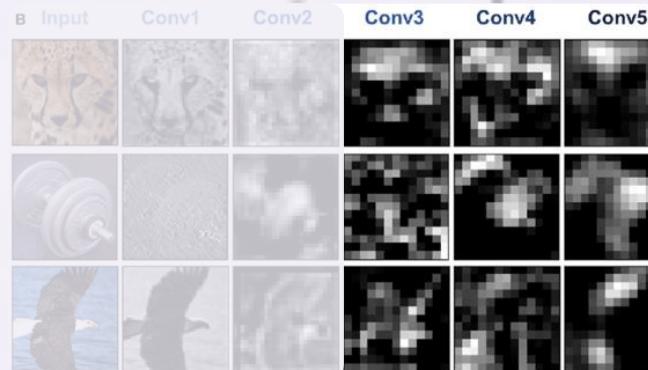
Kernels: 256 Filter (5x5).

Ergebnis: Einfachere Formen wie "Ohren" und "Schnauze" werden kombiniert.

Wie funktioniert AlexNet?

AlexNet nutzt Convolutional Layers, Fully Connected Layers und Aktivierungsfunktionen, um Bilder Schritt für Schritt zu analysieren. Es erkennt Muster wie Kanten, Formen und komplexe Objekte z. B. den Kopf eines Hundes.

Schritt-für-Schritt-Erklärung:



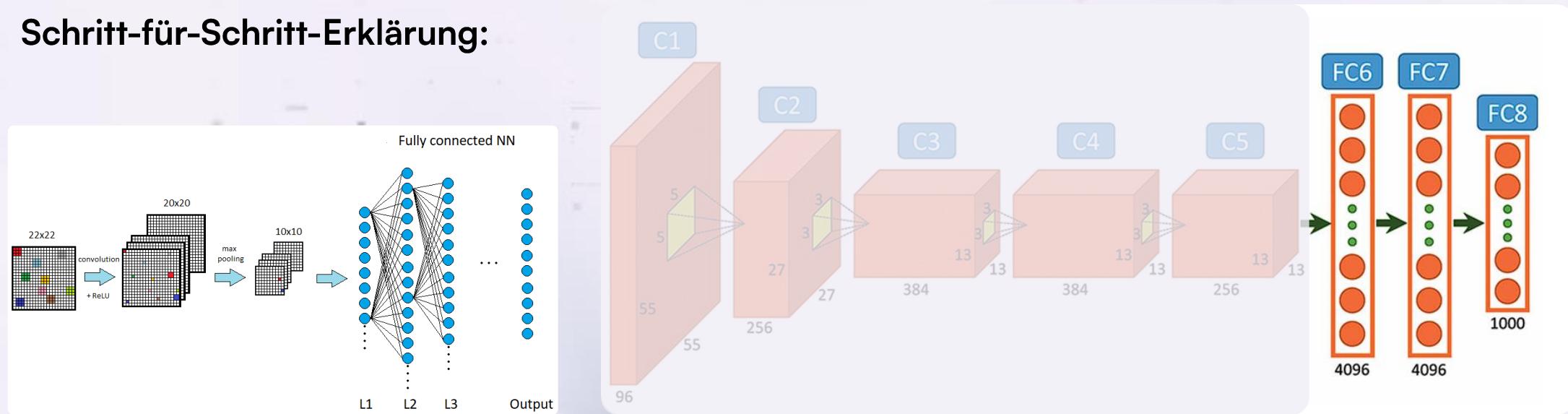
Layer 3—5 (C3—C5): Komplexere Muster

Ergebnis: Der Kopf des Hundes wird erkannt.

Wie funktioniert AlexNet?

AlexNet nutzt Convolutional Layers, Fully Connected Layers und Aktivierungsfunktionen, um Bilder Schritt für Schritt zu analysieren. Es erkennt Muster wie Kanten, Formen und komplexe Objekte.

Schritt-für-Schritt-Erklärung:

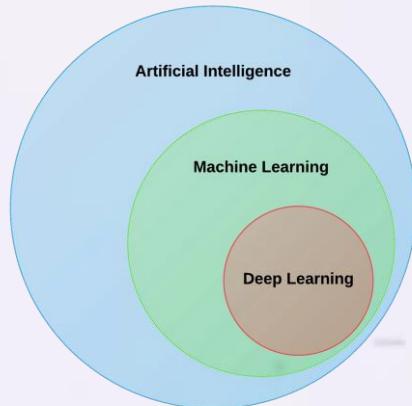


Fully Connected Layers (FC6—FC8): Entscheidung treffen

4096 Neuronen: Kombinieren alle Merkmale (Ohren, Schnauze, etc.).

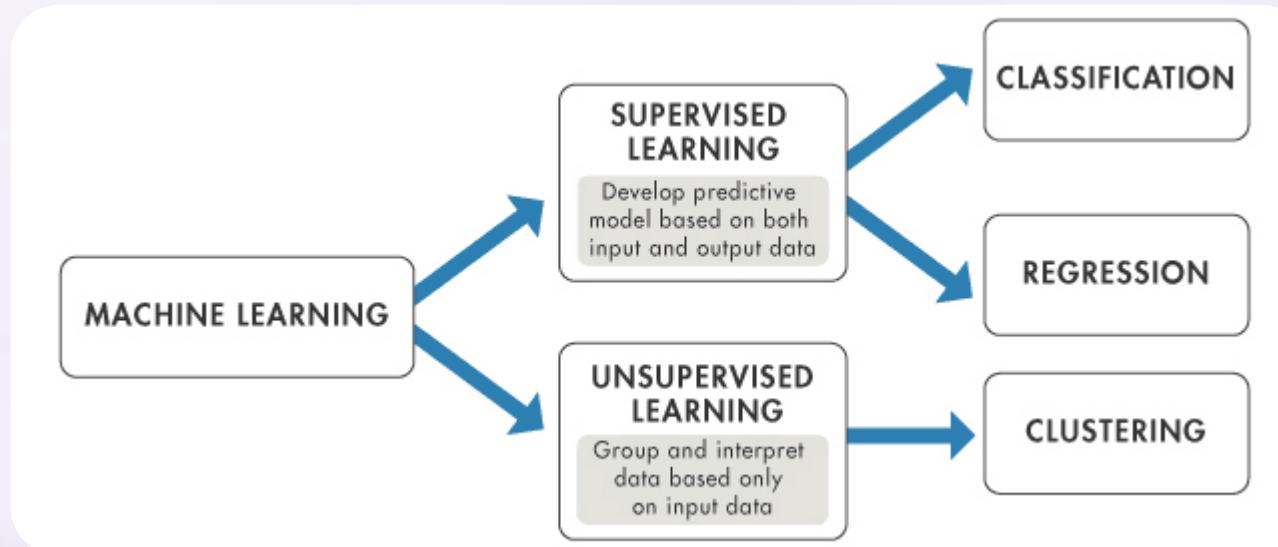
Output: "Das ist ein Hund" mit 98 % Wahrscheinlichkeit.

Machine Learning



Machine Learning (ML) ist ein Teilgebiet der Künstlichen Intelligenz (KI), bei dem Computer aus Daten lernen, um Muster zu erkennen und Vorhersagen zu treffen, ohne explizit programmiert zu werden.

Es gibt verschiedene Methoden “Erfahrungen“ zu lernen und ihre Leistung im Laufe der Zeit zu verbessern.



Überwachtes Lernen (Supervised)

Modelle werden mit gekennzeichneten Daten trainiert, um spezifische Ausgaben vorherzusagen.

Unüberwachtes Lernen (Unsupervised)

Modelle identifizieren Muster in unmarkierten Daten.

Verstärkendes Lernen (Reinforced)

Modelle lernen durch Belohnungen und Strafen in einer dynamischen Umgebung.

Machine Learning Workflow

Projektaufbau (Setup)

Geschäftsziele verstehen

Problem analysieren und klären: „Was wollen wir lösen?“

Lösung auswählen

Passende Machine Learning-Technik identifizieren.

Datenvorbereitung

Quellen finden und relevante Daten erfassen.

Datensätze säubern und analysierbar machen.

Daten anreichern und aufbereiten.

Trainings- und **Testdatensätze** erstellen.

Modellierung

Hyperparameter optimieren

Modelle trainieren

Vorhersagen machen

Leistung bewerten

Deployment

Ergebnisse in Dashboards oder Tools einbetten.

Laufende Modell-Optimierung sicherstellen.

Iterationen und Feedback einarbeiten.

Deep Learning



Spracherkennung

Systeme wie Siri oder Alexa verstehen und verarbeiten gesprochene Sprache.



Bildverarbeitung

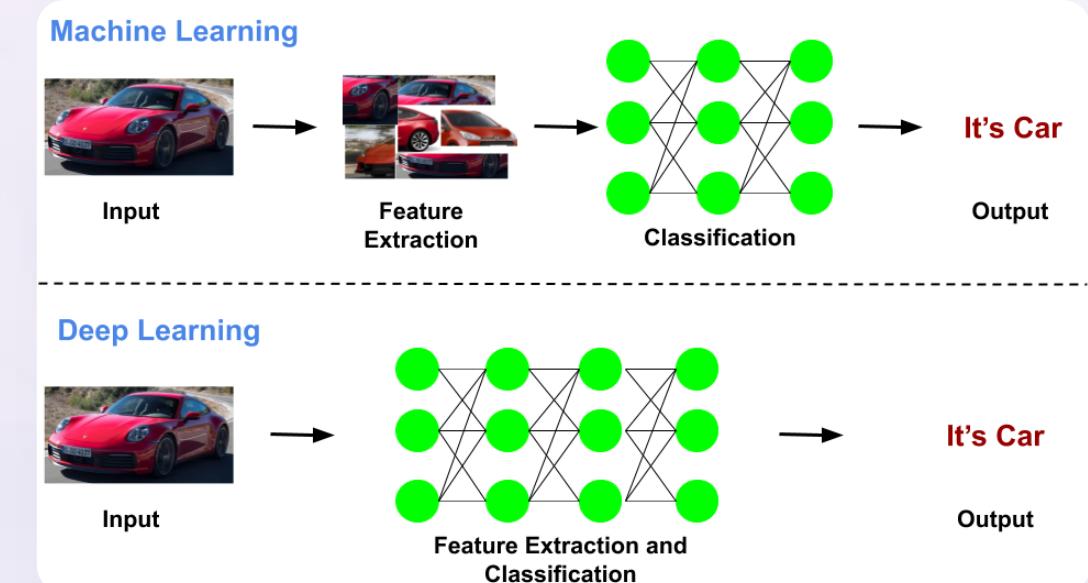
Automatische Erkennung von Objekten in Bildern, z.B. in selbstfahrenden Autos.



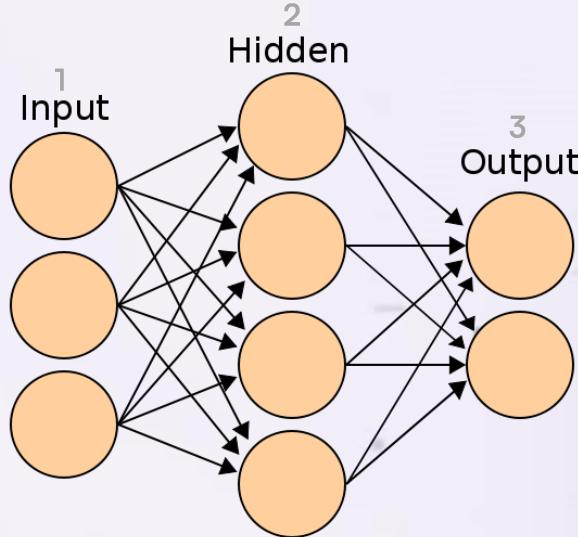
Übersetzung

Automatische Übersetzung von Texten zwischen verschiedenen Sprachen.

Deep Learning (DL) ist eine spezialisierte Unterkategorie des Machine Learning, die auf künstlichen neuronalen Netzwerken mit mehreren Schichten basiert. Diese tiefen Netzwerke ermöglichen es Computern, komplexe Muster in Daten zu erkennen und Aufgaben wie Bilderkennung und Sprachverarbeitung mit höherer Genauigkeit zu bewältigen.

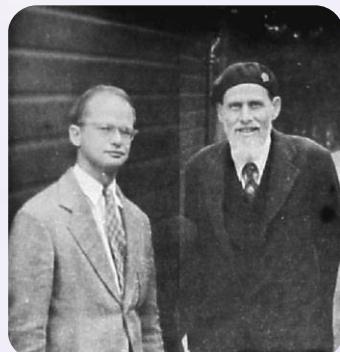


Neural Networks



Neuronale Netzwerke sind ein zentraler Bestandteil der modernen Künstlichen Intelligenz (KI). Inspiriert von der Funktionsweise des menschlichen Gehirns, bestehen sie aus miteinander verbundenen "Neuronen" (Knotenpunkten), die in Schichten organisiert sind.

Jedes "Neuron" überträgt Informationen an die nächste Schicht, bis das Netzwerk eine Vorhersage oder Entscheidung trifft. Der Lernprozess erfolgt durch Fehlerkorrektur — ein Konzept, das als Rückpropagation bekannt ist.



1943:

Warren McCulloch und Walter Pitts entwickelten das erste Modell zur mathematischen Beschreibung von Neuronen.

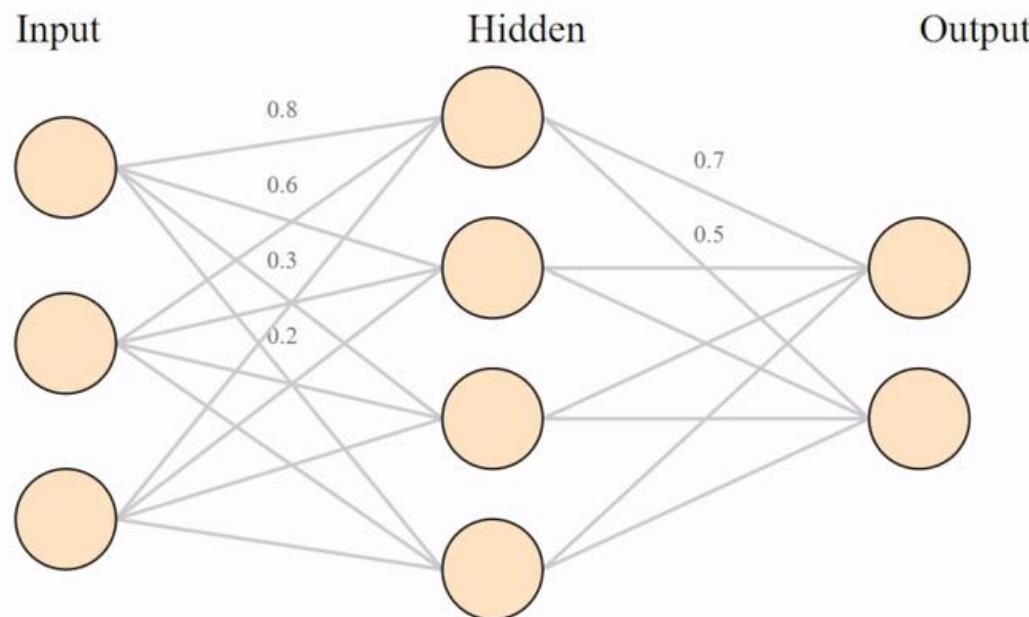


1986:

Geoffrey Hinton und David Rumelhart popularisierten die Technik der Rückpropagation (Backpropagation), wodurch neuronale Netze effektiv trainiert werden konnten.

Neural Networks

Die Funktionsweise neuronaler Netze basiert auf der Verarbeitung von Daten durch Schichten. Informationen fließen von den Eingabeschichten über versteckte Schichten zu den Ausgabeschichten. Dabei lernen die Netzwerke durch Anpassung der Gewichte zwischen den Knoten.



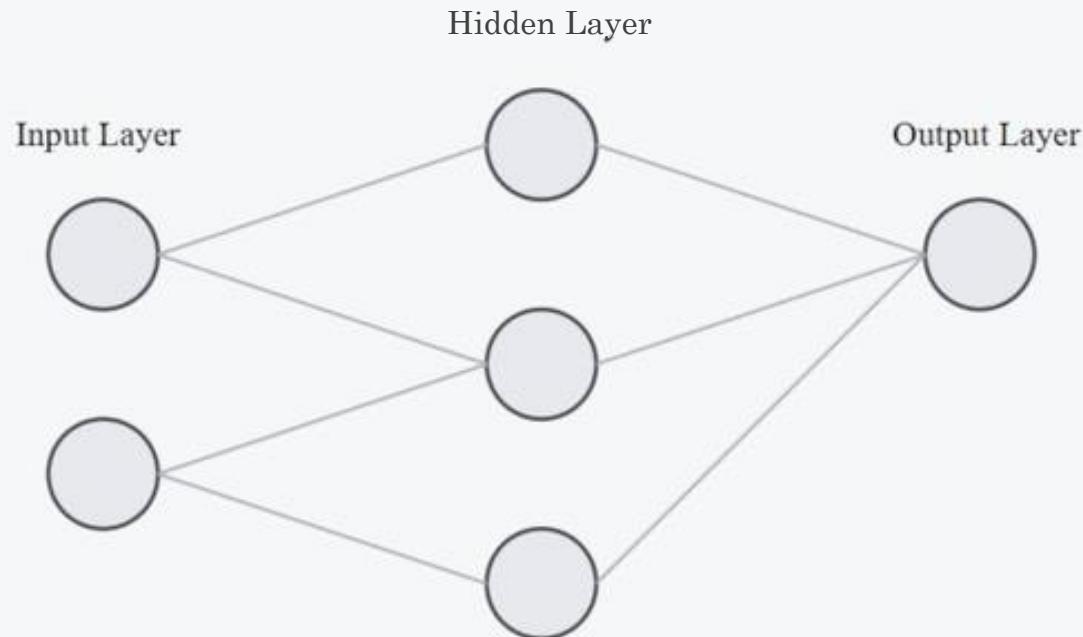
Struktur eines neuronalen Netzes:

- **Input Layer:** Nimmt Eingangsdaten auf.
- **Hidden Layers:** Verarbeitet die Daten, erkennt Muster und abstrahiert Informationen.
- **Output Layer:** Gibt die finale Entscheidung oder Vorhersage.

Gewichte (Weights):

- Jede Verbindung zwischen den Neuronen hat ein Gewicht, das die Stärke der Verbindung angibt.
- Beim Durchlaufen des Netzwerks wird ein Eingabewert mit dem Gewicht multipliziert, bevor er an die nächste Schicht weitergegeben wird.
- Die Anpassung dieser Gewichte während des Trainings verbessert die Genauigkeit des Netzwerks.

Neural Networks



1. Eingabe der Trainingsdaten (Input: [0.5, 0.8])

Datenfluss durch das Netzwerk:

- Eingaben in die Input-Layer werden verarbeitet und durch die Hidden Layers weitergeleitet.
- Jede Verbindung hat ein Gewicht, das den Beitrag eines Knotens zur nächsten Schicht beeinflusst.
- Aktivierungsfunktionen (z. B. ReLU, Sigmoid) bestimmen, ob ein Neuron "feuert".

Backpropagation (Rückpropagation):

- Ein Lernalgorithmus, der die Gewichte anpasst, basierend auf dem Fehler zwischen vorhergesagten und tatsächlichen Ausgaben.
- Ziel: Die Fehlerfunktion (Loss Function) zu minimieren, indem die Gewichte iterativ optimiert werden.

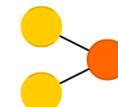
A mostly complete chart of

Neural Networks

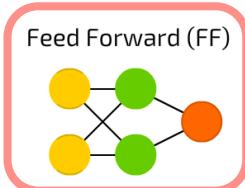
©2016 Fjodor van Veen - asimovinstitute.org

- Backfed Input Cell
- Input Cell
- △ Noisy Input Cell
- Hidden Cell
- Probabilistic Hidden Cell
- △ Spiking Hidden Cell
- Output Cell
- Match Input Output Cell
- Recurrent Cell
- Memory Cell
- △ Different Memory Cell
- Kernel
- Convolution or Pool

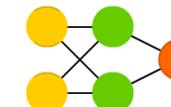
Perceptron (P)



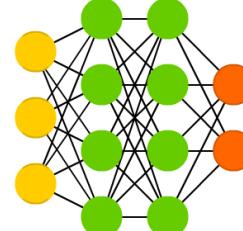
Feed Forward (FF)



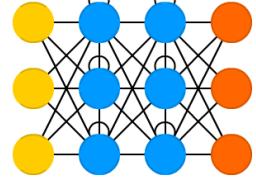
Radial Basis Network (RBF)



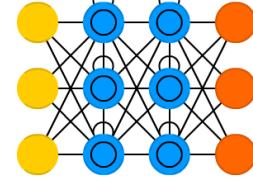
Deep Feed Forward (DFF)



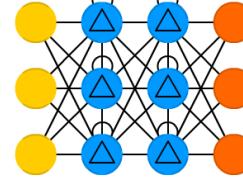
Recurrent Neural Network (RNN)



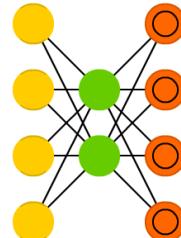
Long / Short Term Memory (LSTM)



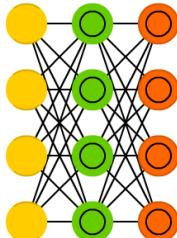
Gated Recurrent Unit (GRU)



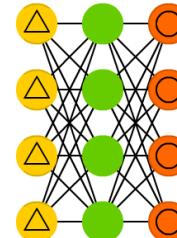
Auto Encoder (AE)



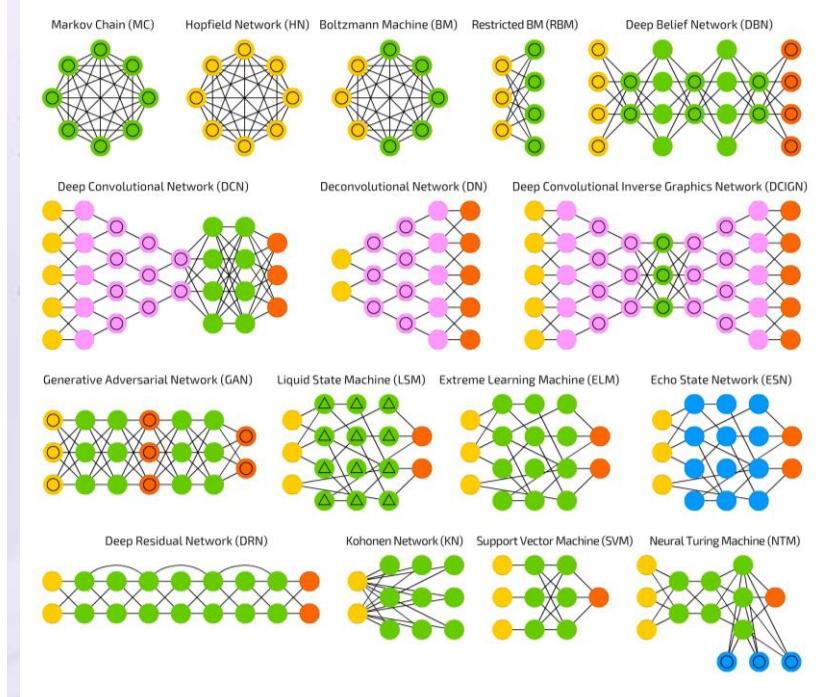
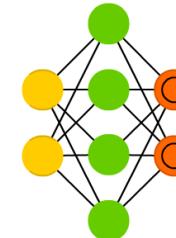
Variational AE (VAE)



Denoising AE (DAE)



Sparse AE (SAE)



Transformer Modelle

Transformer-Modelle revolutionieren das maschinelle Lernen durch ihre Fähigkeit, den Kontext eines Textes effektiv zu erfassen. Vorgestellt im bahnbrechenden Paper "Attention is All You Need" (2017), bilden Transformer die Grundlage moderner KI-Anwendungen wie GPT und BERT. Sie können Texte schreiben, Fragen beantworten, Sprachen übersetzen und sogar Prüfungen bestehen, die für Menschen herausfordernd sind.

 **Attention Is All You Need**

Ashish Vaswani^{*}
Google Brain
avaswani@google.com

Noam Shazeer^{*}
Google Brain
noam@google.com

Niki Parmar^{*}
Google Research
nikip@google.com

Jakob Uszkoreit^{*}
Google Research
uaz@google.com

Llion Jones^{*}
Google Research
llion@google.com

Aidan N. Gomez[†]
University of Toronto
aidan@cs.toronto.edu

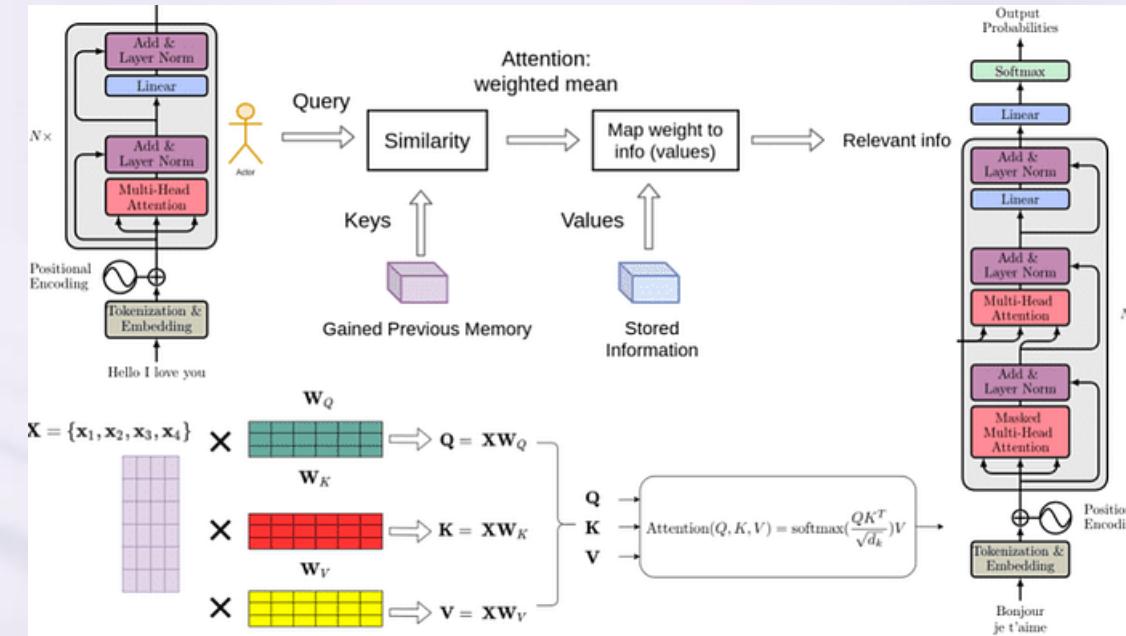
Lukasz Kaiser^{*}
Google Brain
lukasz.kaiser@google.com

Illia Polosukhin[‡]
illia.polosukhin@gmail.com

Abstract

The dominant sequence transduction models are based on complex recurrent or convolutional neural networks that include an encoder and a decoder. The best performing models also connect the encoder and decoder through an attention mechanism. We propose a new simple network architecture, the Transformer, based solely on attention mechanisms, dispensing with recurrence and convolutions entirely. Experiments on two machine translation tasks show these models to be superior in quality while being more parallelizable and requiring significantly less time to train. Our model achieves 28.4 BLEU on the WMT 2014 English-to-German translation task, improving over the existing best results, including ensembles, by over 2 BLEU. On the WMT 2014 English-to-French translation task, our model establishes a new single-model state-of-the-art BLEU score of 41.8 after training for 3.5 days on eight GPUs, a small fraction of the training costs of the best models from the literature. We show that the Transformer generalizes well to other tasks by applying it successfully to English constituency parsing both with large and limited training data.

<https://arxiv.org/abs/1706.03762>



Hauptidee
Transformer halten den Kontext eines Textes über lange Passagen hinweg fest.

Im Gegensatz zu traditionellen Modellen bauen sie Text Wort für Wort, wobei jedes Wort den bisherigen Kontext berücksichtigt.

Transformer Architektur

Die Architektur eines Transformers besteht aus mehreren Schichten, die spezifische Aufgaben erfüllen. Obwohl sie auf den ersten Blick komplex erscheinen, lassen sie sich in fünf Hauptkomponenten unterteilen, die zusammenarbeiten, um Texte zu verstehen und zu generieren.

Tokenisierung:

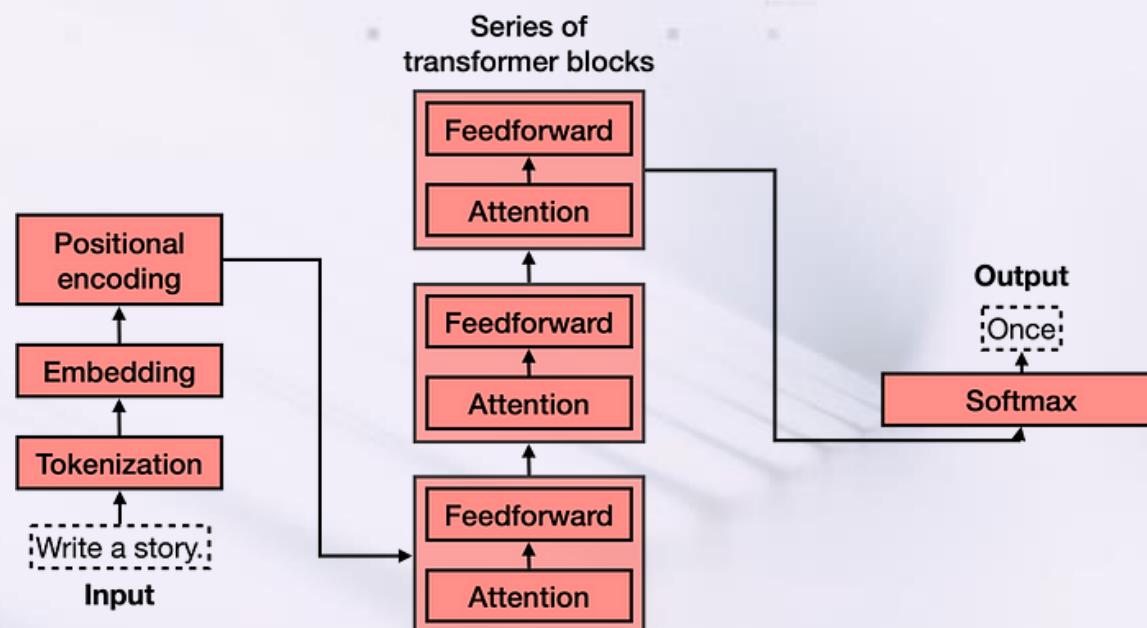
Zerlegt Text in kleinere Einheiten (Tokens).

Einbettung

Übersetzt Tokens in numerische Vektoren.

Positions kodierung

Fügt den Tokens Informationen über ihre Position im Text hinzu. Ermöglicht es dem Modell, die Reihenfolge der Wörter zu verstehen.



Transformer-Block:

Self-Attention: Erkennt Beziehungen zwischen Wörtern, unabhängig von ihrer Position im Text.

Feedforward-Schicht: Verarbeitet und abstrahiert die Informationen aus der Self-Attention

Softmax:

Liefert Wahrscheinlichkeiten für mögliche nächste Wörter.

WAS SIND TOKENS?

GPT 4

I visit the maritime museum with my friends.

Tokens	Characters
9	44

Ich besuche mit meinen Freunden das Schifffahrtsmuseum.

Tokens	Characters
16	55

GPT 3

Ich besuche mit meinen Freunden das Schifffahrtsmuseum.

Tokens	Characters
22	55

Tokens sind die einzelnen Bausteine eines Textes.

Die Art und Weise, wie ein Wort in Tokens aufgeteilt wird, hängt von den Regeln der Tokenisierung ab.

1 Token $\sim=$ 4 Zeichen auf Englisch

1 Token $\sim=$ $\frac{3}{4}$ Wörter

100 Tokens $\sim=$ 75 Wörter

1-2 Sätze $\sim=$ 30 Tokens

1 Absatz $\sim=$ 100 Tokens

1.500 Wörter $\sim=$ 2.048 Tokens

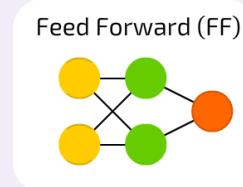
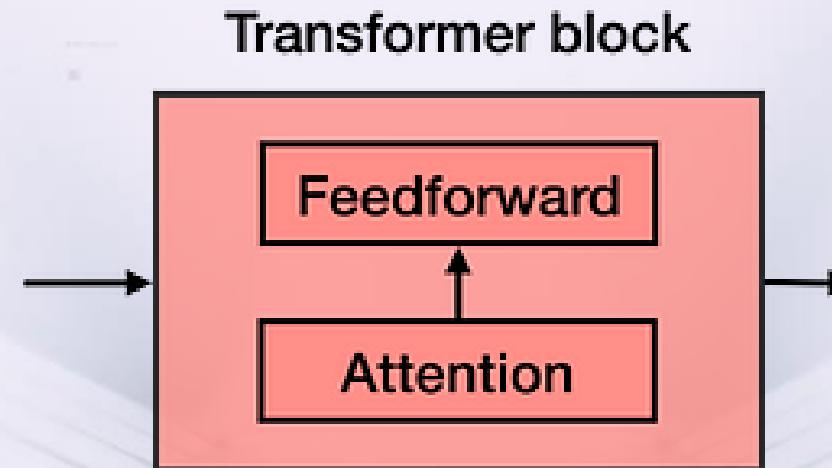
Transformer-Block

Der Transformer-Block ist das Herzstück moderner Transformer-Modelle und kombiniert die **Attention** und **Feedforward-Schichten**. Gemeinsam ermöglichen sie es, komplexe Beziehungen zwischen Wörtern in einem Text zu erfassen und präzise Vorhersagen zu treffen.

Was ist Self-Attention?

Ein Mechanismus, der jedem Wort Gewicht verleiht, basierend darauf, wie wichtig es im Kontext anderer Wörter ist.

Beispiel: Im Satz "Die Katze jagt die Maus" erkennt das Modell, dass "Katze" und "jagt" eng miteinander verbunden sind.

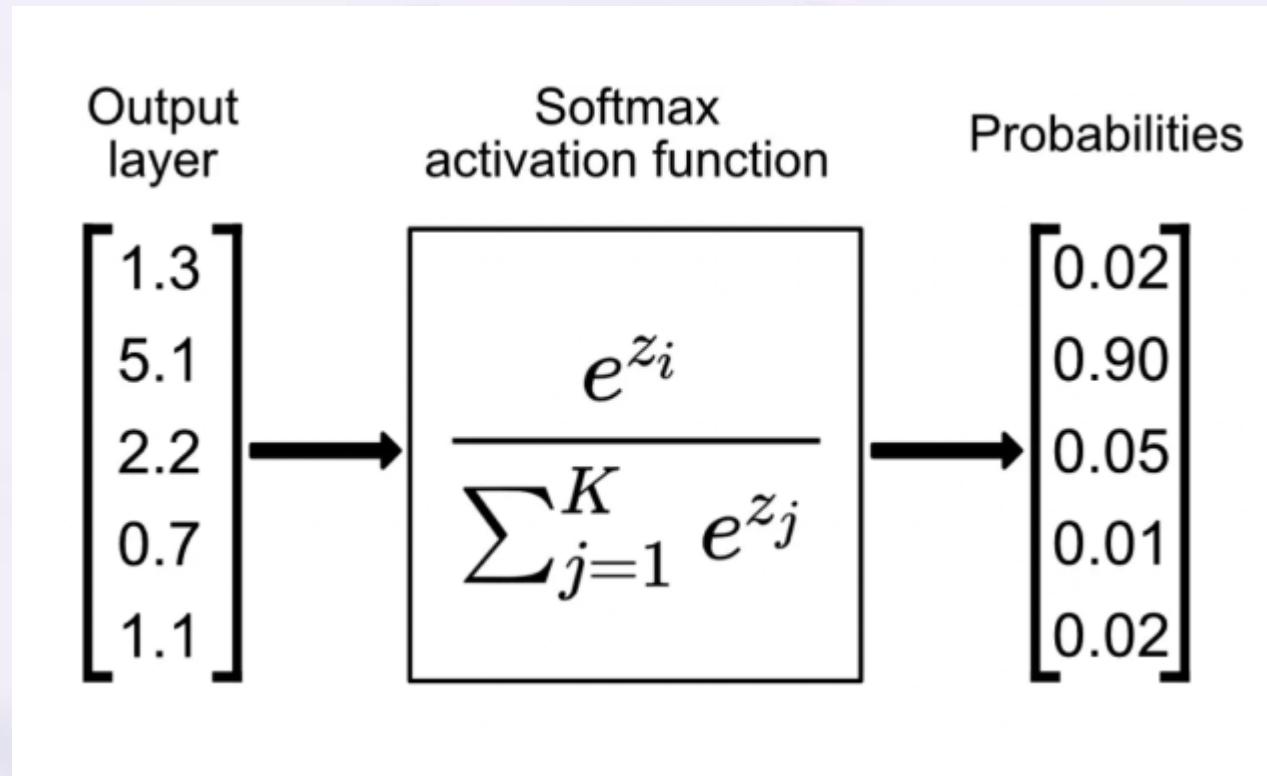


Feedforward-Komponente

Verarbeitung der Ergebnisse der Attention-Schicht.
Besteht aus mehreren vollständig verbundenen neuronalen Schichten.

Softmax

Die Softmax-Schicht ist das letzte Element eines Transformer-Modells und übersetzt die Ergebnisse des Netzwerks in Wahrscheinlichkeiten. Diese Wahrscheinlichkeiten geben an, wie wahrscheinlich jedes Wort als nächstes in einer Sequenz erscheint. Sie ermöglicht es dem Modell, das beste nächste Wort basierend auf Kontext und Mustererkennung zu wählen.



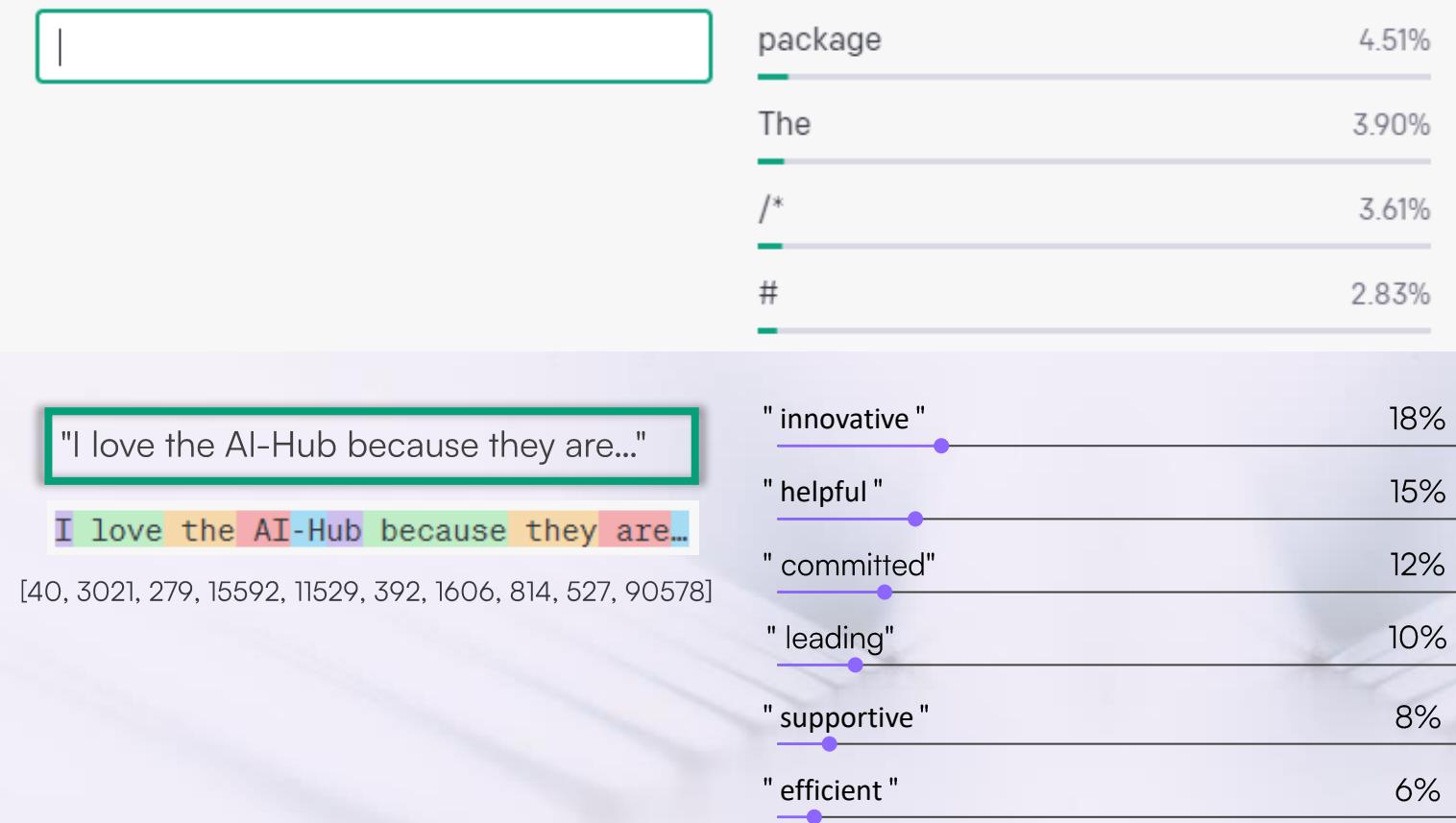
Aufgabe der Softmax-Schicht:

Wandelt die Rohwerte (Scores) der vorherigen Schichten in Wahrscheinlichkeiten um, die zusammen 1 ergeben.

Beispiel: Für die Eingabe "Schreibe eine Geschichte" liefert der Transformer folgende Wahrscheinlichkeiten:
"Es" = 0,5
"Da" = 0,3
"Hier" = 0,2

AUTO COMPLETE BEISPIEL

No Context



LLMs zerteilen also die Eingabe und Ausgabe in Tokens und berechnen die Wahrscheinlichkeit hinter der die jeweiligen Token stehen.

Der Temperaturparameter beeinflusst, wie stark Wahrscheinlichkeiten gewichtet werden (höhere Temperatur = mehr Variation).

Beim Top-k-Sampling wählt das Modell eine vorgegebene Nummer von den wahrscheinlichsten nächsten Tokens.

Daten sind das neue Gold

Laut Statista beträgt das weltweit generierte **Datenvolumen 2024 etwa 153 Zettabyte** und wird bis 2025 auf über 180 Zettabyte anwachsen!

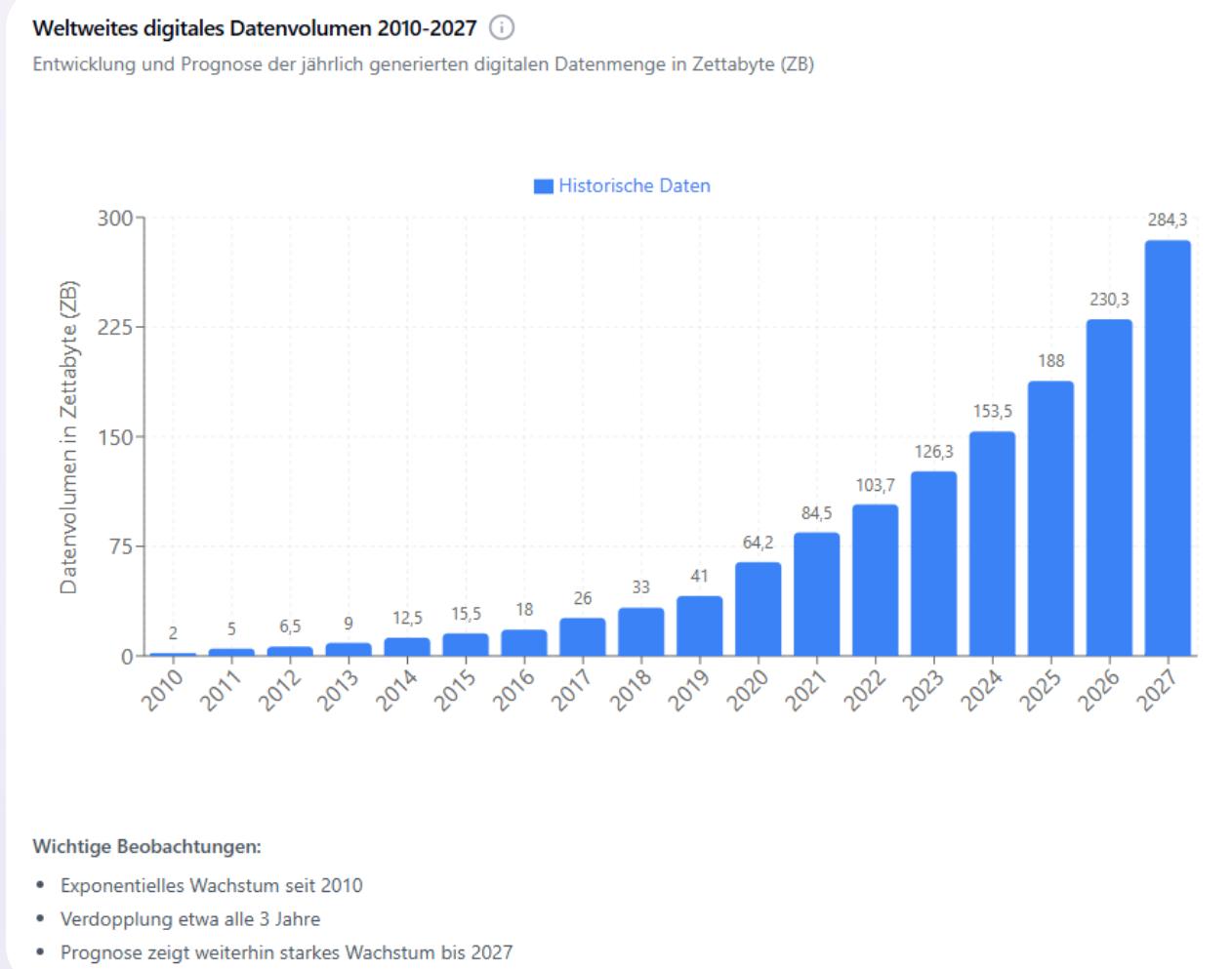
Zettabyte = 1 Billion Gigabyte

Genug Daten, um fast jedes Detail unseres Lebens zu dokumentieren.

Grundlage für Fortschritte in **Machine Learning (ML)** und **Deep Learning (DL)**.

Nutzung:

Gesundheitswesen, Finanzwesen, autonome Systeme und vieles mehr.



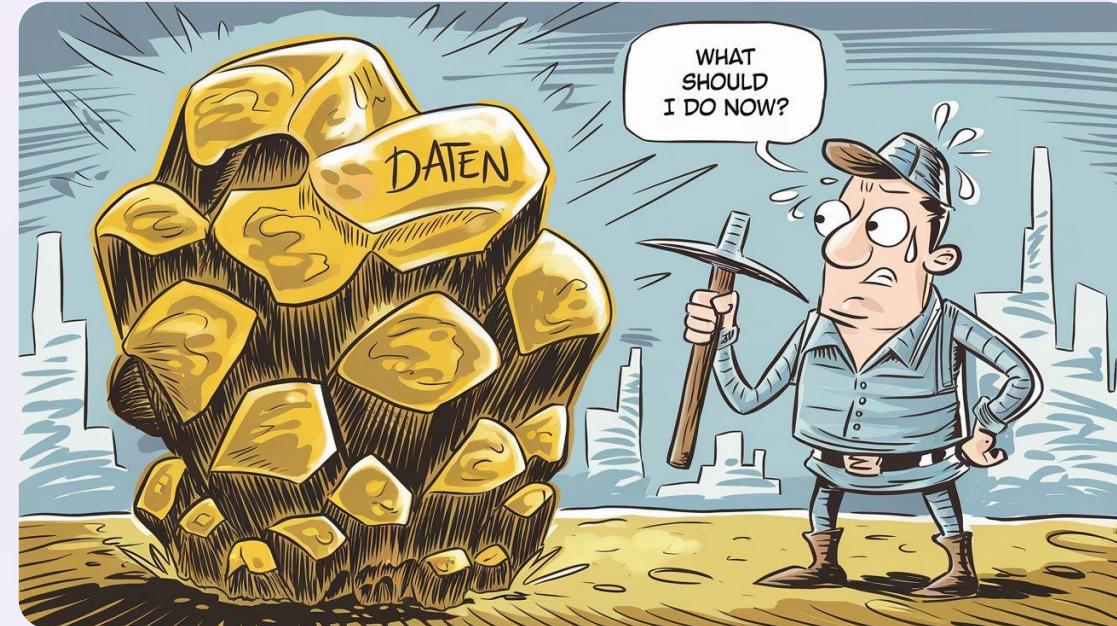
Nicht alles was glänzt..

2024

153

zetta-bytes*

1 Gigabyte (GB)	= 1.000 Megabyte
1 Terabyte (TB)	= 1.000 Gigabyte
1 Petabyte (PB)	= 1.000 Terabyte
1 Exabyte (EB)	= 1.000 Petabyte
1 Zettabyte (ZB)	= 1.000 Exabyte



Daten sind das neue Gold" — doch wie rohes Gold müssen sie erst verarbeitet werden, bevor sie wertvoll werden.

Nur durch Reinigung, Strukturierung und gezielte Aufbereitung können sie KI-Modelle antreiben und echte Erkenntnisse liefern.



Smartphone

Ein Smartphone hat 512 GB Speicher

→ Für 1 Zettabyte bräuchte man über 2 Milliarden Smartphones



Festplatten

Eine 20 TB Festplatte ist heute Standard

→ Für 1 Zettabyte bräuchte man 50 Millionen solcher Festplatten



Netflix Server

Es etwa 100 Petabyte gespeichert

→ 1 Zettabyte entspricht 10.000 Netflix-Speicherzentren



Digitalisierte Bücher

Ein eBook benötigt ungefähr 2 MB

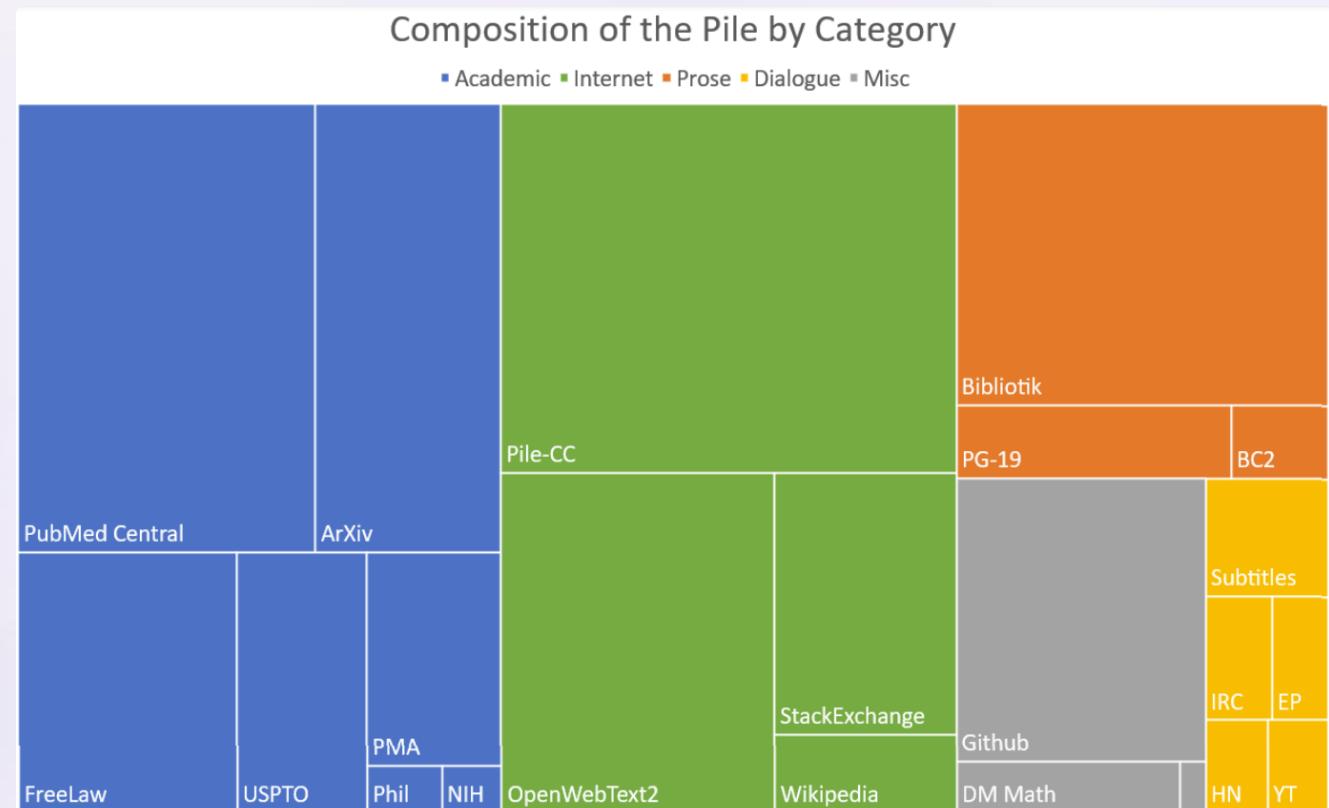
→ 1 Zettabyte könnte 500 Billionen eBooks speichern

Der Haufen

The Pile ist ein **825 GB** großer, vielfältiger Datensatz, der von EleutherAI entwickelt wurde, um KI-Modelle mit einer breiten Palette an Texten zu füttern.

Dieser "Haufen" wurde für das Training von GPT genutzt, und diente auch als Grundlage für Modelle wie Microsoft, Meta AI's.

Durch die Vielfalt der enthaltenen Texte ermöglicht "The Pile" es KI-Modellen, ein breites Spektrum an Sprachmustern und Kontexten zu erlernen, was zu beeindruckenden Fähigkeiten in der Textgenerierung geführt hat.



Von Rohdaten zu Trainingsdaten

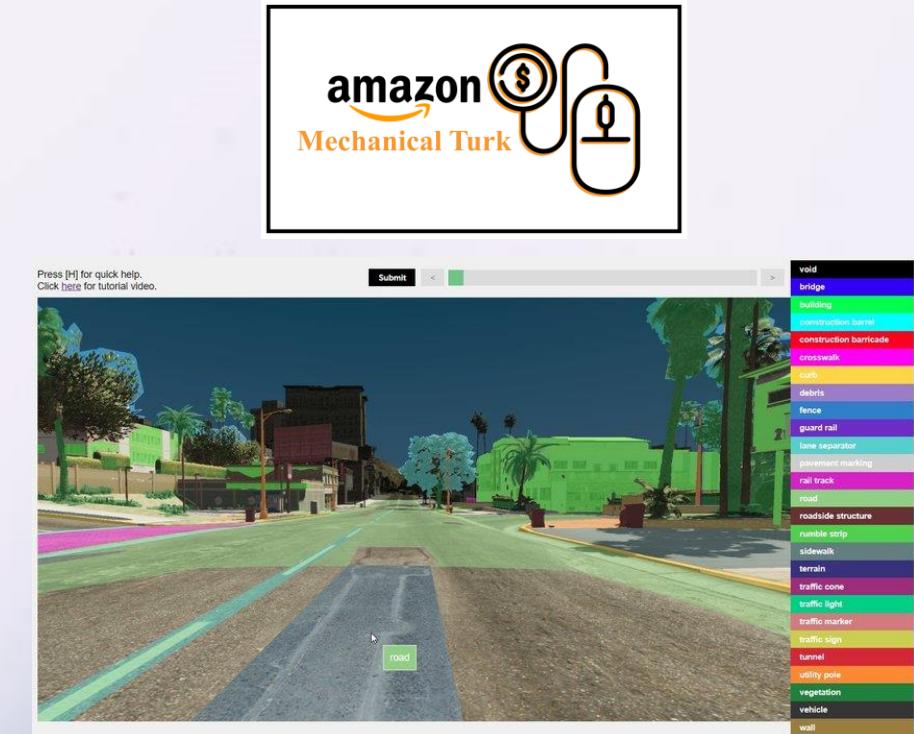
Rohdaten aus dem Internet enthalten oft Fehler, Duplikate und Inkonsistenzen. Ohne Bereinigung und Labeling können sie nicht effektiv für KI-Modelle genutzt werden.



Datenbereinigung:

Entfernen von Duplikaten, Fehlern und irrelevanten Einträgen.
Ziel: Konsistenz und Verlässlichkeit der Daten sicherstellen.

„Shit in, shit out“



Datenlabeling:

Manuelle oder automatische Annotation von Daten, z. B. Markierung eines Bildes als "Hund".

Wird häufig durch Plattformen wie Amazon Mechanical Turk oder Clickworker durchgeführt.

Q&A

The End