# *Künstliche Intelligenz - Eine Einführung*

*Lukas Hutter*

*08. März 2024*

Die Folien zu diesem Handout sind unter https://el-uhu.github.io/ai-intro-2023 zu finden.

## *Grundlagen*

### *Geschichte*

- 1950 Alan Turing: Turing Test [1]
- 1951 Marvin Minsky: SNARC, analoges Neuronales Netzwerk
- 1956 Dartmouth Workshop [2]
  - offizielle Geburtsstunde
  - Minsky, McCarthy, Shannon, Rochester
  - Ziel: Ausloten der Möglichkeit intelligente Maschinen zu entwickeln ( aus Erfahrung lernen, Probleme lösen)
  - Begriff *Artificial Intelligence* geschaffen
- 1960-1980 Expertensysteme und Symbolische KI
- 1980-1990 Neuronale Netze
- 1990-2000 SVMs und Kernelmethoden
- 2010 - heute Deep Learning
- 2022 - heute Generative KI

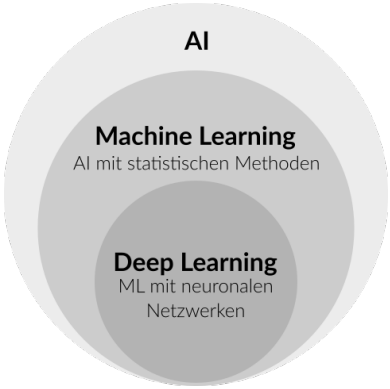
[1] A. M. Turing. Computing machinery and intelligence. *Mind*, LIX(236):433–460, Oct. 1950

[2] J. McCarthy, M. L. Minsky, N. Rochester, and C. E. Shannon. A Proposal for the Dartmouth Summer Research Project on Artificial Intelligence, August 31, 1955. *AI Magazine*, 27(4):12–12, Dec. 2006. Number: 4

Abbildung 1: Marvin Minsky, Claude Shannon, Ray Solomonoff and other scientists at the Dartmouth Summer Research Project on Artificial Intelligence (Photo: Margaret Minsky).

### *Terminologie*

Modell - Begriff, der sich im Kontext der künstlichen Intelligenz auf die trainierte Version eines bestimmten Lernalgorithmus bezieht.




Artificial Intelligence / Künstliche Intelligenz - Teilgebiet der Computerwissenschaften, das sich mit der Entwicklung intelligenter Maschinen befasst, die in der Lage sind menschenähnliches Verhalten zu simulieren und es Computern ermöglichen Aufgaben zu erledigen, deren Lösung typischerweise nach menschlicher Intelligenz verlangt

- **Narrow AI / Weak AI** - spezifische Probleme, zB Voice Assistants, Bilderkennung, Empfehlungssysteme
- **General AI / Strong AI** - Verstehen, Lernen, Wissen kontextunabhängig Einsetzen

MACHINE LEARNING / MASCHINELLES LERNEN Teilgebiet der künstlichen Intelligenz, Algorithmen, die es Computern erlauben automatisch auf Basis von Daten zu lernen.

- **Supervised** - Das Erkennen von Mustern wird auf Basis vorklassifizierter Daten erlernt $\longrightarrow$ Klassifikation neuer Daten
- **Unsupervised** - Muster werden ohne spezifische Anleitung selbstständig in den Daten erkannt $\longrightarrow$ Erkennen verborgener Beziehungen in den Daten
- **Reinforcement Learning** - Ein vor-trainiertes ML-Modell erhält während des Einsatzes Feedback in Form von *Belohnungen* und *Strafen* und lernt über die Zeit die *Belohnungen* zu maximieren (vgl. Pavlov'sche Konditionierung)

**ML-Systeme:**

- **Support Vector Machines (SVMs)**
  - Datenpunkte werden durch eine Vielzahl an Eigenschaften beschrieben $\longrightarrow$ Punkte in einem hochdimensionalen Raum (*Feature Space*)
  - Ziel: Klassifizierung durch Teilung des Feature Spaces mittels eines geraden Schnitts (lineare Separierbarkeit)
  - Trick: Verformen des Feature Spaces mit Hilfe mathematischer Transformationen (Kernel Functions)
  - Lernen zielt darauf ab die optimale Transformation zu finden, die eine maximale Unterscheidbarkeit der Klassen gewährleistet.
- **Entscheidungsbäume & Random Forests**
  - hierarchischer Ansatz, der Klassifizierung als mehrstufigen Prozess von Ja/Nein-Fragestellungen umsetzt
  - **Random Forests** - kombiniert die Klassifikation eines *Ensembles* an unterschiedlichen, zufällig generierten Entscheidungsbäumen und verbessert so die Robustheit und Generalisierbarkeit von Klassifizierungen.

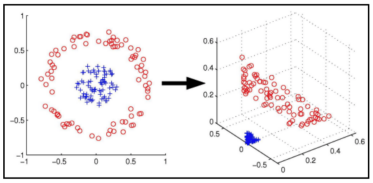


Abbildung 2: Transformation des Feature-Spaces bei SVMs

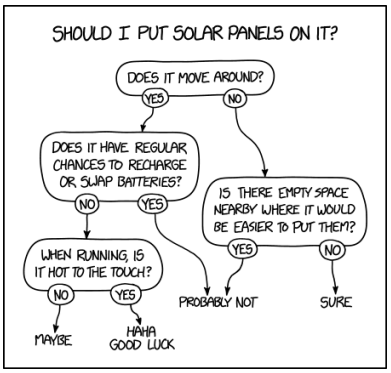



Abbildung 3: Ein einfacher Entscheidungsbaum; Quelle: xkcd

DEEP LEARNING Teilgebiet des maschinellen Lernens, in dem Lernalgorithmen zum Einsatz kommen, sich an neuronalen Netzwerken aus der Natur orientieren

- Knoten (~ Neuronen) in einem Netzwerk sind in Schichten miteinander verbunden und können einander *aktivieren* oder *hemmen*
- Die Stärke der Verbindungen wird im Rahmen des Lernprozesses angepasst
- Schichten: Input $\longrightarrow$ *Deep Layers* $\longrightarrow$ Output
- *Deep Layers* ermöglichen das Erkennen von Mustern
- große Datenmengen, komplexe Aufgaben

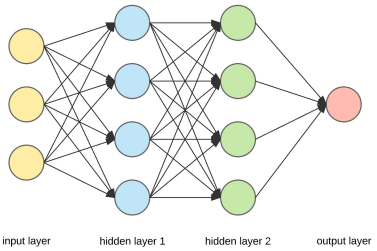


Abbildung 4: Allgemeiner Aufbau eines künstlichen neuronalen Netzwerks (ANN)

BACKPROPAGATION Schlüsselalgorithmus zum iterativen Trainieren von neuronalen Netzwerken.

- Vorwärtsschritt (Vorwärtspropagierung) - Aktivierung der einzelnen Knoten wird nach Eingabe eines Inputs ermittelt
- Vergleich des Outputs mit dem *wahren Wert* $\longrightarrow$ Messen des Fehlers
- Rückwärtsschritt (Zurückpropagierung) - Anpassen der Verbindungsstärke um Fehler zu minimieren

CONVOLUTIONAL NEURAL NETWORKS spezielle neuronale Netze; entwickelt für Daten im Rasterformat $\longrightarrow$ Einsatz im Bereich Computer Vision / Bildanalyse

- Räumlich nahe beieinander liegende Inputs werden mathematisch zusammengefasst (*Convolutional Layer*) $\longrightarrow$ Knoten eines Layers sind nicht mit allen Knoten eines vorigen Layers verknüpft (*Local receptive fields*) $\longrightarrow$ Muster können erkannt werden, egal wo im Bild sie sich befinden $\longrightarrow$ einfachere Trainierbarkeit
- *Pooling* - Verwerfen überflüssiger Informationen (wie beispielweise genaue Position eines Features) $\longrightarrow$ Effizienz, Robustheit
- Aktivierungsmuster zwischen den Layers sind bei verschiedenen rezeptiven Feldern gleich gestaltet (*weight sharing*) $\longrightarrow$ Generalisierbarkeit

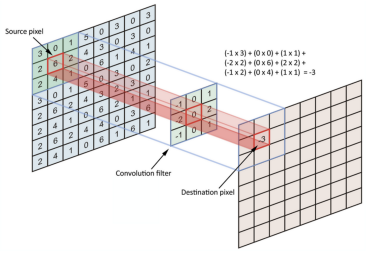


Abbildung 5: Convolution

RECURRENT NEURAL NETWORKS Spezialarchitektur neuronaler Netze, die sich besonders gut für die Analyse von Mustern in sequentiellen Daten eignet $\longrightarrow$ Sprachanalyse. Verfügt über ein Gedächtnis, welches durch Rückkoppelungsmechanismen zwischen den Layern erzeugt wird.

*Generative AI*

KLASSISCHE ML SYSTEME ergeben Outputs die einen Input *klassifizieren/einordnen/labeln*.

MODERNE AI SYSTEME kombinieren verschiedene Arten klassischer ML Systeme in einem komplexeren Aufbau, um auf Basis eines Inputs *neue Inhalte zu generieren* (Sprache, Audio, Bilder), indem ein vor-trainiertes Modell eine wahrscheinliche Antwort auf einen Input-*Prompt* erzeugt. (GPT, LaMDA, PaLM, DALL-E, Midjourney)

GENERATIVE ADVERSARIAL NETWORKS[3] (ab 2014) bauen auf *Reinforcement Learning auf* und stellen zwei Netzwerke in einen Wettstreist (Nullsummenspiel)

- *Generator* - erzeugt Daten (Bilder, Text)
- *Diskriminator* - lernt *echte* Daten zu erkennen und klassifiziert Outputs des Generators

Klassifikation wird mit wahrem Wert verglichen und verwendet um die Verbindungsstärke der Knoten beider Netzwerke zu aktualisieren.

[3] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative Adversarial Nets

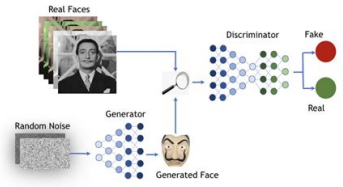



Abbildung 6: Aufbau eines GANs

LARGE LANGUAGE MODELS Große, generative AI Systeme (GPT-n, BERT, PaLM, DALL-E, Midjourney) verwenden Sprache als Input (*Prompts*); Sprache wichtige Schnittstelle für Interaktion mit ML Systemen

WIE GEHT AI MIT SPRACHE UM? In Sprachmodellen werden Wörter (oder allg. *Tokens*) zunächst in Vektoren übersetzt (*Embedding*), um syntaktische und semantische Bedeutung zu kodieren (Computer arbeiten mit mathematischen Objekten)

Recurrent Neuronal Networks (RNNs) sind grundsätzlich gut geeignet, aber langsam, da Worte nur nacheinander verarbeitet werden können (nicht parallelisierbar).

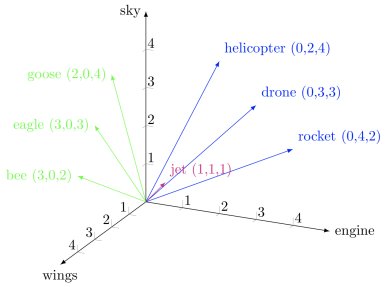



Abbildung 7: Veranschaulichung eines Embeddings

TRANSFORMER ARCHITEKTUR [4] transformiert Token-Sequenz (Satz) in andere Token-Sequenz, z.B. Übersetzung von Deutsch auf Englisch; findet Anwendung in GPT, BARD

Architektur, die parallele Verarbeitung ermöglicht, indem die Position von Worten innerhalb eines Satzes als Teil des Embedding-Vektors gespeichert wird (*Positional Encoding*)

[4] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention Is All You Need, Dec. 2017. arXiv:1706.03762 [cs]

Bestehen allg. aus *Encoder*- und *Decoder*-Blöcken (neuronale Netze), die Sätze als Vektoren abbilden und Satzbausteine zueinander in Bezug setzen.

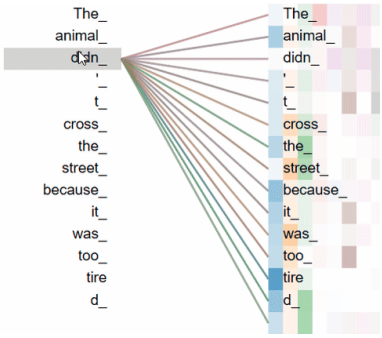


Abbildung 8: Attention System

Encoder Input-Embeddings (z.B mathematische Abbildung deutscher Sätze) werden in einem neuronalen Netz namens *Encoder* verarbeitet, der erlernt die Beziehungs-Stärke der Wörter zueinander zu messen (*Attention*).

Decoder Ähnlich gestaltet wie Encoder: erlernt Output-Embeddings(z.B. mathematische Abbildung englischer Sätze) und verwendet Attention-System um Beziehungsstärke der Wörter innerhalb dieser zu messen. Erlernt mittels eines weiteren Attention-Systems Input- und Output-Embeddings miteinander in Bezug zu setzen und für einen bestimmten Input den besten (wahrscheinlichsten) Output zu ermitteln.

Generative Pre-trained Transformer (GPT)[5]

[5] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever. Improving Language Understanding by Generative Pre-Training

- basiert auf modifizierter Transformer-Architektur
- nutzt riesige Trainingsdatenmengen
- sogenannte *Foundation Models*: GPT-1, GPT-2, GPT-3,...
- Mehrstufiger Lernprozess:
  1. auf Basis selbstständig generierter Lückentexte
  2. Supervised Learning
- GPT-4[6]:
  - etwa 1 Billion Parameter (10x vs GPT-3)
  - Embedding: mehr als 12000 Dimensionen
  - kann 25000 Wörter auf einmal verarbeiten (8x vs GPT-3)
  - mutlimodal: versteht Text und Bilder
  - kann sich an 64000 Wörter in einer Konversation erinnern (8x vs GPT-3.5)

[6] OpenAI. GPT-4 Technical Report, Mar. 2023. arXiv:2303.08774 [cs]

*Literatur*

[1] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative Adversarial Nets.

[2] J. McCarthy, M. L. Minsky, N. Rochester, and C. E. Shannon. A Proposal for the Dartmouth Summer Research Project on Artificial Intelligence, August 31, 1955. *AI Magazine*, 27(4):12–12, Dec. 2006. Number: 4.

[3] OpenAI. GPT-4 Technical Report, Mar. 2023. arXiv:2303.08774 [cs].

[4] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever. Improving Language Understanding by Generative Pre-Training.

[5] A. M. Turing. Computing machinery and intelligence. *Mind*, LIX(236):433–460, Oct. 1950.

[6] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention Is All You Need, Dec. 2017. arXiv:1706.03762 [cs].