

Deep Learning mit Python & Keras



https://github.com/PhilippBongartz/GfU_DeepLearning_2022

Philipp Bongartz

11.04.2022 - 14.04.2022 Remote





Biographie

- Diplom in Mathematik, Uni Bonn
- Promotion in Bioinformatik, HITS
- Seit 2019 bei EXXETA in Mannheim
Mathematiker, Data Scientist, Machine Learning Engineer

Beratungskompetenz

- Data Science, Deep Learning, DWH & ETL, Algorithm Design
- Computer Vision
- Python-Stack: Theano, Keras, SciPy, Numpy, Matplotlib, Jupyter Notebook, Pandas, ScikitLearn
- C
- DWH: T-SQL, SQL-Server, SSIS

Sprachen

- Deutsch, Englisch

Dr. Philipp Bongartz

Data Scientist, Developer

Dr. Philipp Bongartz designt seit 2013 Algorithmen und Machine Learning Architekturen, die im engen Zusammenspiel komplexe Probleme performant lösen. Er sammelte im Rahmen seiner Promotion Erfahrung im Deep Learning, im klassischen Algorithm Design und in der statistischen Analyse großer Datenmengen. Bei EXXETA ist er neben den Themen Machine Learning und Data Science auch im Data Warehousing aktiv.

Auszug relevante Projekterfahrung

Lead Data Scientist, App zum Auslesen von Personalausweisen, Produktentwicklung

- Implementierung einer ML-Pipeline zum Auslesen von Personalausweisen mit Komponenten aus Bilderkennung und -bearbeitung, OCR, Optimierung, Dynamic Programming, Clustering, sowie Deep Learning.

Machine Learning Engineer, Vorhersage von Strompreisen, Produktentwicklung

- Entwicklung und Implementierung einer KI-Anwendung für die Echtzeit Preisvorhersage des Intraday Stromhandels mit Rekurrenten Neuronalen Netzen

Data Scientist, Predictive Maintenance auf Basis von Sensordaten, Schienen-Logistik Firma

- Entwicklung eines Self-Supervised Deep Learning Models zur Klassifizierung von Zeitreihen
- Entwicklung einer regelbasierten Zeitreihenklassifikation für Batteriespannungsverläufe im Rahmen von Monitoring und Predictive Maintenance

Data Scientist, Modellierung von Vibrationsübertragung in Sportwagen, Sportwagenhersteller

- Datengetriebene Vibrationsreduktion im Sportwagen durch kausale Analyse der Anregungspfade verschiedener Frequenzbänder des Innenraumschalls mittels FFT, Causal Inference und Zeitreihenmodellierung.

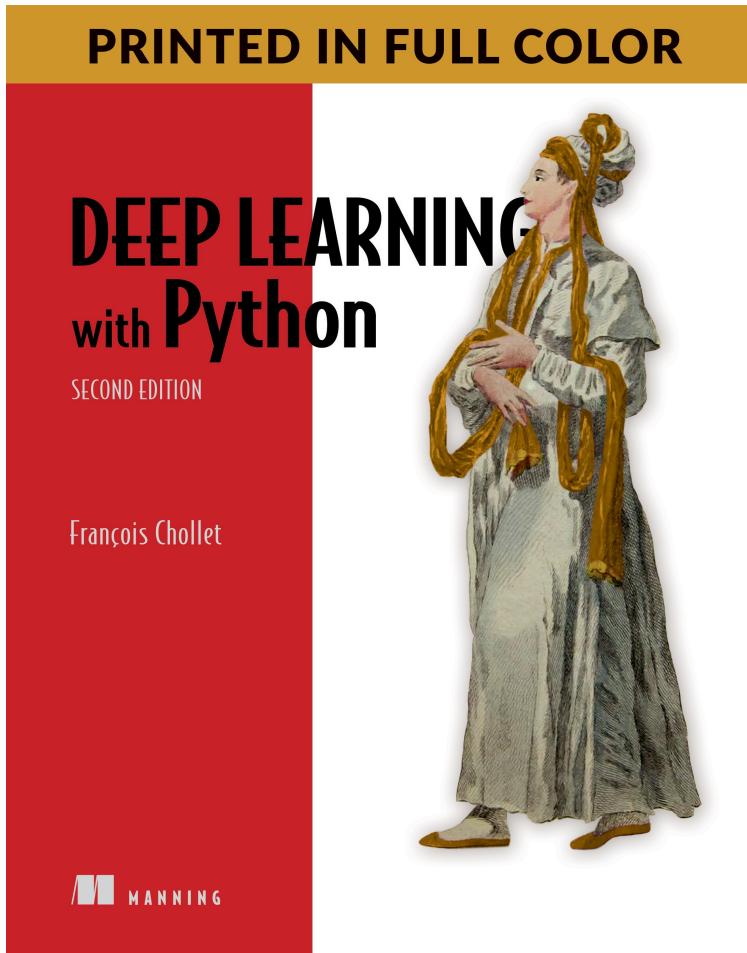
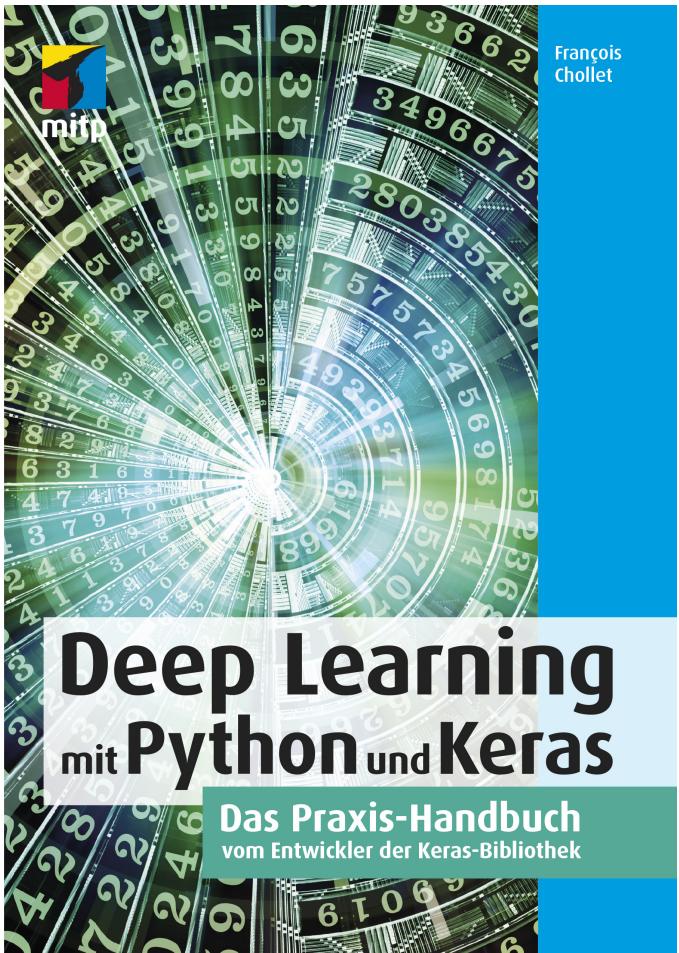


Vorstellungsrunde

- 1. Wie heißt ihr? Was ist euer Aufgabenfeld/Hintergrund?**
- 2. Was führt Euch hierher?**
- 3. Warum interessiert Euch Deep Learning?**
- 4. Welches Vorwissen habt ihr bzgl. Python, Data Science, Machine Learning, Deep Learning?**



Der Schulung zugrunde liegendes Buch



Deep Learning mit Python & Keras

1. Tag: Tensoren & Grundlagen neuronaler Netze
2. Tag: Feedforward Networks und deren Training
3. Tag: Convolutional Neural Networks
4. Tag: Textverarbeitung (RNNs, Transformers)

1. Tag: Einführung Deep Learning mit Python & Keras

1. Einstieg in Deep Learning

- Präsentation

2. Tensoren

- Präsentation
- Übung

3. Einführung in neuronale Netze

- Übung
- Präsentation

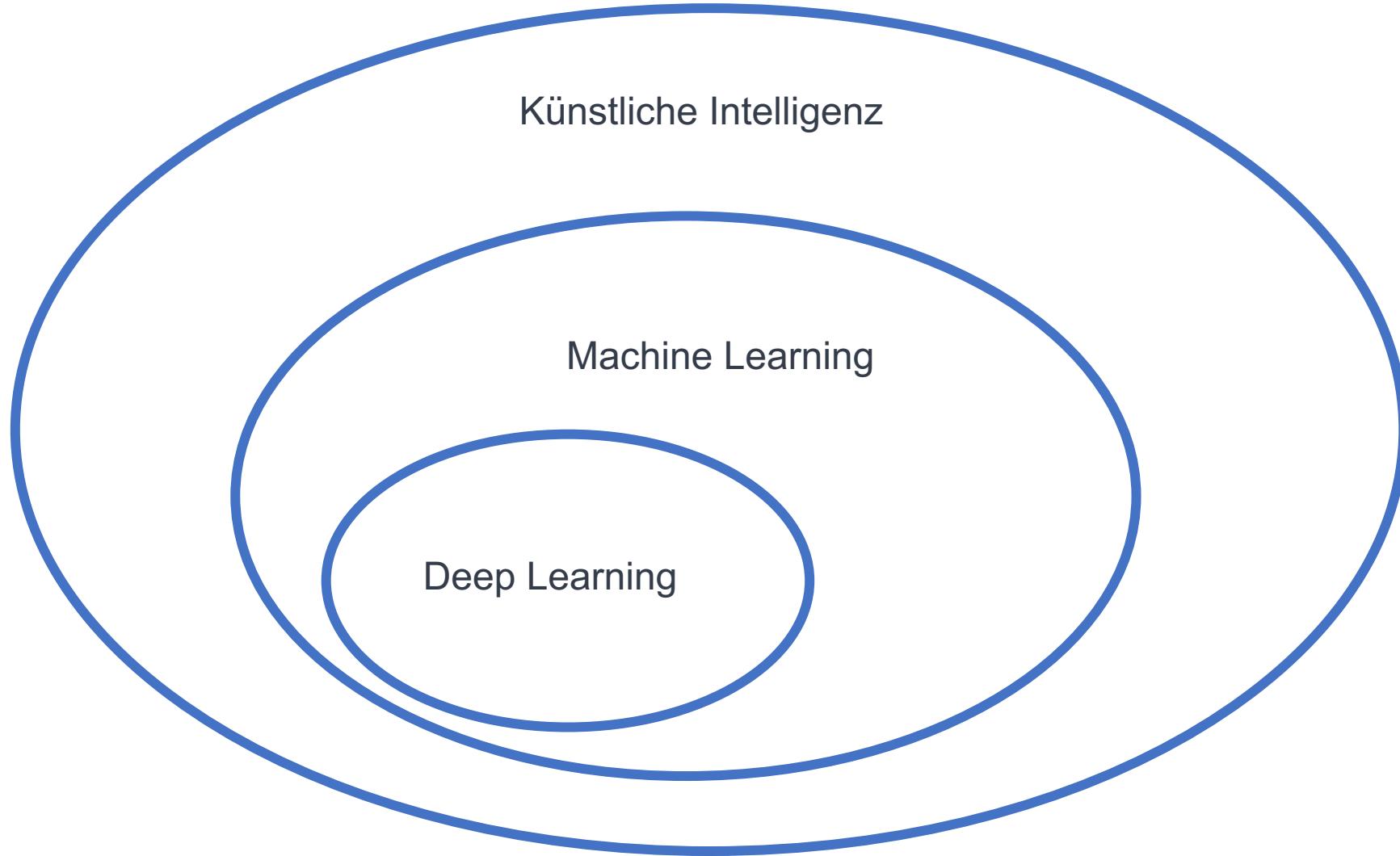
4. Gradient Descent & Backpropagation

- Präsentation
- Übung



Deep Learning Überblick

Woher komme ich, wohin gehe ich und warum heiße ich Deep Learning?



Geschichte der KI



- Dartmouth Workshop '56
- Großer Optimismus in den 60ern
- AI Winter nach Lighthill Report '74
- Expert Systems in den 80ern
- AI Winter nach Lisp Machines
- Optimierungsalgorithmen in den 90ern
- Machine Learning in den Nuller Jahren
- Deep Learning in den 2010ern

Was versteht man unter Künstlicher Intelligenz?

DID YOU SEE
THE CLEVERBOT-
CLEVERBOT CHAT?

I AM NOT A
ROBOT, I'M A
UNICORN.



YEAH, IT'S HILARIOUS,
BUT IT'S JUST CLUMSILY
SAMPLING A HUGE DATABASE
OF LINES PEOPLE HAVE
TYPED. CHATTERBOTS STILL
HAVE A LONG WAY TO GO.



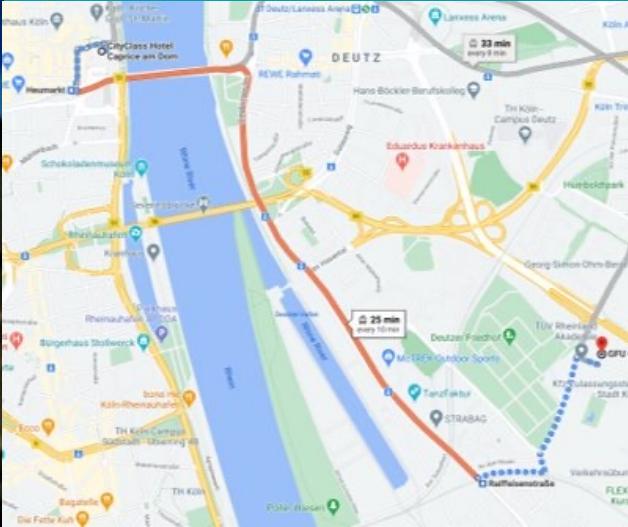
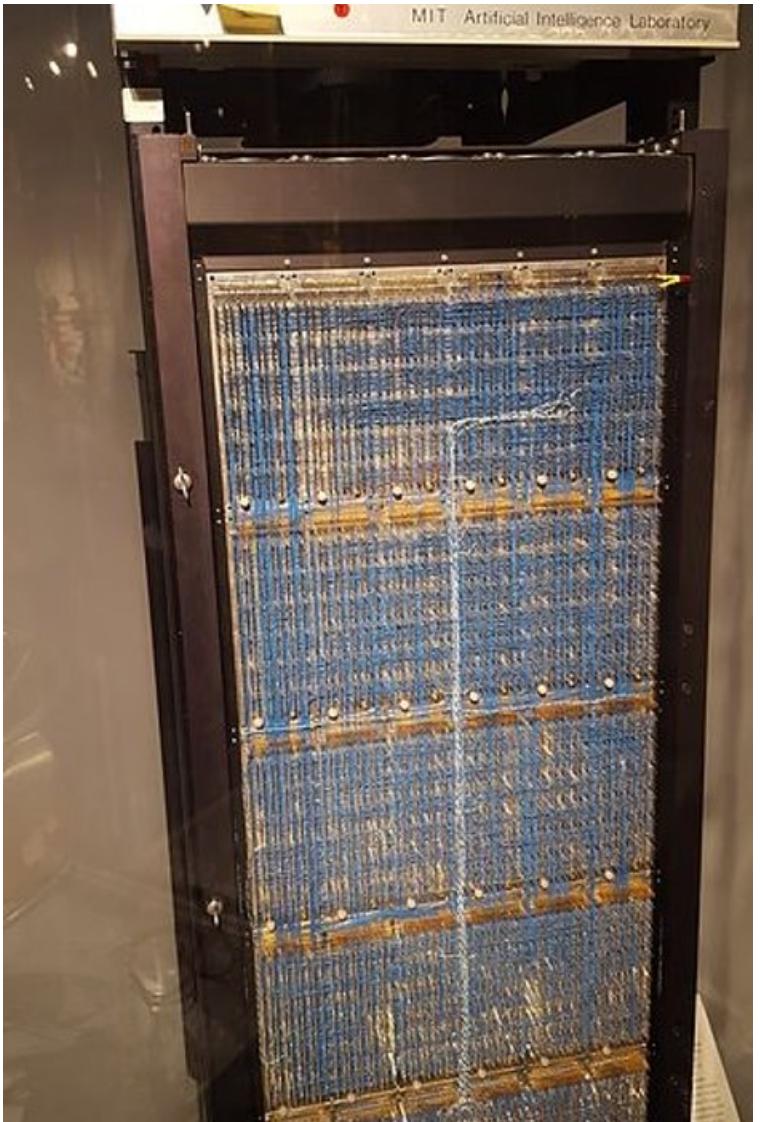
SO... COMPUTERS HAVE MASTERED
PLAYING CHESS AND DRIVING
CARS ACROSS THE DESERT, BUT
CAN'T HOLD FIVE MINUTES
OF NORMAL CONVERSATION?



IS IT JUST ME, OR
HAVE WE CREATED
A BURNING MAN
ATTENDEE?



Was versteht man unter KI?



Künstliche Intelligenz ist im weitesten Sinne die Automatisierung menschlicher Hirnleistung.



Was versteht
man unter
"menschlicher
Hirnleistung"?

Gameplay



Was versteht man unter "menschlicher Hirnleistung"?

Motorik



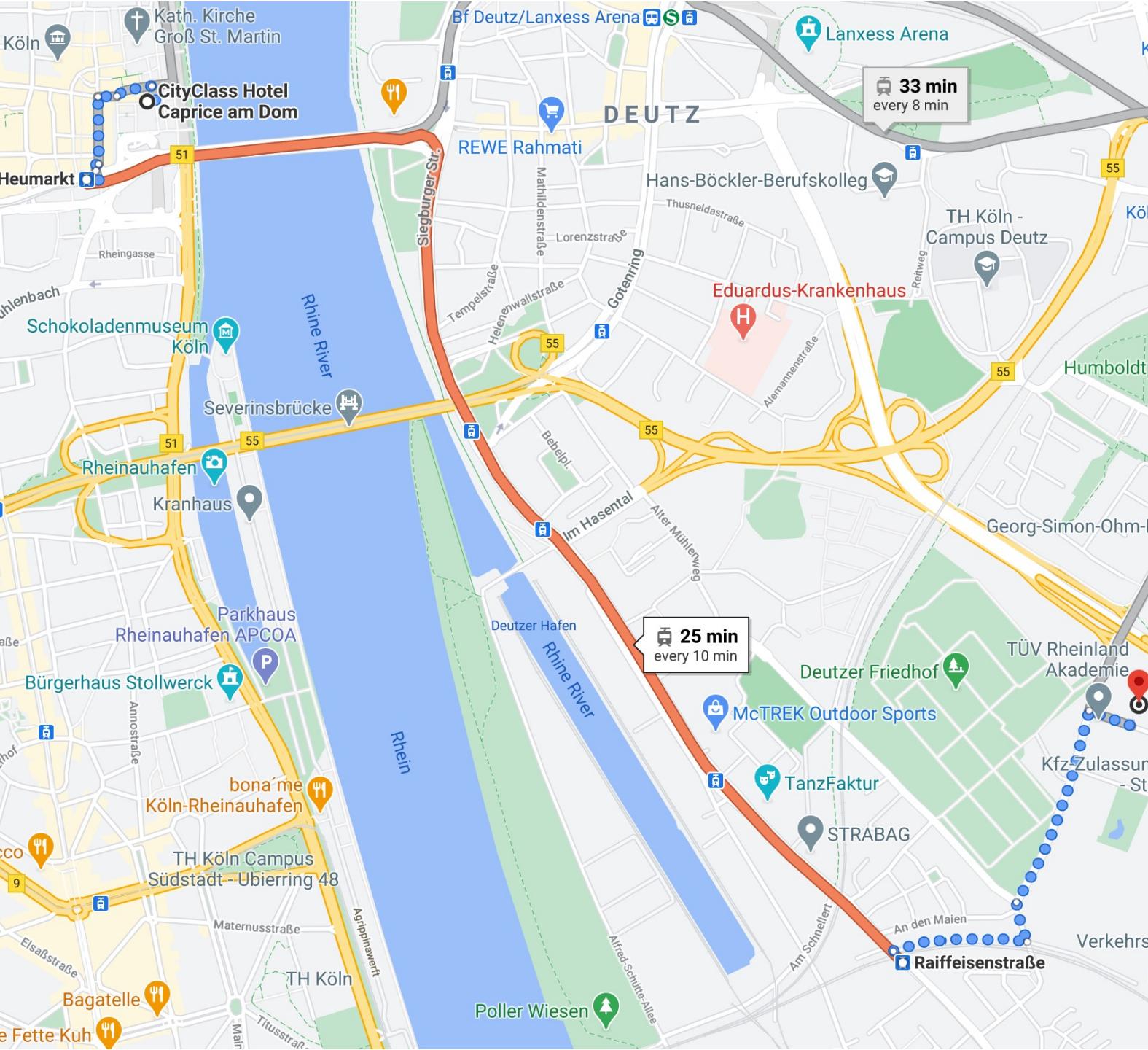
Was versteht man unter "menschlicher Hirnleistung"?

Wahrnehmung



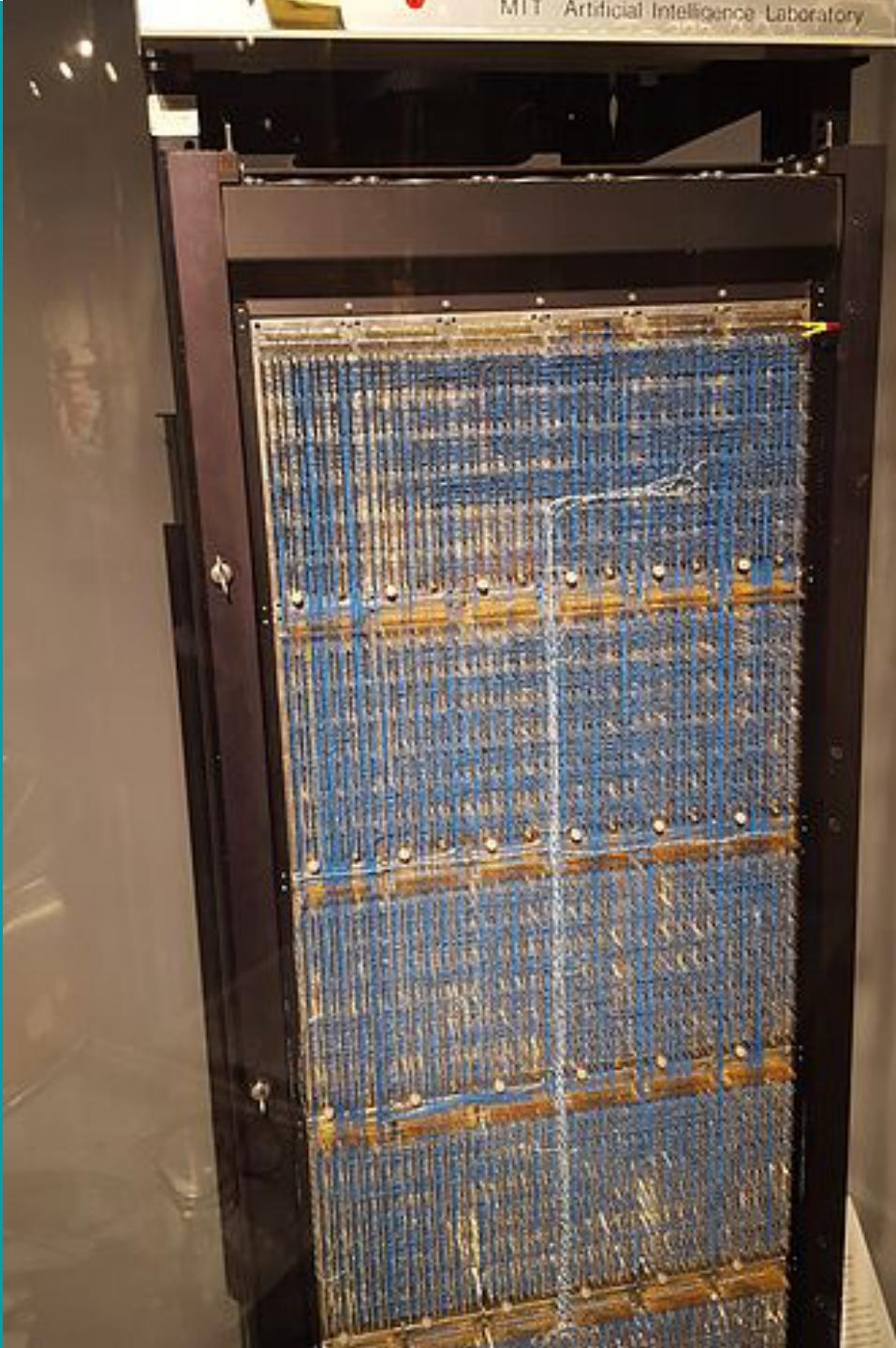
Was versteht man unter "menschlicher Hirnleistung"?

Navigation



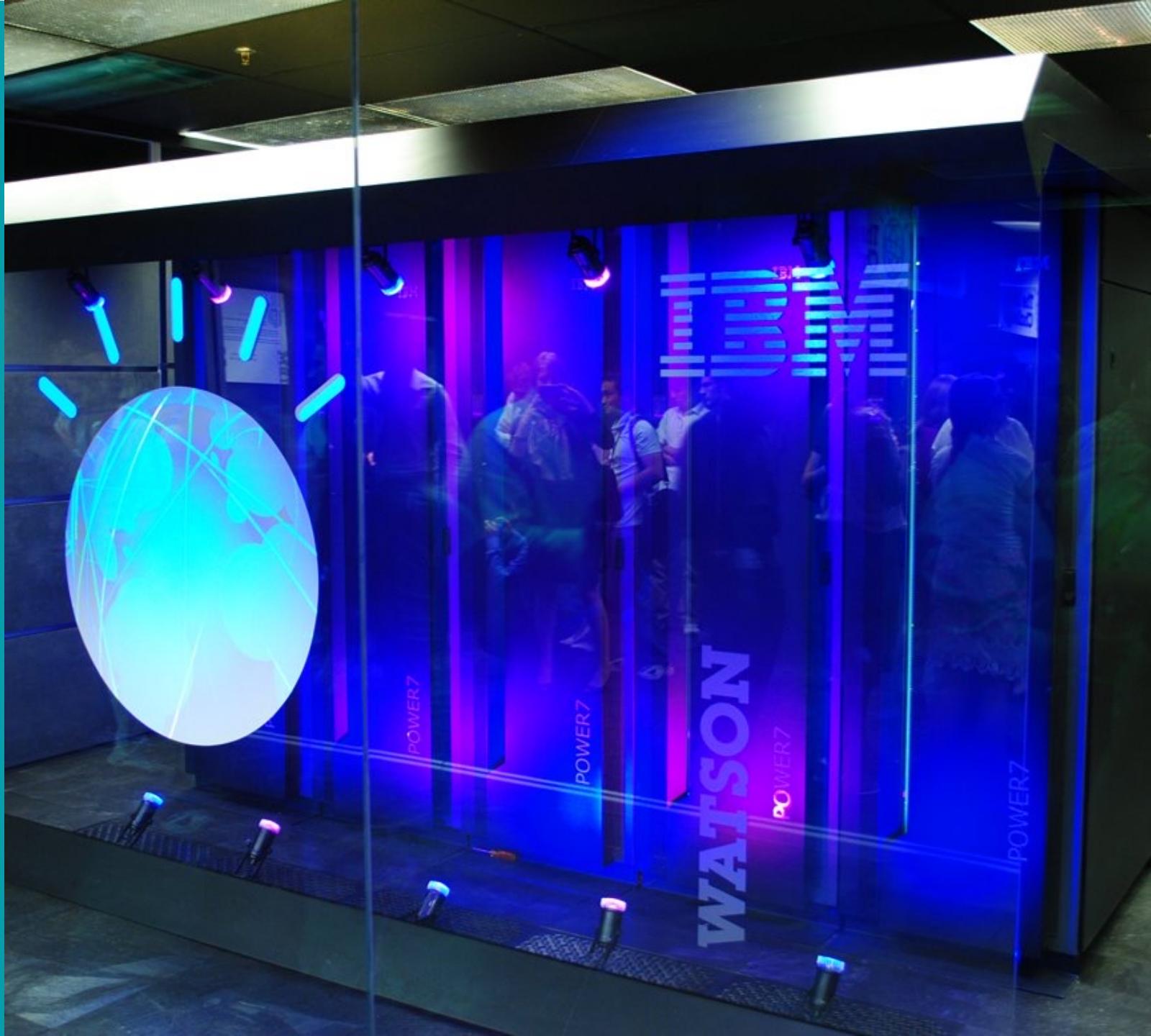
Was versteht man unter "menschlicher Hirnleistung"?

Logik



Was versteht man unter "menschlicher Hirnleistung"?

Kommunikation und
Textverständnis



Was ist Machine Learning?

Unter Machine Learning fallen alle Algorithmen, die durch Erfahrung besser darin werden, eine Aufgabe zu erledigen. Zum Beispiel:

- Clustering (K-means, K-Nearest-Neighbours, DBScan, ...)
- Regression Analysis (Linear Regression, Nonlinear Regression, ...)
- Decision Trees (Random Forests, BoostedTrees, ...)
- Support Vector Machines
- ...



Machine Learning lässt sich einteilen in:



Supervised Learning

- Im Supervised Learning sind Daten *mit zugehörigem Label* gegeben.
- Ziel des maschinellen Lernalgorithmus ist es das Label vorherzusagen.
- Supervised Learning lässt sich einteilen in:

Klassifikation

Regression



Unsupervised Learning

- Im Gegensatz zum Supervised Learning sind beim Unsupervised Learning Daten *ohne zugehöriges Label* gegeben.
- Daher ist hier in der Regel das Ziel eines maschinellen Lernalgorithmus unbekannte Muster oder Strukturen in den gegebenen Daten zu finden.
- Typische Problemstellungen des Unsupervised Learnings:

Clustering

Dimensionsreduktion

Erkennen von
Ausreißern

Datenvisualisierung

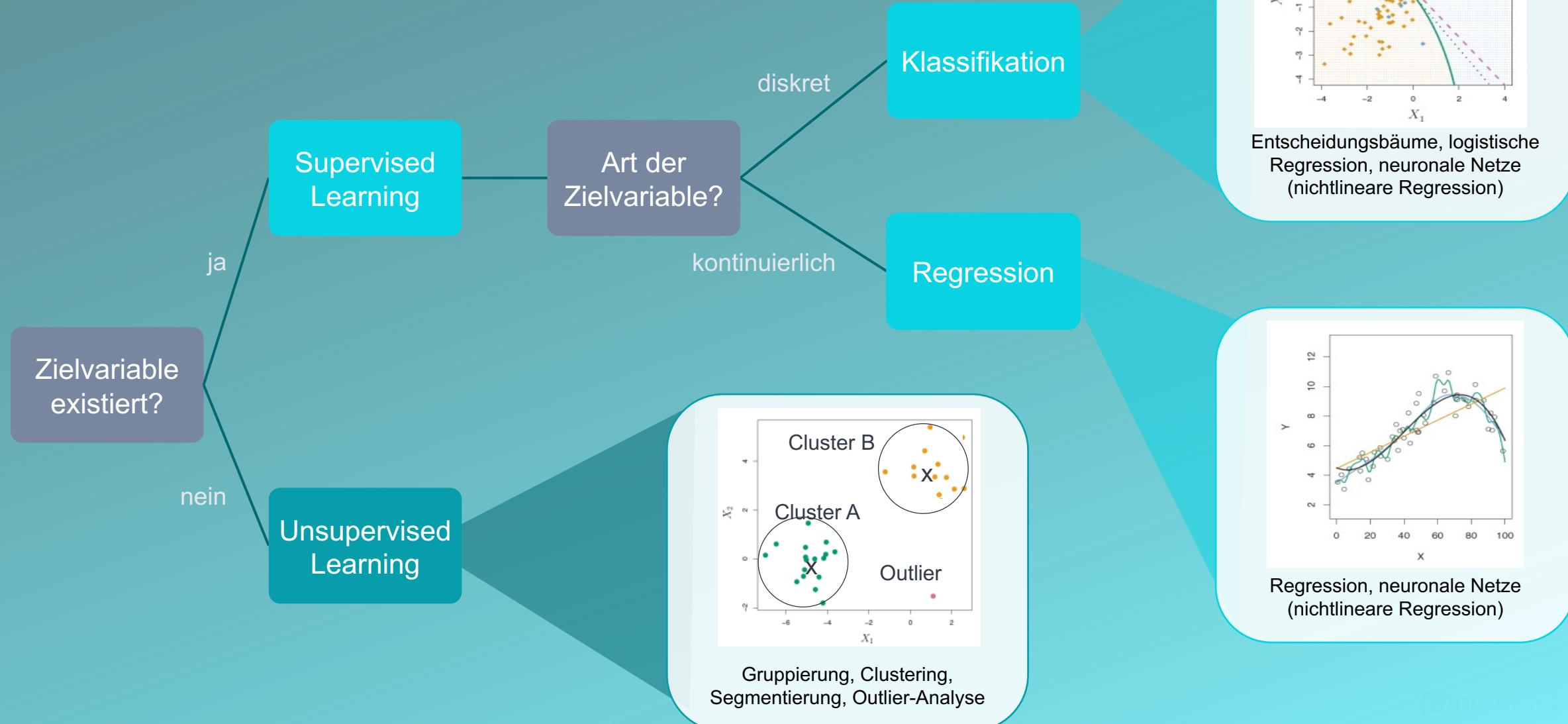
Generierung

Assoziationen

Data Mining



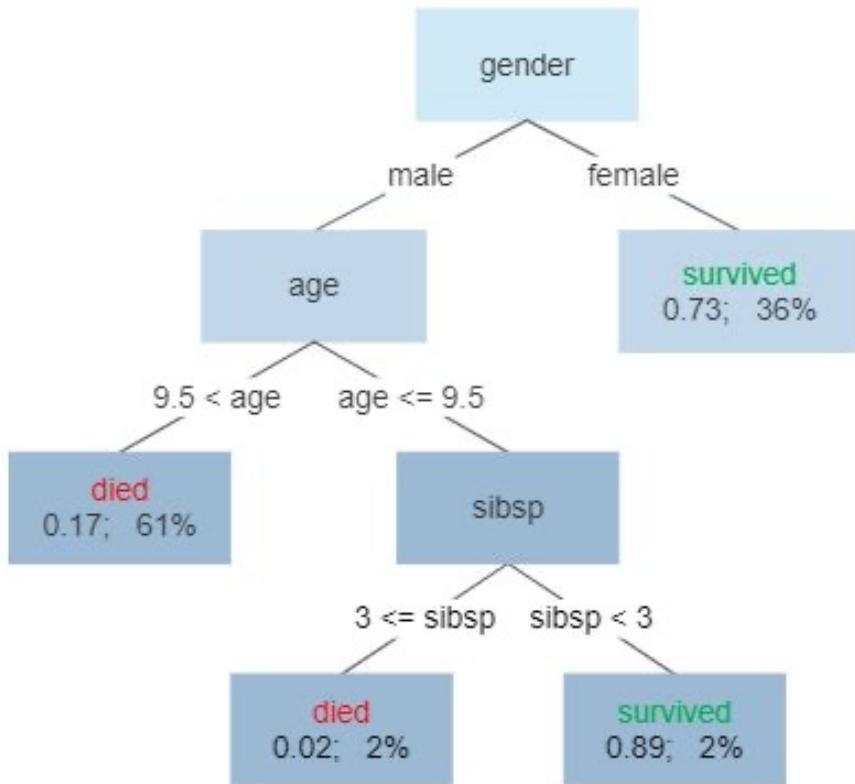
Taxonomie der statistischen Lernverfahren



Entscheidungsbäume – Inzwischen wichtigste ML-Algorithmen neben DL

Einfach aber mächtig

Survival of passengers on the Titanic

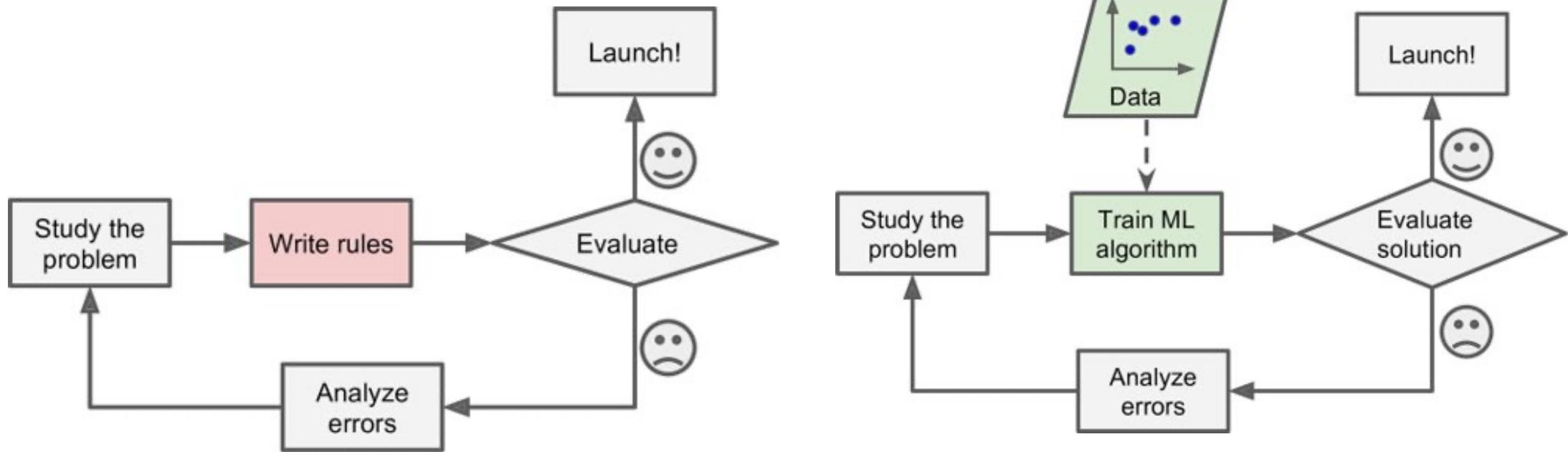


Boosting: Jeder neue Baum versucht die Fehler der bisherigen Bäume zu verringern.

Bagging: Jeder Baum wird auf einem zufälligem Teil der Daten trainiert, am Ende wird ein Konsensus aller Bäume berechnet.

Warum Machine Learning?

ML skaliert mit Compute und Data – Expert Systems nicht



Deep Learning

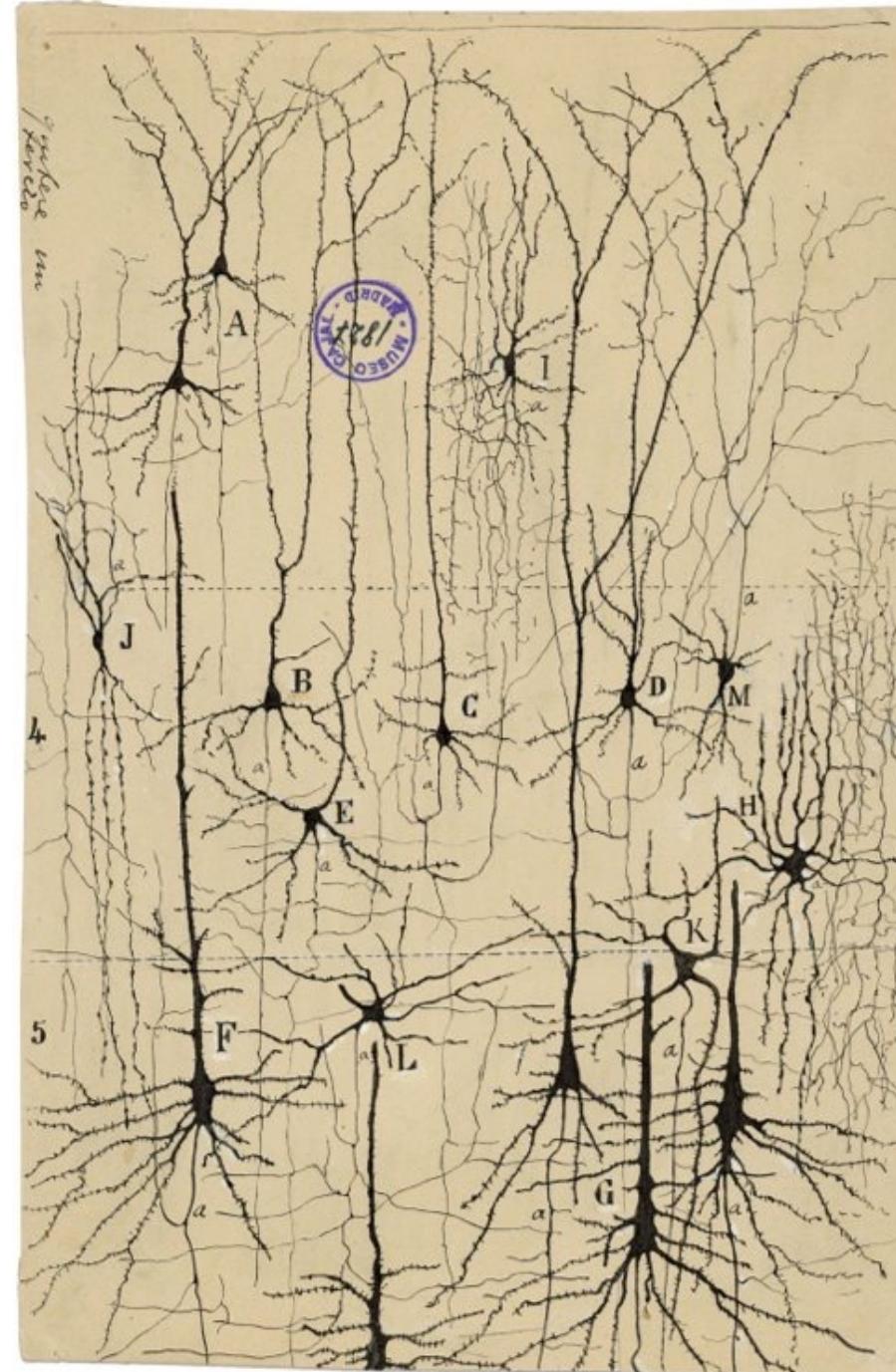


Das Neuron

Vom biologischen zum künstlichen Neuron

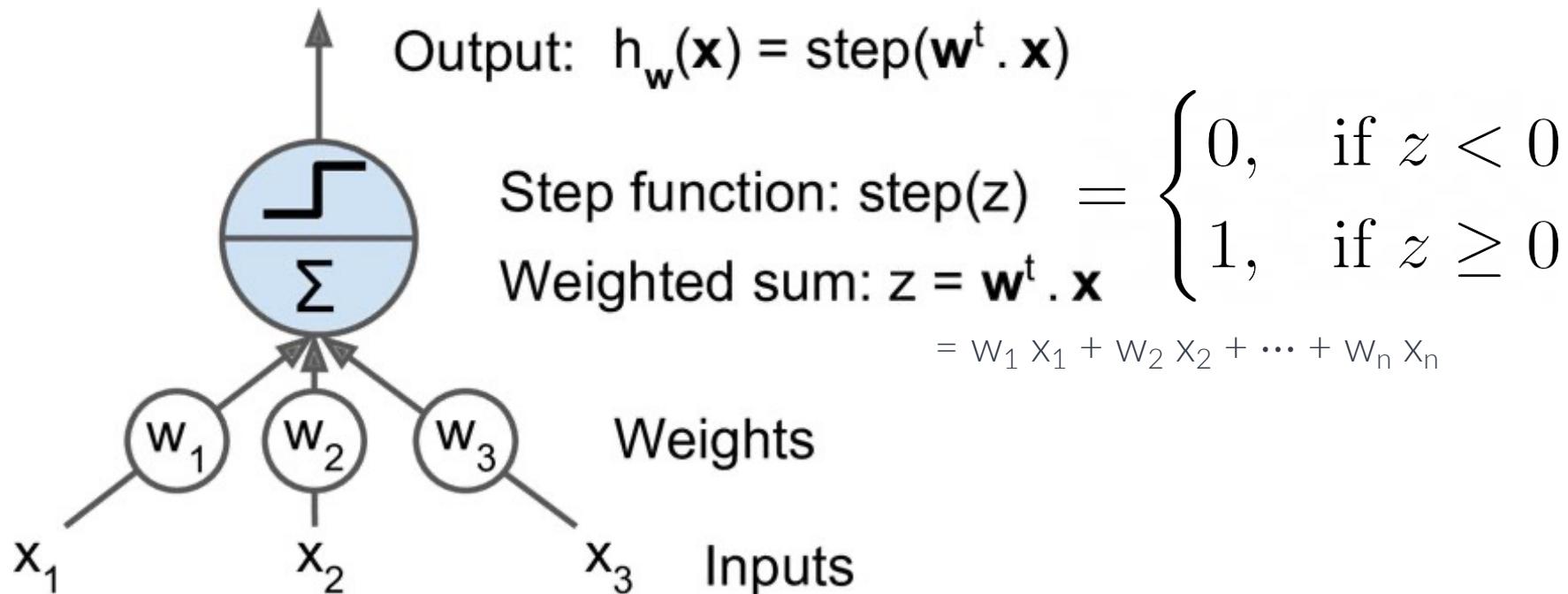
- 1943 Warren McCulloch und Walter Pitts
- Threshold Logic Unit oder Linear Threshold Unit
- Eine Version von „integrate and fire“

$$y_k = \varphi \left(\sum_{j=0}^m w_{kj} x_j \right)$$



Wiki Commons: Santiago Ramon y Cajal

Das künstliche Neuron



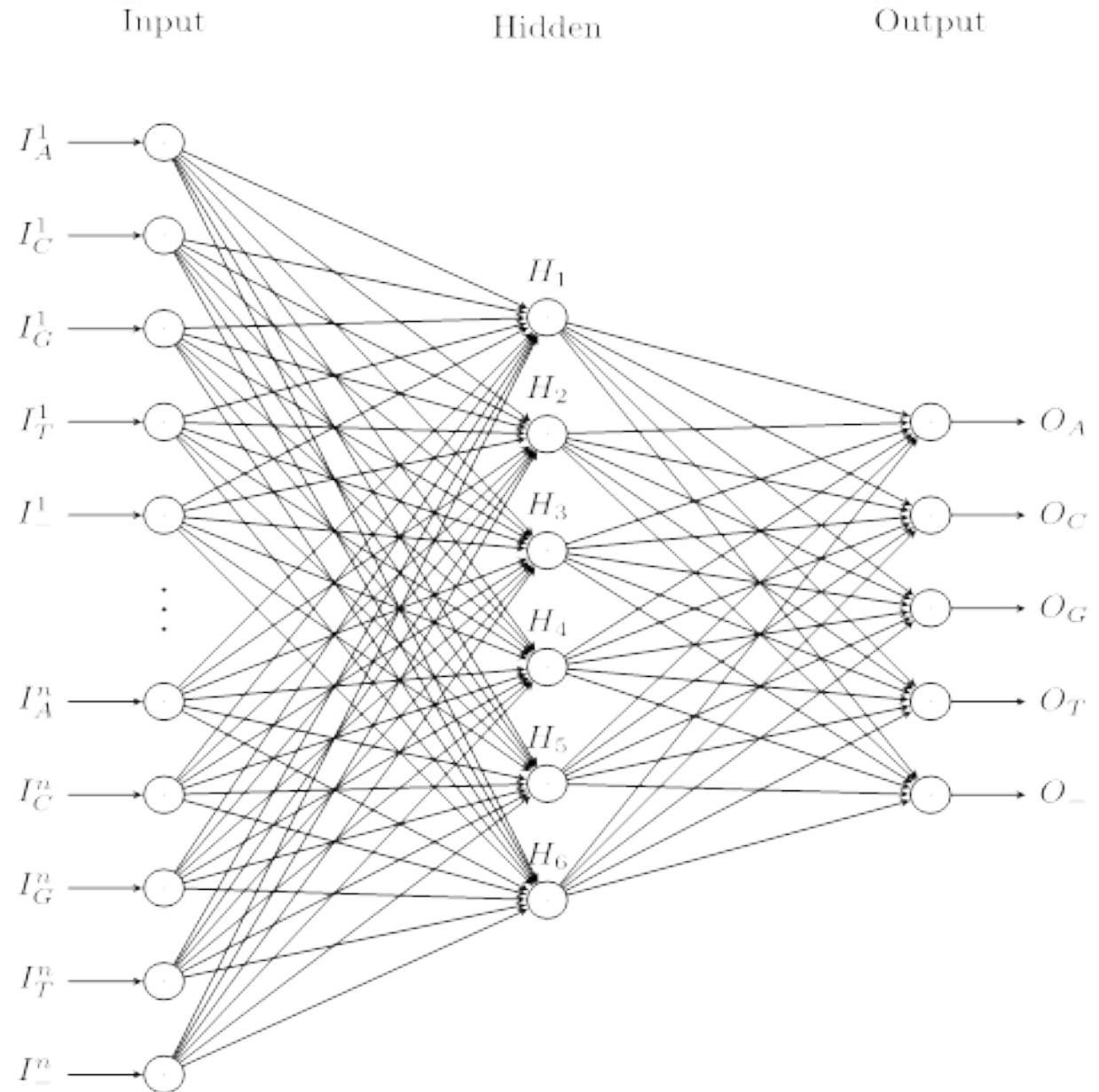
Source: Hands-on Machine Learning, Aurelion Geron

- Ein einfacherer Klassifizierer
- Linear kombiniert Inputs und gibt nach der Schrittfunktion 0 oder 1 zurück je nachdem ob das Ergebnis positiv oder negativ war

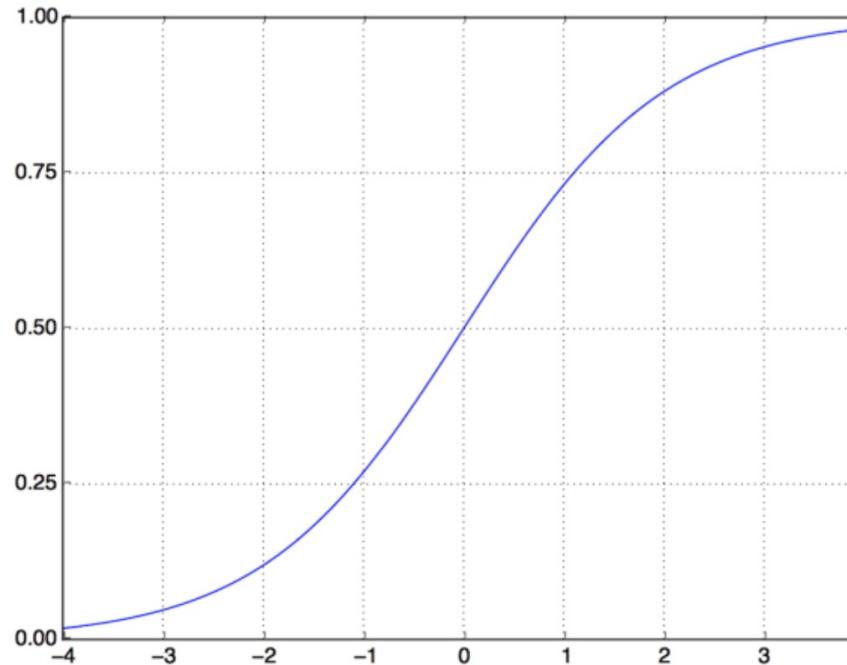
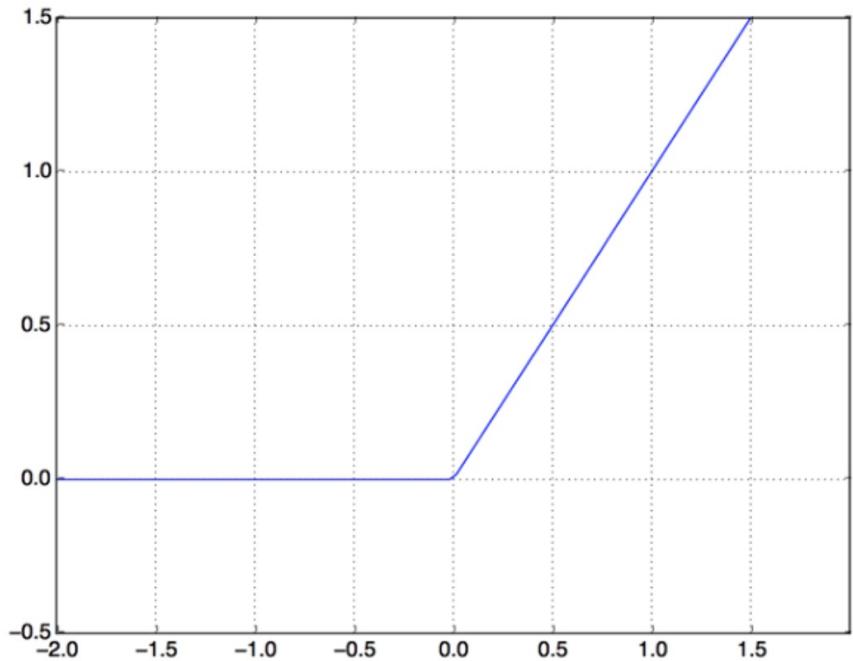


Vanilla Neural Network aka Perceptron

- Künstliche Neuronale Netzwerke:
 - Neurone werden (üblicherweise) schichtenweise in einem Netzwerk angeordnet
 - Output einer Schicht bildet Input der nächsten Schicht
- Das ursprüngliche Perceptron ist ein Classifier, d.h. es teilt Input in Klassen ein.
- Massives Stapeln von Neuronen in Schichten:
 - Deep Learning



Activation Functions



Backpropagation '86

Learning representations by back-propagating errors

David E. Rumelhart*, Geoffrey E. Hinton†
& Ronald J. Williams*

* Institute for Cognitive Science, C-015, University of California,
San Diego, La Jolla, California 92093, USA

† Department of Computer Science, Carnegie-Mellon University,
Pittsburgh, Philadelphia 15213, USA

We describe a new learning procedure, back-propagation, for networks of neurone-like units. The procedure repeatedly adjusts the weights of the connections in the network so as to minimize a measure of the difference between the actual output vector of the net and the desired output vector. As a result of the weight adjustments, internal 'hidden' units which are not part of the input or output come to represent important features of the task domain, and the regularities in the task are captured by the interactions of these units. The ability to create useful new features distinguishes back-propagation from earlier, simpler methods such as the perceptron-convergence procedure¹.

There have been many attempts to design self-organizing neural networks. The aim is to find a powerful synaptic modification rule that will allow an arbitrarily connected neural network to develop an internal structure that is appropriate for a particular task domain. The task is specified by giving the

more difficult when we introduce hidden units whose actual or desired states are not specified by the task. (In perceptrons, there are 'feature analysers' between the input and output that are not true hidden units because their input connections are fixed by hand, so their states are completely determined by the input vector: they do not learn representations.) The learning procedure must decide under what circumstances the hidden units should be active in order to help achieve the desired input-output behaviour. This amounts to deciding what these units should represent. We demonstrate that a general purpose and relatively simple procedure is powerful enough to construct appropriate internal representations.

The simplest form of the learning procedure is for layered networks which have a layer of input units at the bottom; any number of intermediate layers; and a layer of output units at the top. Connections within a layer or from higher to lower layers are forbidden, but connections can skip intermediate layers. An input vector is presented to the network by setting the states of the input units. Then the states of the units in each layer are determined by applying equations (1) and (2) to the connections coming from lower layers. All units within a layer have their states set in parallel, but different layers have their states set sequentially, starting at the bottom and working upwards until the states of the output units are determined.

The total input, x_j , to unit j is a linear function of the outputs, y_i , of the units that are connected to j and of the weights, w_{ji} , on these connections

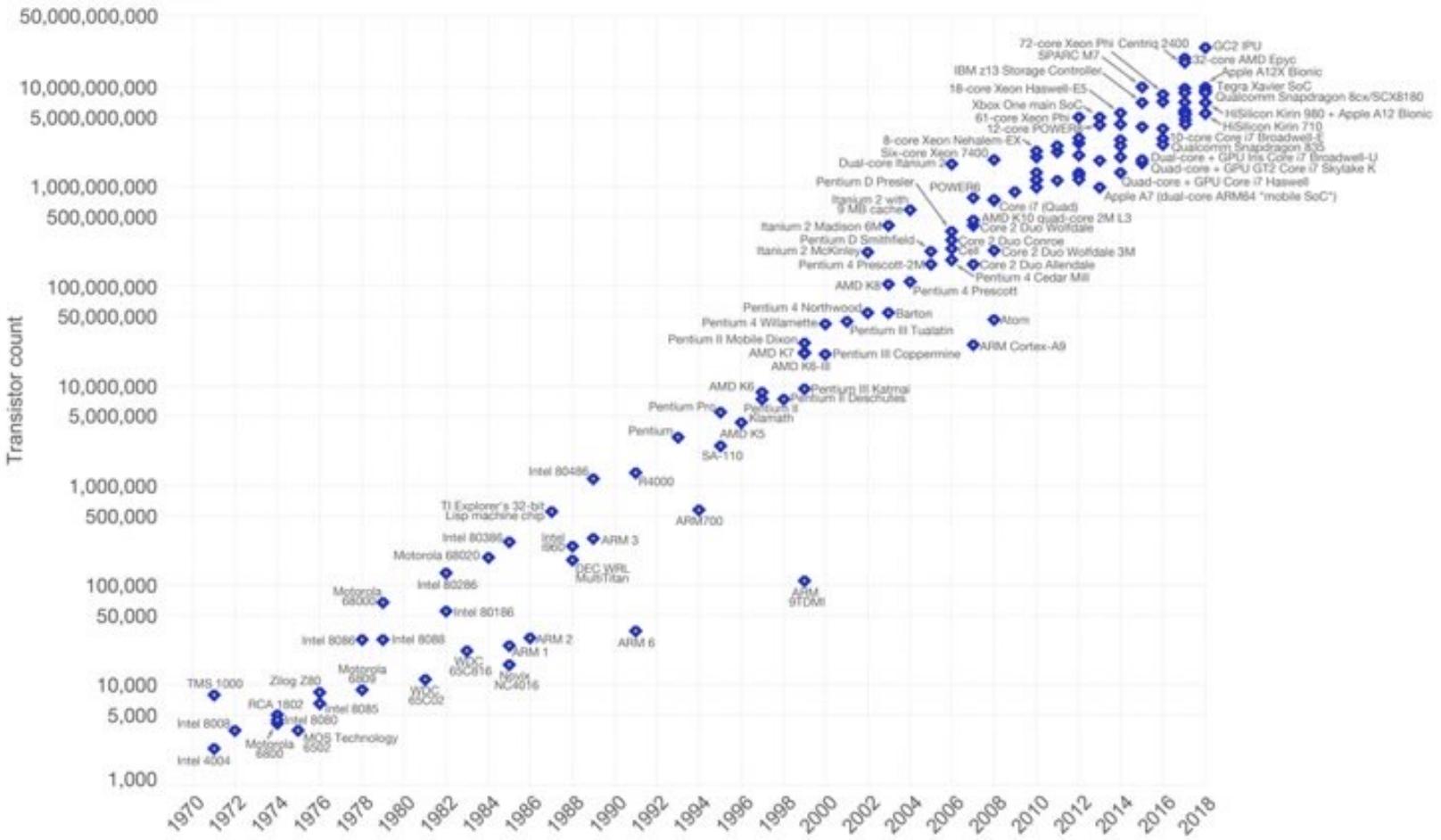
$$x_j = \sum_i y_i w_{ji} \quad (1)$$



Moore's Law & GPUs

Moore's Law – The number of transistors on integrated circuit chips (1971-2018)

Moore's law describes the empirical regularity that the number of transistors on integrated circuits doubles approximately every two years. This advancement is important as other aspects of technological progress – such as processing speed or the price of electronic products – are linked to Moore's law.



Data source: Wikipedia (https://en.wikipedia.org/wiki/Transistor_count)

The data visualization is available at OurWorldInData.org. There you find more visualizations and research on this topic.

Licensed under CC-BY-SA by the author Max Roser.



Deep Learning Durchbrüche



Deep Learning Durchbrüche



Deep Learning Durchbrüche



Deep Learning Durchbrüche

OpenAI

API

PROJECTS

BLOG

ABOUT

Better Language Models and Their Implications

We've trained a large-scale unsupervised language model which generates coherent paragraphs of text, achieves state-of-the-art performance on many language modeling benchmarks, and performs rudimentary reading comprehension, machine translation, question answering, and summarization—all without task-specific training.

February 14, 2019
24 minute read



Jupyter Notebooks



- Notebook vereint Text und Code in einzeln ausführbaren Code-Zellen
- Interaktiv, gut debugbar und gut dokumentierbar

1. Jeder öffnet das Jupyter Notebook 1.2.Jupyter_Markdown.ipynb
 - Startmenü → Anaconda Prompt → jupyter notebook
2. Wir versuchen Tensorflow zu importieren
3. Außerdem sollte NumPy, keras, matplotlib ... von der IT vorinstalliert sein