Grundlagen - Wie funktioniert KI?

Martin Kaufmann

cnlab Herbsttagung 2024: KI und Sicherheit
Gleisarena, Zürich, 4. September 2024

cnlab

© 2024

Chatbots

ChatGPT

Diese Chatbots basieren auf „Machine Learning“ und „Artificial Neural Networks“.

Gemini

Copilot

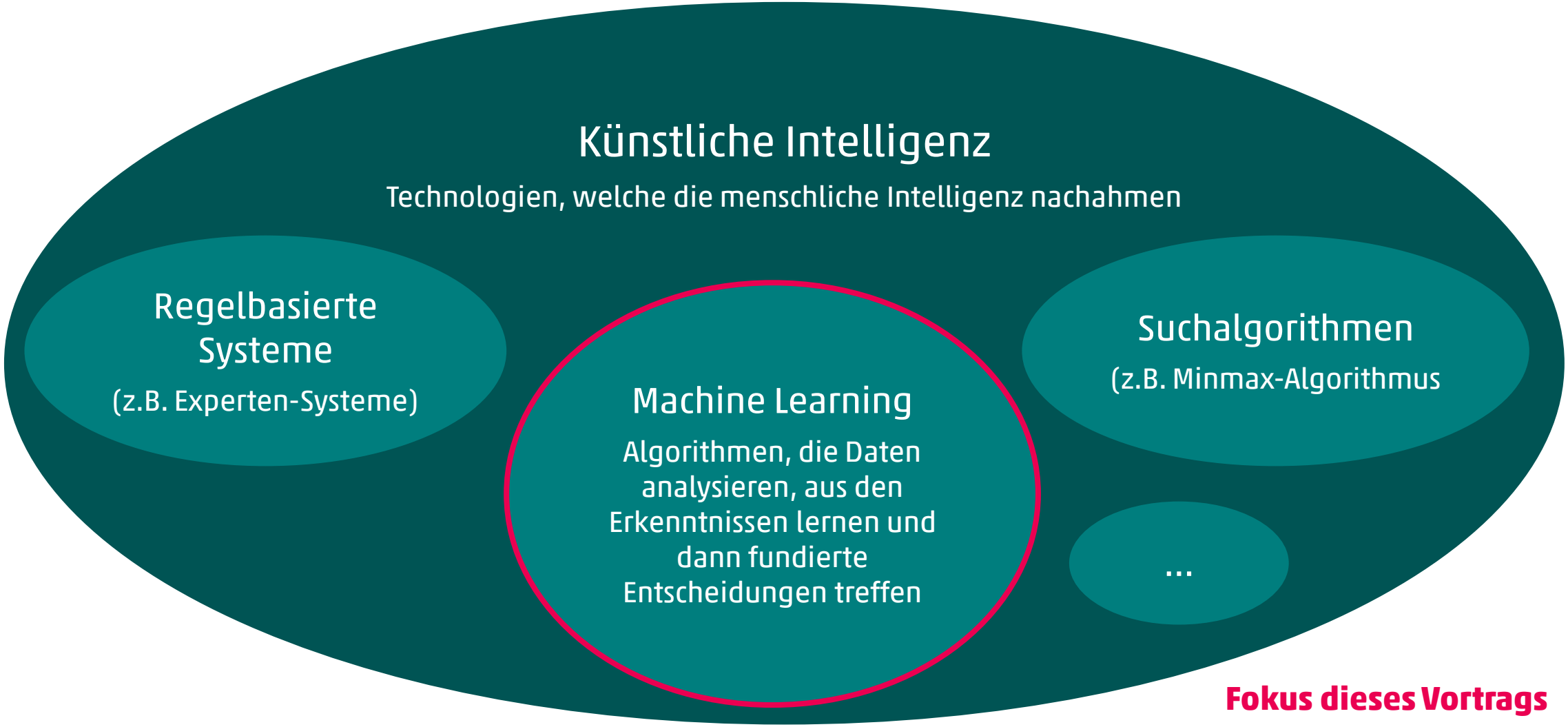
Ziele

Wir möchten verstehen…

- Was ist eigentlich KI?
- Was ist der Unterschied zwischen KI und Machine Learning?
- Wie funktioniert Machine Learning?
- Was sind Artificial Neural Networks?

Vorwarnung: ML ist viel Mathematik

Künstliche Intelligenz (KI) und Machine Learning (ML)

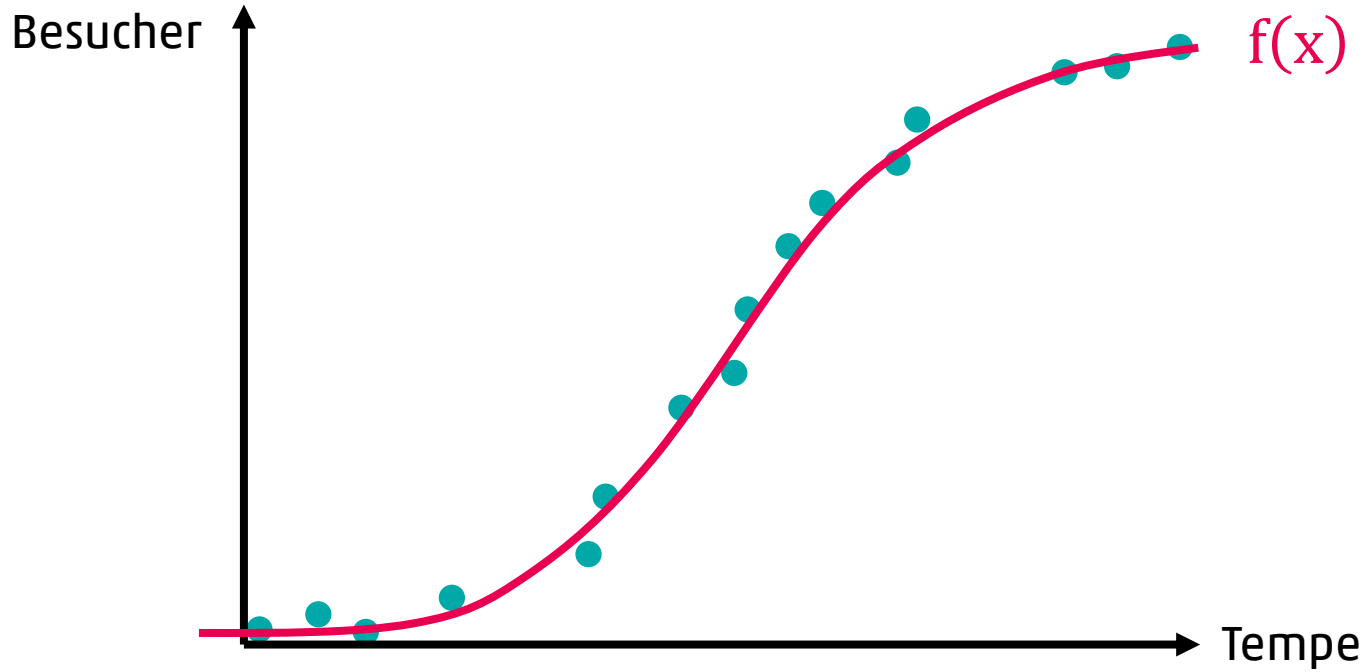


Fokus dieses Vortrags

Vorhersage (Regression)

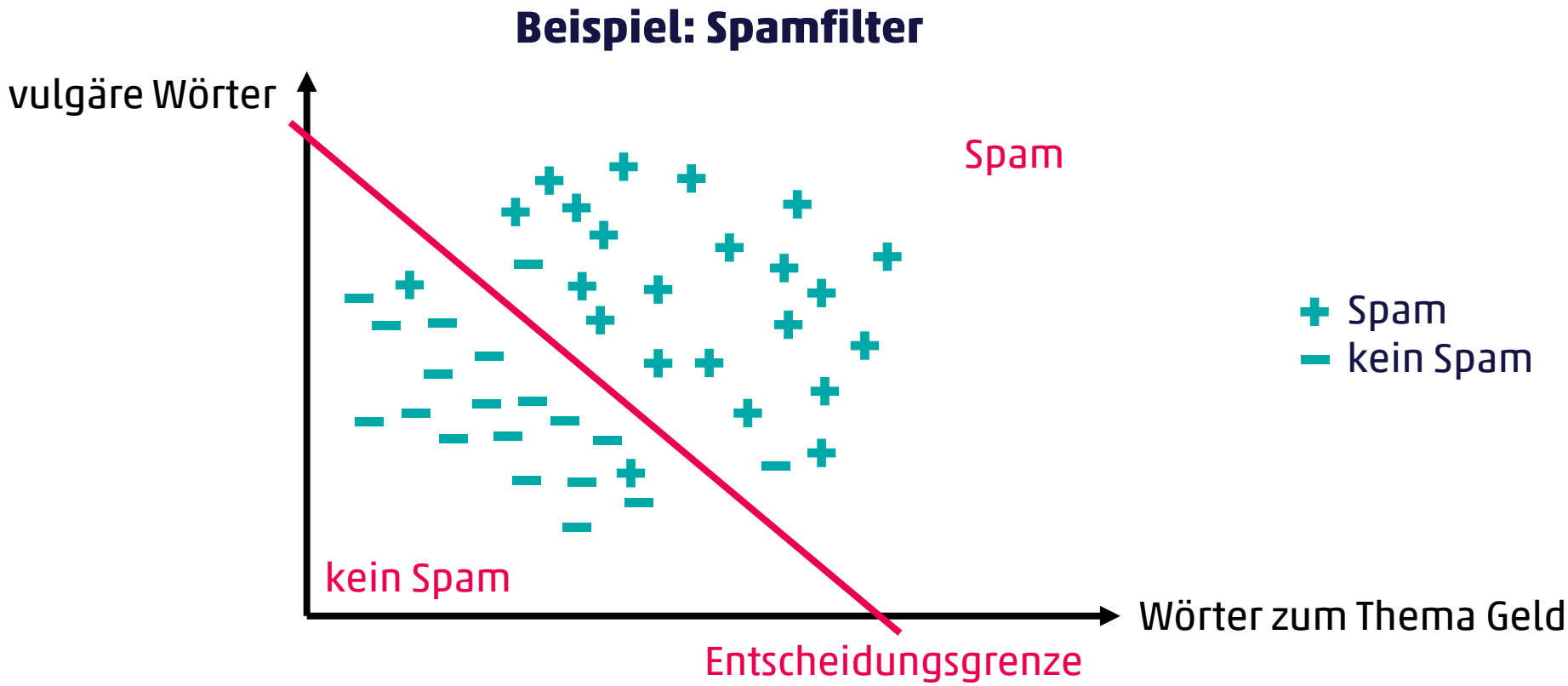
- Ziel: Vorhersage von realen Werten (z.B. Wettervorhersage)
- Gegeben: Messungen für unterschiedliche Inputs $\vec{x}$
- Gesucht: Modell (Funktion), welches die Beziehung zwischen Input $\vec{x}$ und Messung y möglichst gut beschreibt

Beispiel: Besucher im Freibad



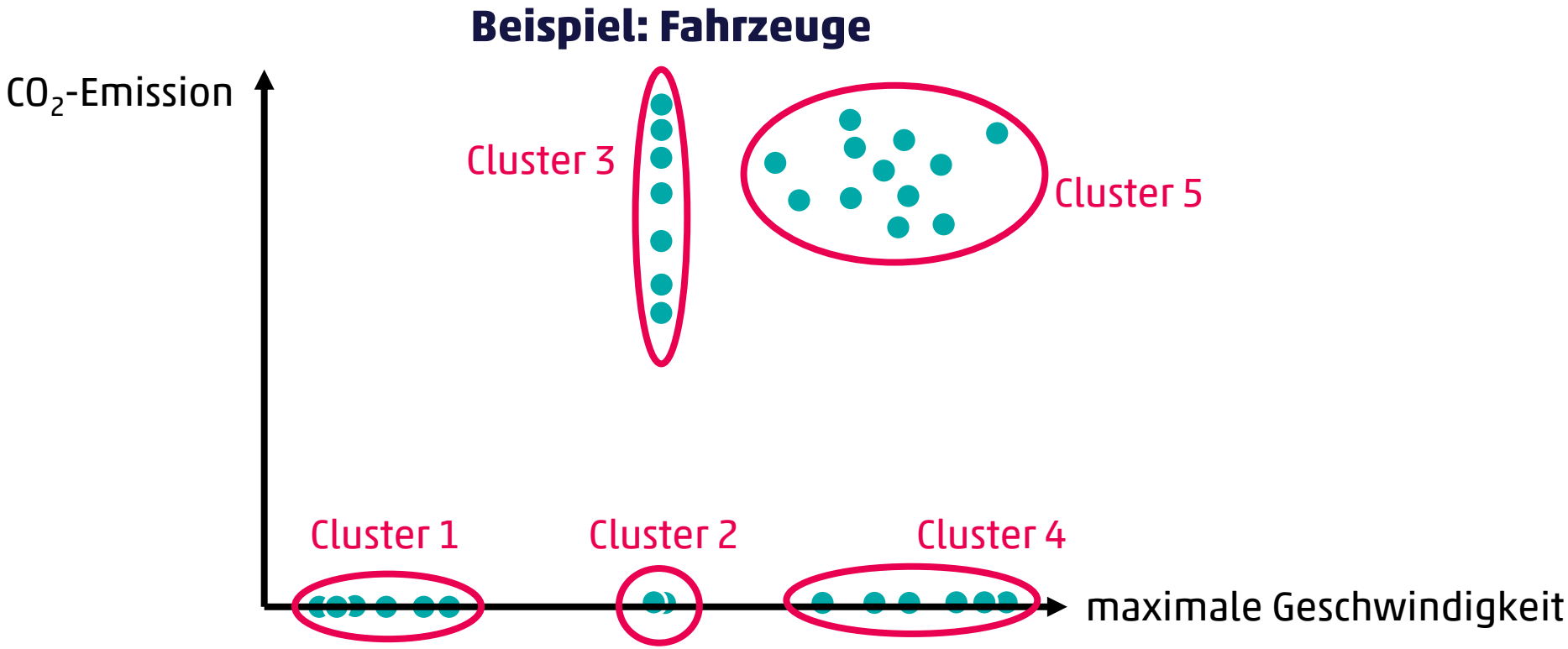
Klassifizierung (Classification)

- Ziel: Datenpunkte einer Klasse zuordnen (z.B. E-Mail → Spam / kein Spam)
- Gegeben: Datenpunkte mit Labels
- Gesucht: Modell, welches neuen Datenpunkten ein Label zuordnet

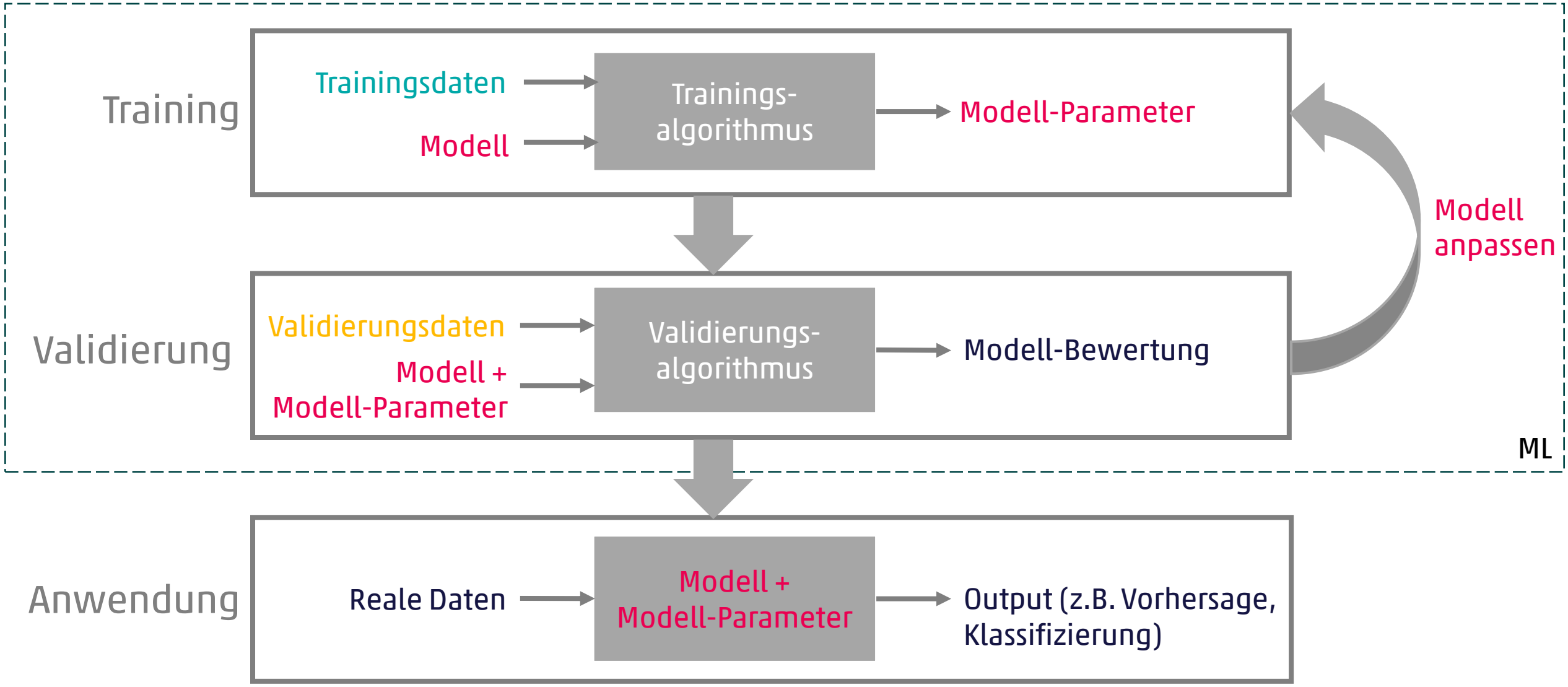


Clustering (Unsupervised Classification)

- Ziel: Entdeckung von Ähnlichkeitsstrukturen in Datensätzen
- Gegeben: Datenpunkte ohne Labels
- Gesucht: Clusters (Gruppen), sodass ähnliche Punkte im selben Cluster sind und unähnliche Punkte in unterschiedlichen Clustern

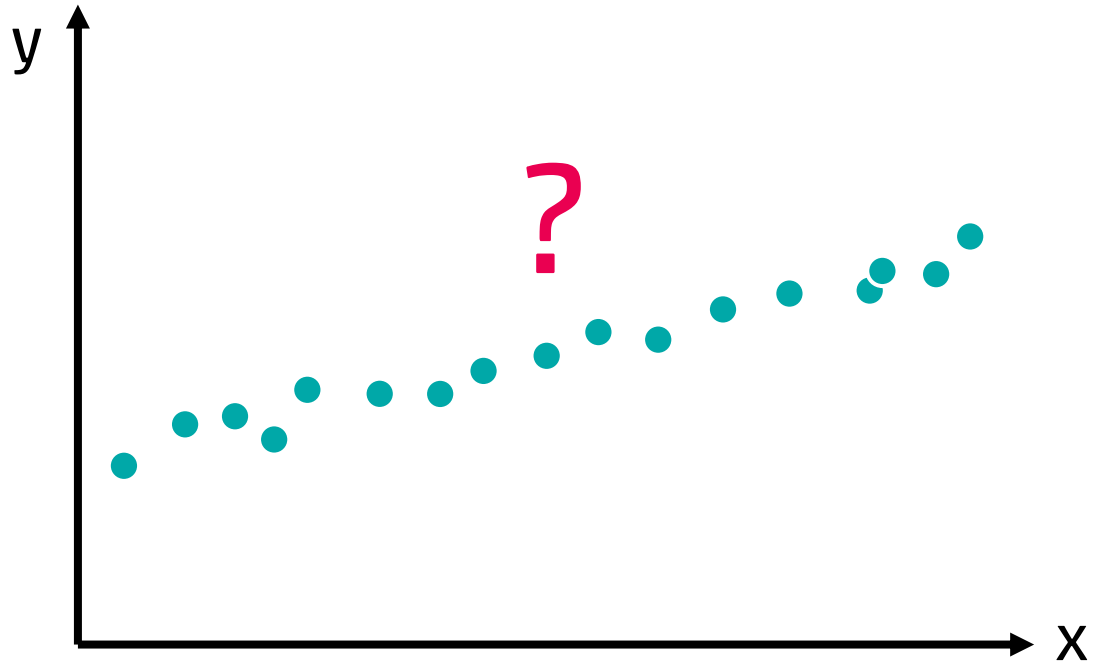


Grundprinzip von Machine Learning (ML)



Beispiel – Vorhersage (Regression)

- Ziel: Vorhersage von y für gegebenes x
- Gegeben: n Messungen y_i (Punkte) für unterschiedliche Inputs x_i
- Gesucht: Modell (Funktion), welches die Beziehung zwischen x und y möglichst gut beschreibt



Beispiel – Training

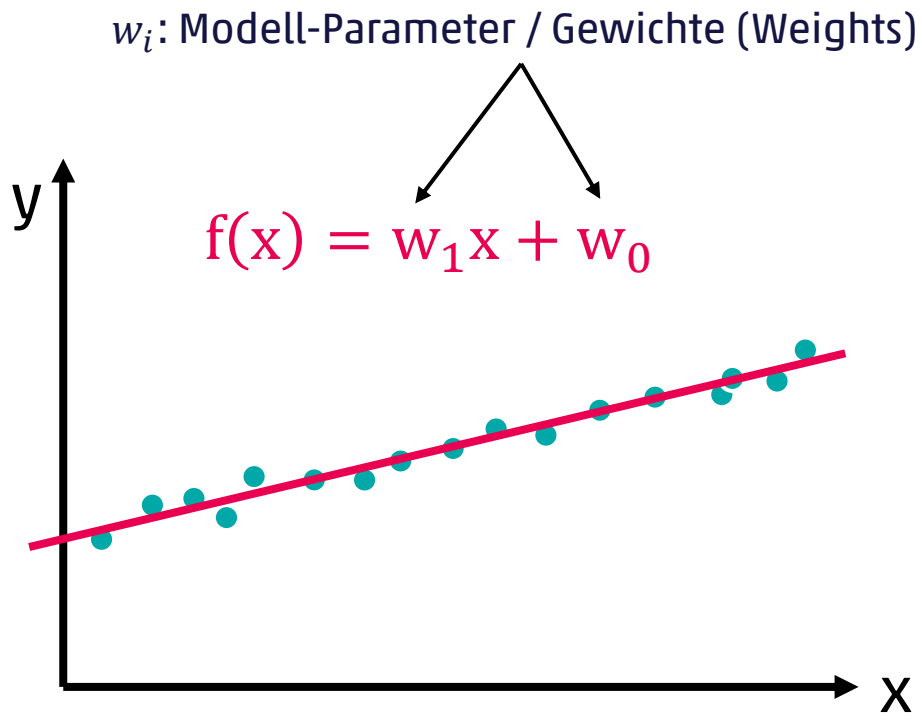
Beispiel – Wahl des Modells

Welches Modell (Funktion) sollen wir wählen?

- Lineare Funktion (Lineare Regression)
- Polynom vom Grad m
- Exponentialfunktion
- Logarithmusfunktion
- …

Die Wahl des Modells ist in der Regel nicht trivial!

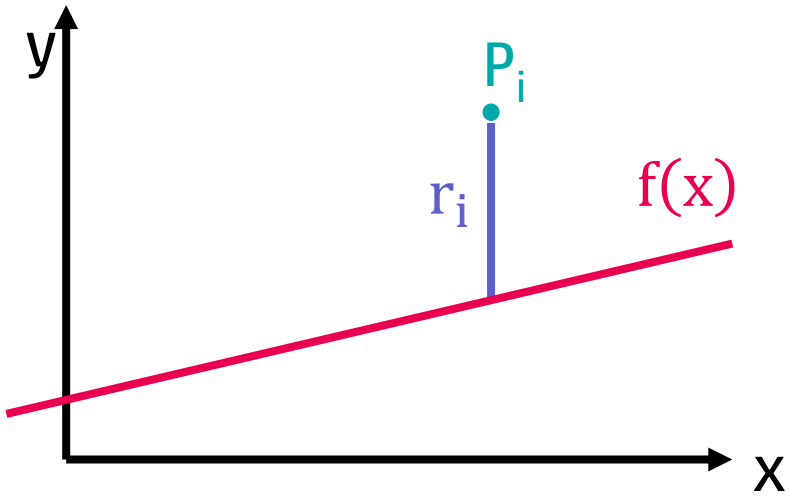
(mehr dazu später)



Beispiel – Trainingsalgorithmus

- Verwendet eine Fehlerfunktion (oft auch Verlustfunktion / „Loss Function“ genannt)
- Findet die Model-Parameter $\vec{w}_{min}$, welche den Fehler für die Trainingsdaten minimieren
  - Die optimalen Modell-Parameter können meist nicht effizient berechnet werden und man gibt sich mit einer Approximation zufrieden.
  - Für die Approximation wird beispielsweise (Stochastic) Gradient Descent verwendet.

Beispiel: Quadratischer Fehler



$$R = \sum_{i=1}^{n} r_i^2$$

Bemerkung:

Classification kann analog mit anderer Fehlerfunktion gelöst werden

Gradient Descent

- Findet ein lokales Minimum

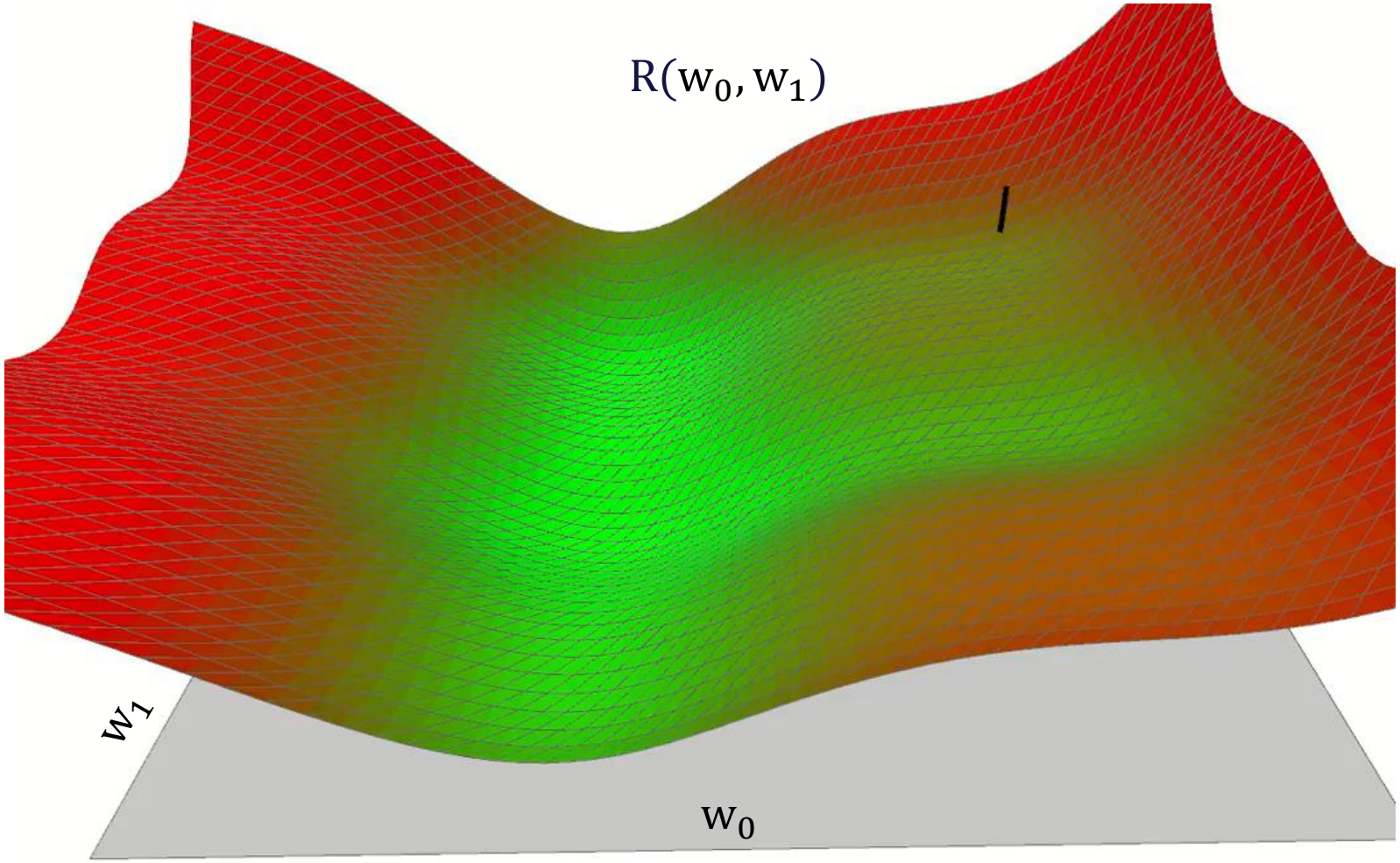
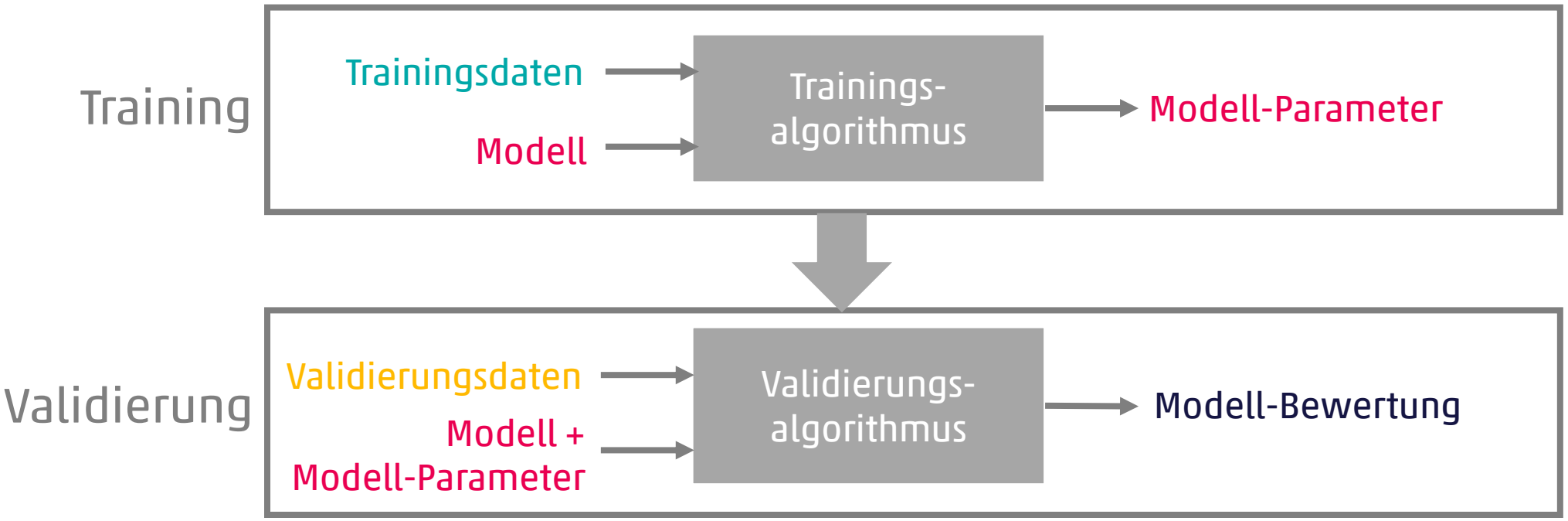


Image Source: https://www.youtube.com/watch?v=kJgx2RcJKZY

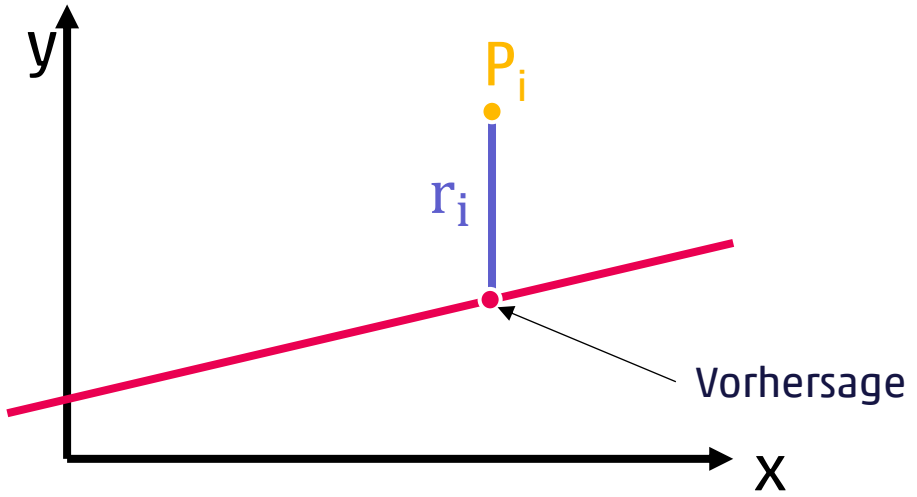
Beispiel – Validierung



Beispiel – Validierung

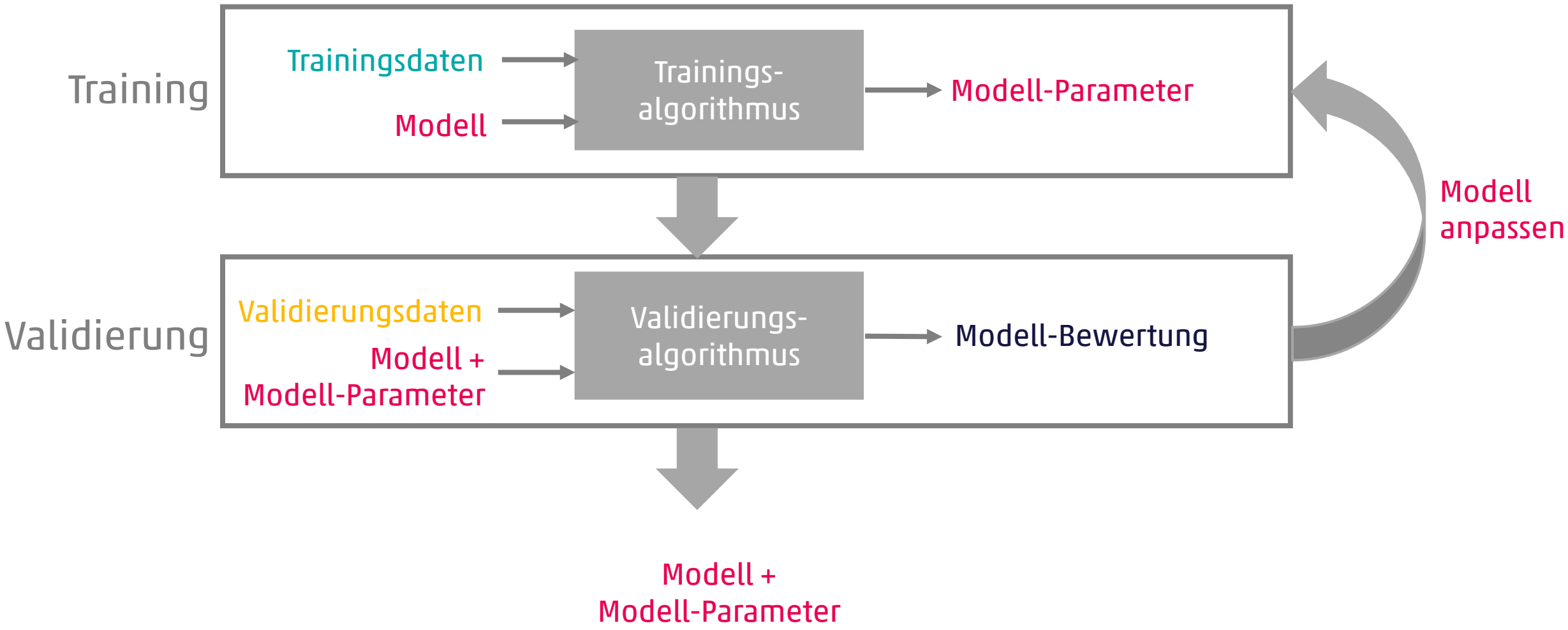
- Nutzt das Modell mit den berechneten Modell-Parametern, um für die Validierungsdaten Vorhersagen zu machen.
- Berechnet für die Validierungsdaten den Fehler zwischen den Vorhersagen und den tatsächlichen Werten (Prediction Error)

$$f(x) = w_1 x + w_0$$



$$R = \sum_{i=1}^{n} r_i^2$$

Beispiel – Validierung

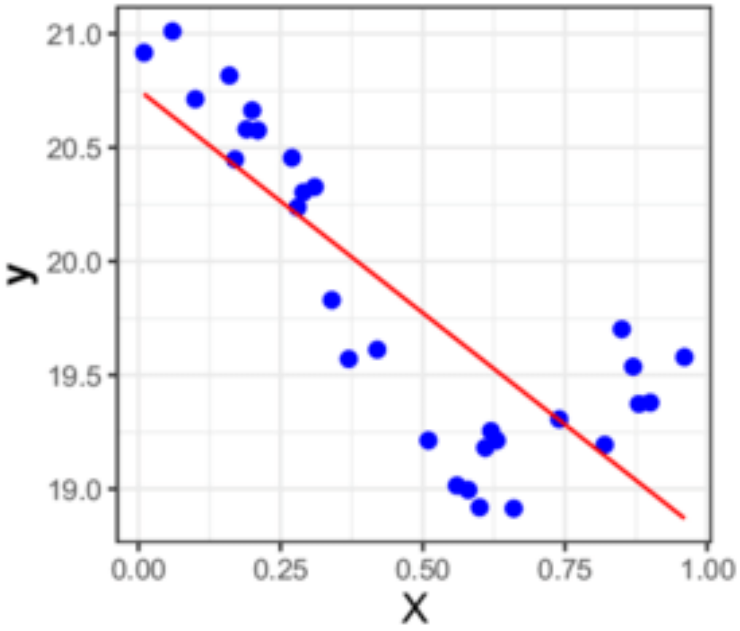


Wahl des Modells: Modell-Komplexität

- Underfitting: Modell ist zu „einfach“, um die Komplexität der Trainingsdaten zu erfassen
- Overfitting: Modell „merkt“ sich Trainingsdaten, anstatt einen Trend zu „lernen“

Polynomial fit degree 1

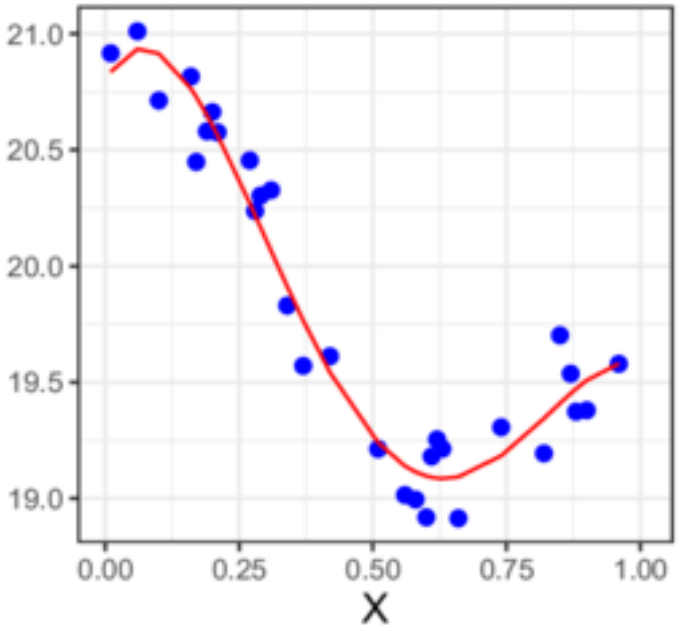
Training error: 0.4
Prediction error: 0.42



Underfit

Polynomial fit degree 4

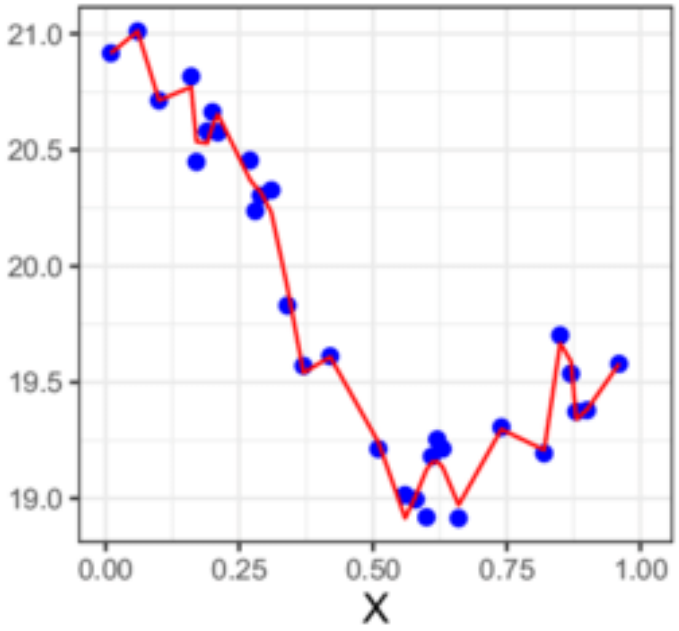
Training error: 0.14
Prediction error: 0.17



Good fit

Polynomial fit degree 20

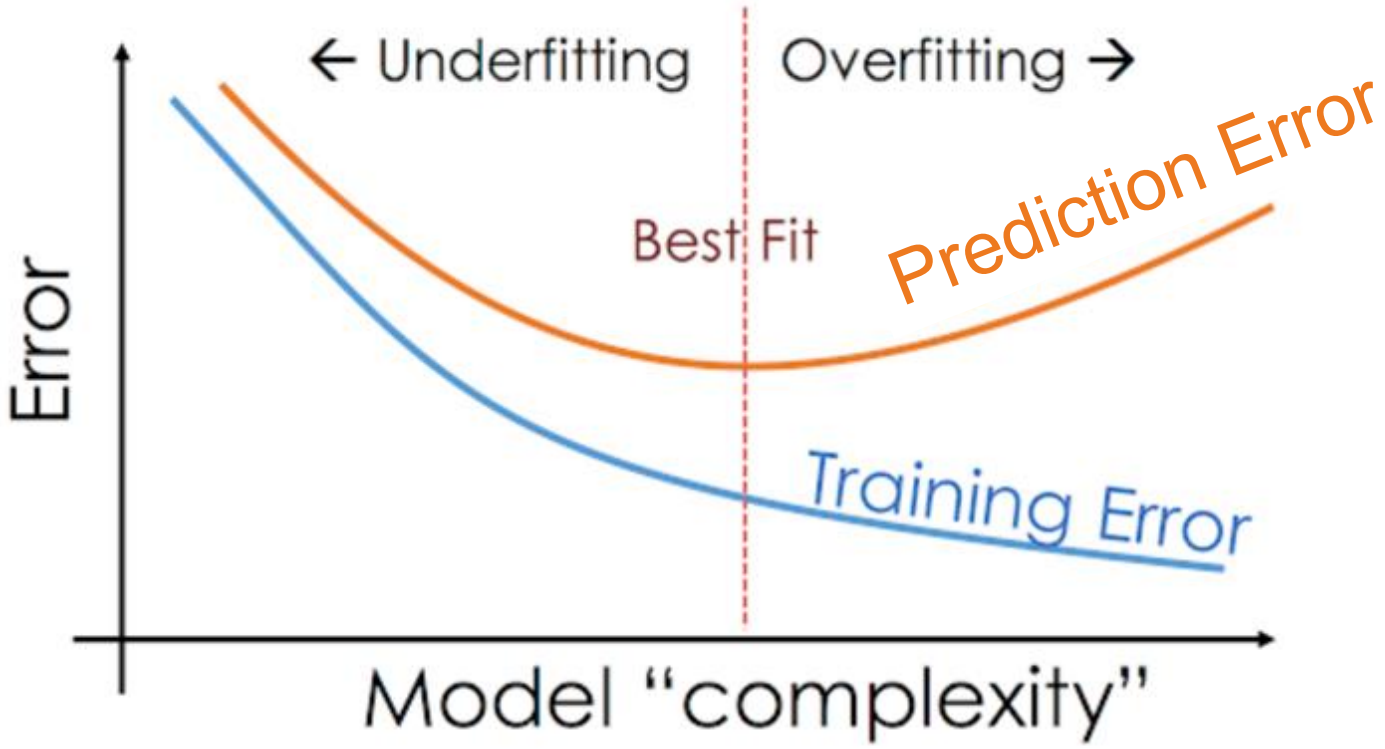
Training error: 0.07
Prediction error: 2000



Overfit

Bild-Quelle: Illustration of the underfitting/overfitting (researchgate.net)

Wahl des Modells: Modell-Komplexität



Artificial Neural Network

Lösung:

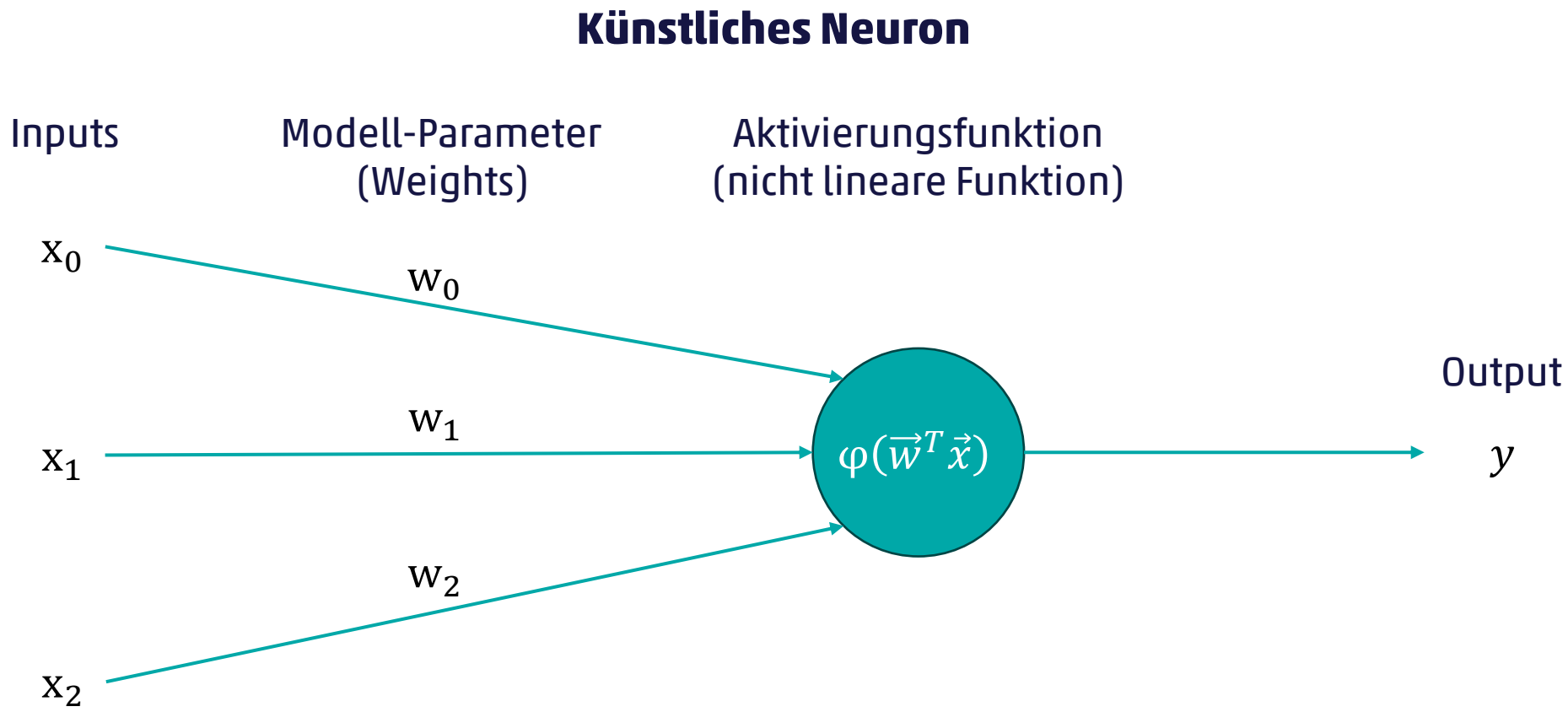
Komplexes Modell + Regulierung

Regulierung (Regularization): Techniken, um das resultierende Modell mit seinen Modell-Parametern "einfach" zu halten

Bild-Quelle: Overfitting and Underfitting in Machine Learning (analyticsvidhya.com)

Artificial Neural Network (ANN)

- ANNs sind vom menschlichen Gehirn inspirierte Modelle
- ANNs bestehen aus vielen künstlichen Neuronen (oft mehrere Millionen oder sogar Milliarden)



Artificial Neural Network (ANN)

- ANNs sind komplexe Modelle, deren Modell-Parameter man „effizient“ berechnen kann
- Viele Layers → Deep Neural Network (DNN) / Deep Learning

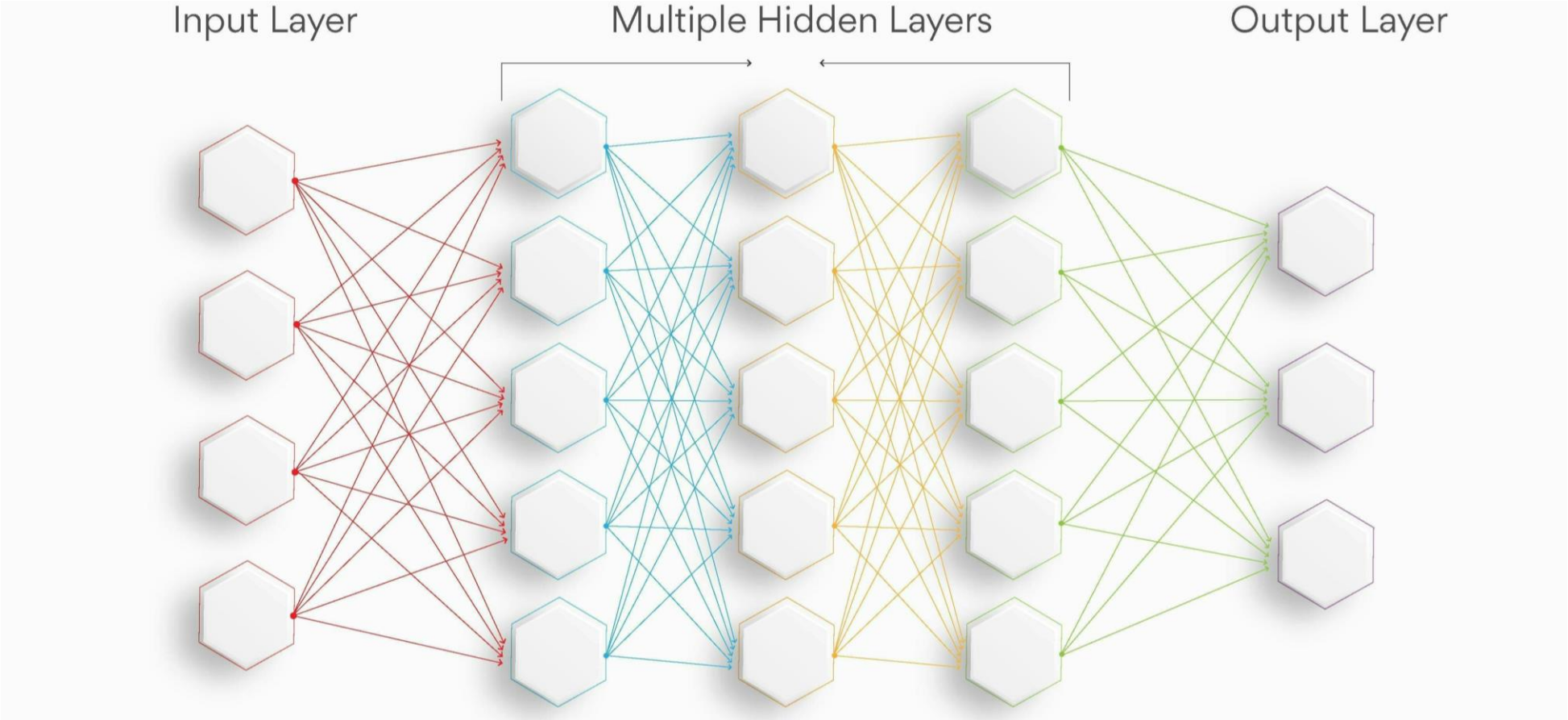
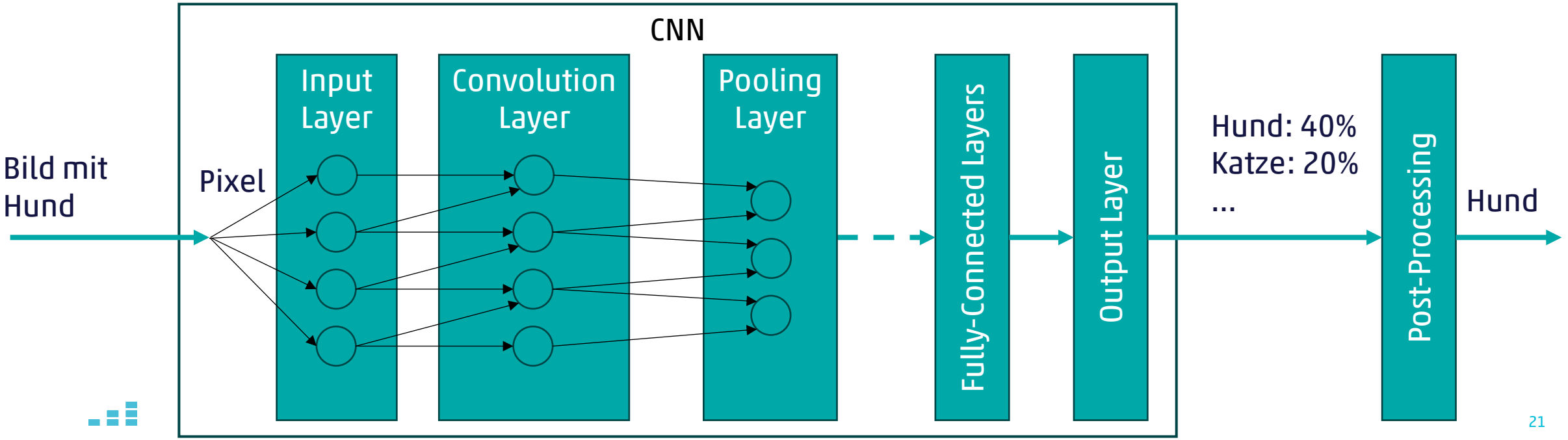


Bild-Quelle: Role of Artificial Neural Network in Artificial Intelligence (turing.com)

Convolutional Neural Network (CNN)

- CNNs sind spezialisierte ANNs
- CNNs werden häufig für die Bildverarbeitung eingesetzt (z.B. Bilderkennung und Bildklassifizierung)
- Convolution-Layer: Input nur von vorherigen „benachbarten“ Neuronen (Reduktion der #Modell-Parameter)
- Pooling-Layer: Zusammenfassen von vorherigen „benachbarten“ Neuronen (Reduktion der Breite des ANN)

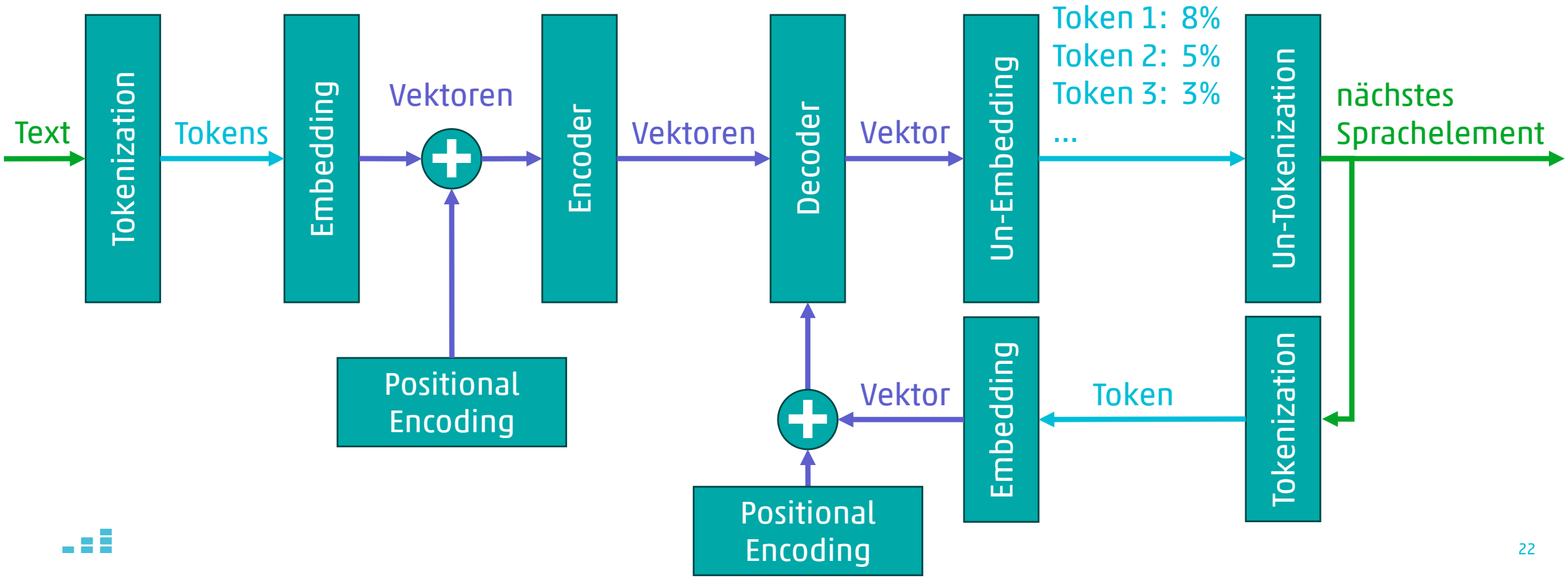
Beispiel: Bildklassifizierung



Large Language Models (LLM)

- LLMs sind „grosse“ bzw. komplexe Modelle, die Text generieren oder verarbeiten
- Die besten LLMs basieren auf ANNs (mit Transformer-Architektur)

Beispiel: Textgenerierung



Kurzfassung

- Was ist eigentlich KI?
  - Technologien, welche die menschliche Intelligenz nachahmen
- Was ist der Unterschied zwischen KI und ML?
  - ML ist eine Technologie, mit der man KI implementieren kann
- Wie funktioniert ML?
  - man wählt ein Modell und berechnet mit einem Trainingsalgorithmus die Modell-Parameter, sodass für die Trainingsdaten der Fehler minimal ist
- Was sind ANNs?
  - ANNs sind komplexe Modelle, deren Modell-Parameter man „effizient“ berechnen kann
  - ANNs werden in fast allen modernen Modellen verwendet

Vielen Dank für Ihre Aufmerksamkeit_

Martin Kaufmann

info@cnlab-security.ch
+41 55 214 33 40

cnlab security AG
Obere Bahnhofstrasse 32b
CH-8640 Rapperswil-Jona
Switzerland

cnlab

© 2024