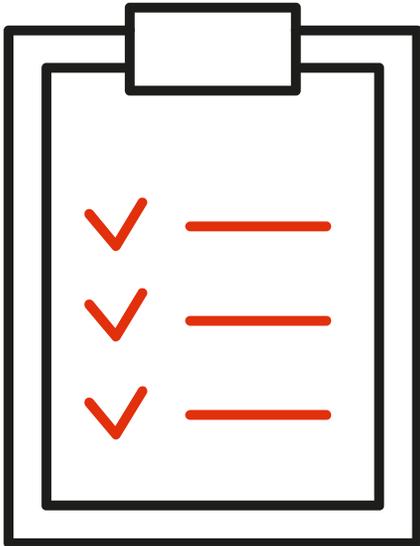


Eintauchen in Deep Learning mit künstlichen neuronalen Netzen

07.12.2022

Dr. Susanne Franke und Martin Folz, M. Sc.
d-opt GmbH und Mittelstand-Digitalzentrum Chemnitz

Agenda



- **Vorstellung und Begrüßung**
- KI in der Praxis: Bildgestützte Qualitätskontrolle
- Künstliche Intelligenz, Maschinelles Lernen und Deep Learning
- Einführung KI
- Neuronale Netze
 - Grundlegende Funktionsweise, Aktivierungsfunktion
 - Wie lernt ein Neuronales Netz?
 - Optimizer- und Lossfunktion
 - Netzarchitekturen
- Hyperparameter

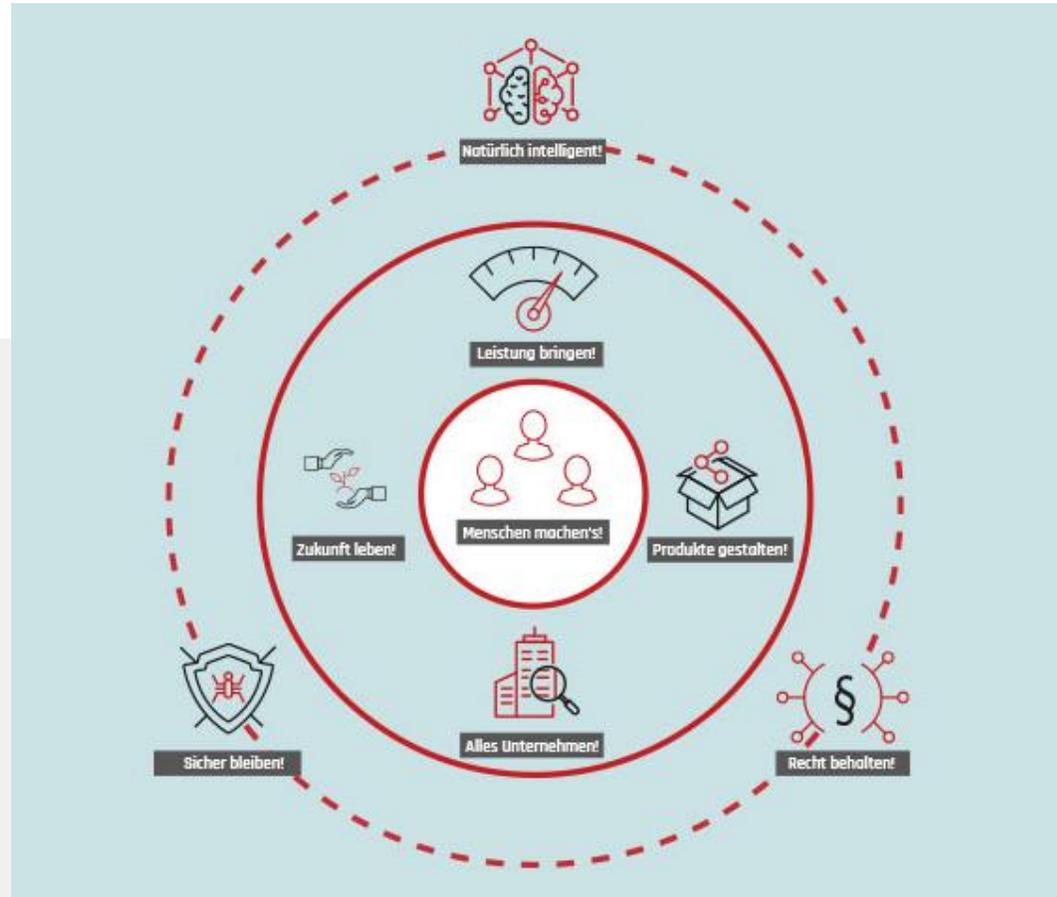
Unser Ziel: Digitalisierung unterstützen

Sächsischer Mittelstand in Industrie, Handwerk und Handel als Zielgruppe

- **Kostenfreier und anbieterneutraler Wissens- und Technologietransfer:**
 - Veranstaltungen wie Workshops, Seminare und Expertenrunden anbieten
 - Fachwissen zugänglich machen
 - Digitalisierungsprojekte begleiten
 - Trainings- und Testumgebungen zur Verfügung stellen
 - Lösungen mit Hilfe von Demonstratoren veranschaulichen
 - Erfahrungsaustausch zwischen Unternehmen fördern

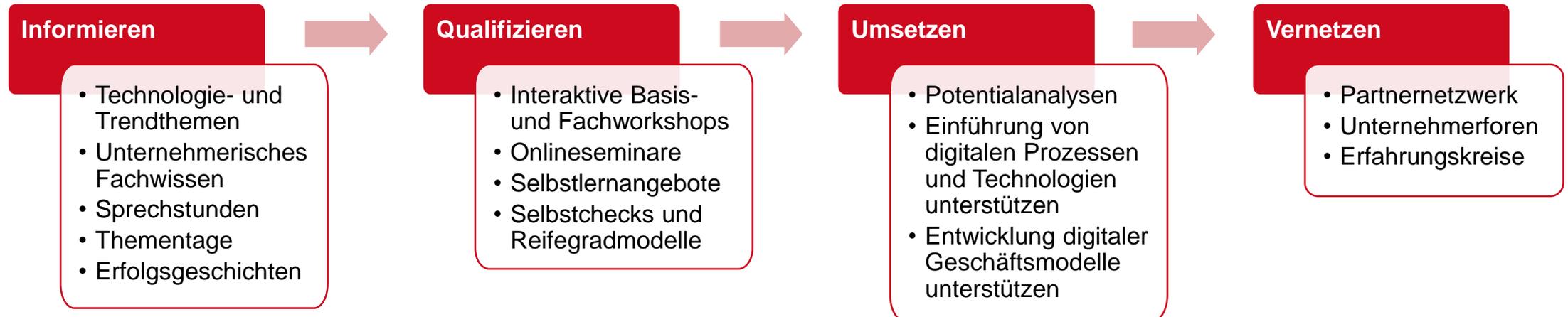
Unsere Themen

Im Fokus steht der Mensch



Praxisnahe Unterstützung

Angebote für Einsteiger in die Digitalisierung und Erfahrene



Digitalisierungsprojekte begleiten

Individuelle Projekte im Unternehmen

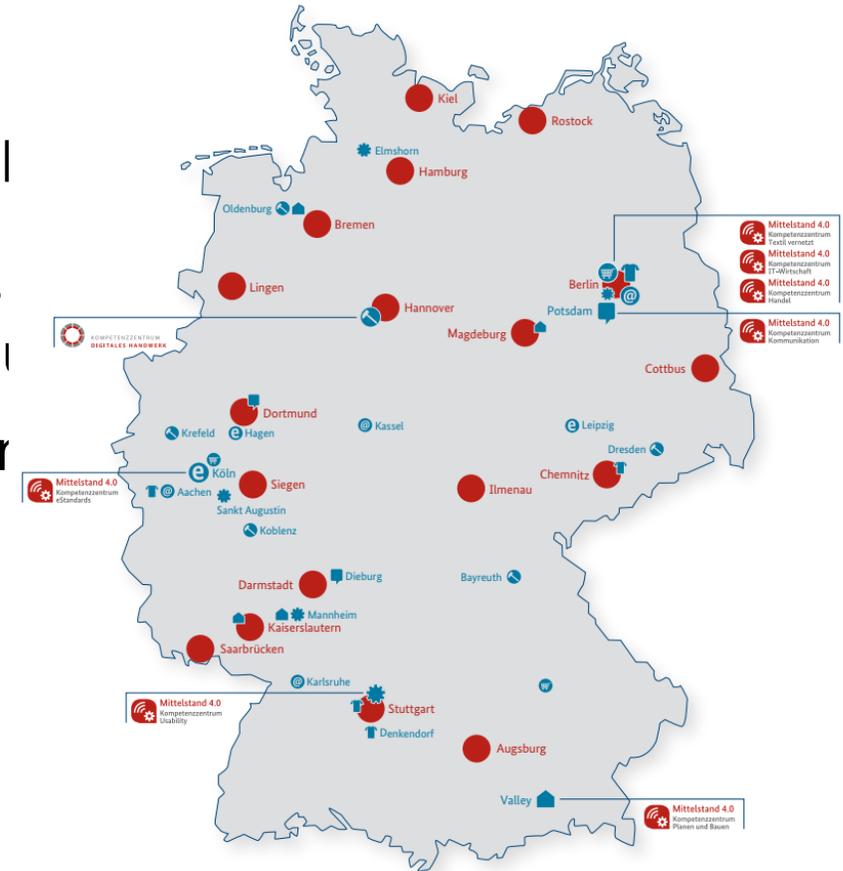
- Bis zu 5 Monate Projektdauer
- Ist-Zustand gemeinsam analysieren
- Lösungskonzept (und Prototyp) gemeinsam entwickeln
- Ergebnisse dokumentieren / Lastenheft erstellen
- Öffentliche Berichterstattung



Das Mittelstand-Digital Netzwerk

Regionale Zentren und Zentren mit Themenschwerpunkt

- Mit dem Mittelstand-Digital Netzwerk unterstützt das Wirtschaft und Energie die Digitalisierung in kleinen und mittleren Unternehmen
- Das Zentrum in Chemnitz fokussiert die Digitalisierung in der Produktion
- Bundesweite Vernetzung



Unsere Experten vor Ort

Starkes Partnernetzwerk



KI-Hub Sachsen Thüringen

Künstliche Intelligenz für KMU in Mitteldeutschland

Erfahrene KI-Trainer vermitteln Grundlagen und machen neueste Erkenntnisse der KI-Forschung greifbar.

Die Angebote sind ideal, um

- zu lernen und verstehen, was KI ist und wie sie funktioniert
- Potentiale auszuloten
- erste eigene Projekte zu starten



d-opt GmbH: Entwicklungspartner für industrielle Datenanalyse



KI- und Prozessberatung

- Prozessanalyse
- Potenzialbewertung
- Auswahl und Einsatz geeigneter KI-Methoden
- Prozessoptimierung



Datenanalyse Softwareentwicklung

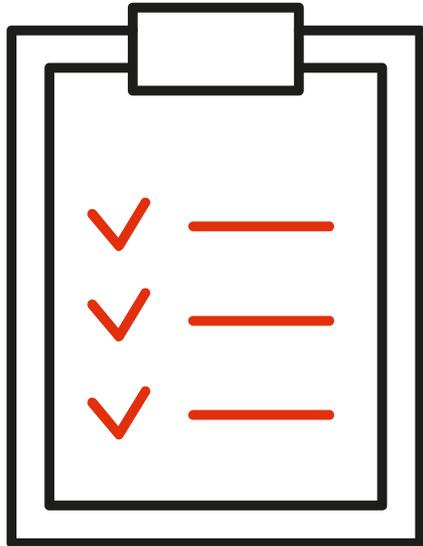
- Auftragsanalyse für individuelle Problemstellungen
- Entwicklung und Einführung von KI-Software



Aus- und Weiterbildung

- Softwareschulungen
- Seminare und Workshops für Mitarbeitende
- wissenschaftliche Lehraufträge

Agenda



- Vorstellung und Begrüßung
- **KI in der Praxis: Bildgestützte Qualitätskontrolle**
- Künstliche Intelligenz, Maschinelles Lernen und Deep Learning
- Einführung KI
- Neuronale Netze
 - Grundlegende Funktionsweise, Aktivierungsfunktion
 - Wie lernt ein Neuronales Netz?
 - Optimizer- und Lossfunktion
 - Netzarchitekturen
- Hyperparameter

KI in der Praxis: Bildgestützte Qualitätskontrolle

Problemstellung:

- Mikroelektronikindustrie: Fertigung von Platinen
- erhöhte Ausschussraten bei Lötverbindungen auf Platinen
- automatische optische Inspektion trifft Vorauswahl, aber: zu hohe Fehlerquote
→ umständliche Kontrolle und Nacharbeit durch Mitarbeitende erforderlich

KI in der Praxis: Bildgestützte Qualitätskontrolle

Datenbasis:

- hochauflösende Bildaufnahmen der Platinen
- Einschätzungen der Mitarbeitenden zu Fehlerstellen

KI in der Praxis: Bildgestützte Qualitätskontrolle

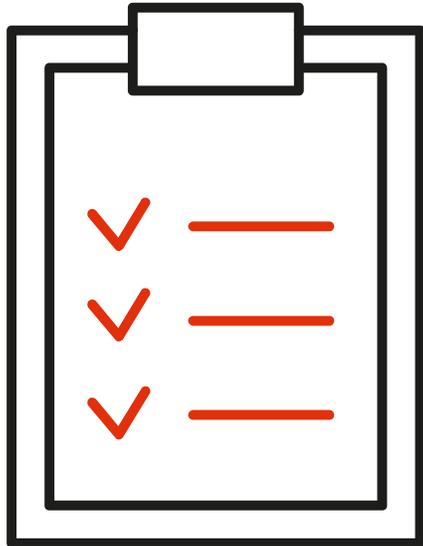
Lösung:

- Automatisierung der optischen Prüfprozesse mittels KI-basierter Bildverarbeitung
- Assistenzsystem zur KI-basierten Fehlerstelleneinschätzung

- ✓ Zeitersparnis
- ✓ höhere Zuverlässigkeit
- ✓ Entlastung der Mitarbeitenden

→ **erhöhte Wertschöpfung**

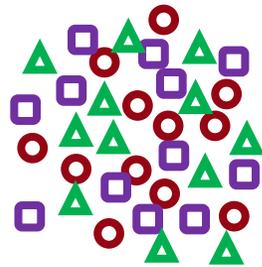
Agenda



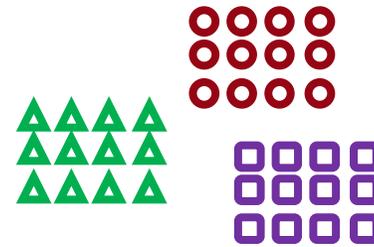
- Vorstellung und Begrüßung
- KI in der Praxis: Bildgestützte Qualitätskontrolle
- **Künstliche Intelligenz, Maschinelles Lernen und Deep Learning**
- Einführung KI
- Neuronale Netze
 - Grundlegende Funktionsweise, Aktivierungsfunktion
 - Wie lernt ein Neuronales Netz?
 - Optimizer- und Lossfunktion
 - Netzarchitekturen
- Hyperparameter

Künstliche Intelligenz, Maschinelles Lernen und Deep Learning

Eingangsdaten

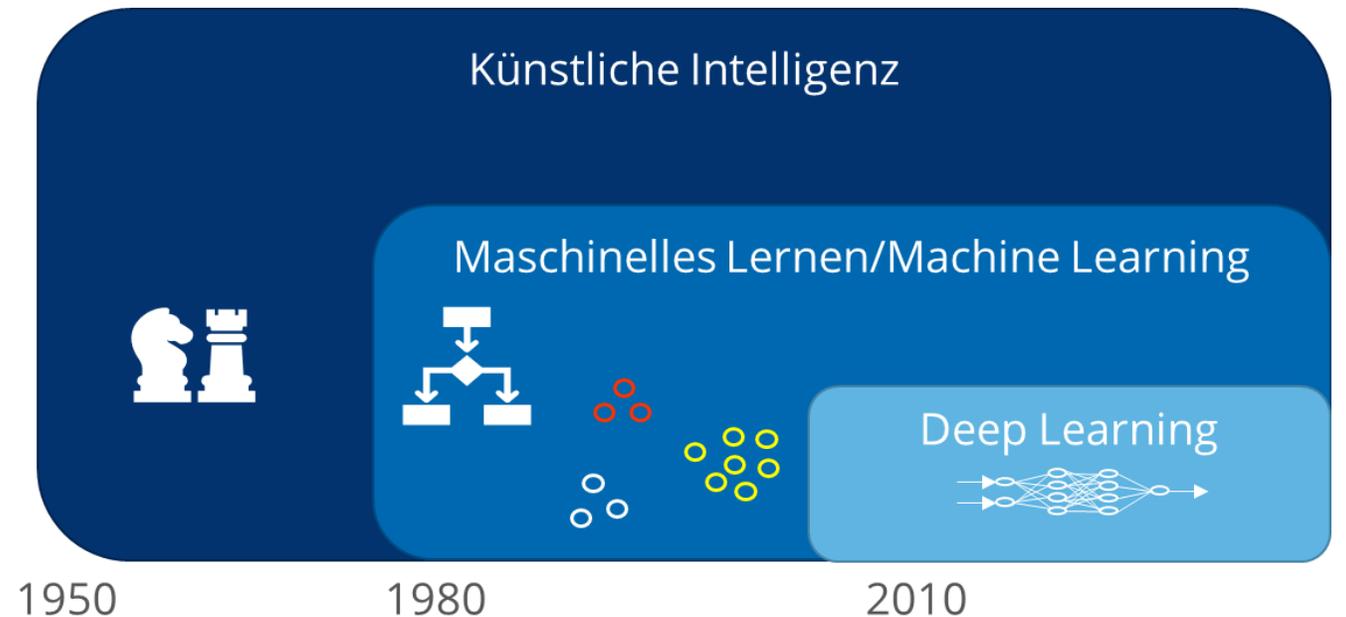


Ausgangsdaten

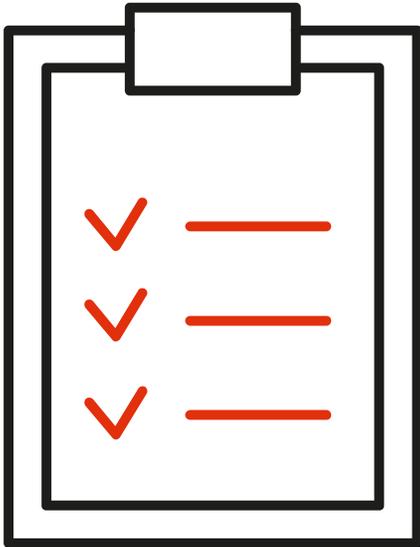


Künstliche Intelligenz, Maschinelles Lernen und Deep Learning

- Wissen generieren aus Daten
- Algorithmen, die automatisch Zusammenhänge erkennen und lernen
- automatische Verbesserung der Leistung durch neu gewonnene Erfahrungen



Agenda



- Vorstellung und Begrüßung
- KI in der Praxis: Bildgestützte Qualitätskontrolle
- Künstliche Intelligenz, Maschinelles Lernen und Deep Learning
- **Einführung KI**
- Neuronale Netze
 - Grundlegende Funktionsweise, Aktivierungsfunktion
 - Wie lernt ein Neuronales Netz?
 - Optimizer- und Lossfunktion
 - Netzarchitekturen
- Hyperparameter

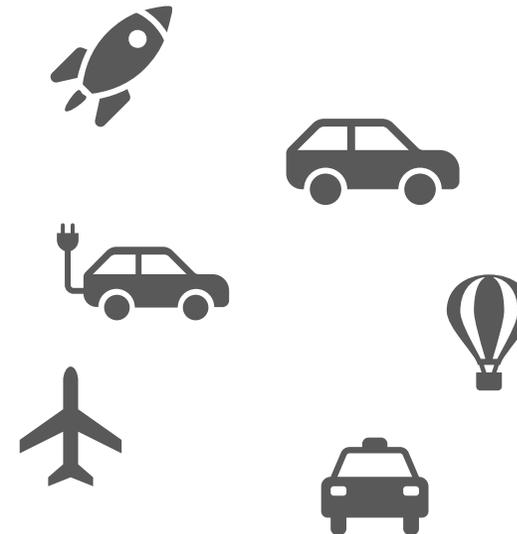
Lernparadigmen

Ziel:

Mustererkennung in vorhandenen
Daten

Unterschied:

Sind Informationen über das Ziel
bekannt?



Lernparadigmen

Ziel:

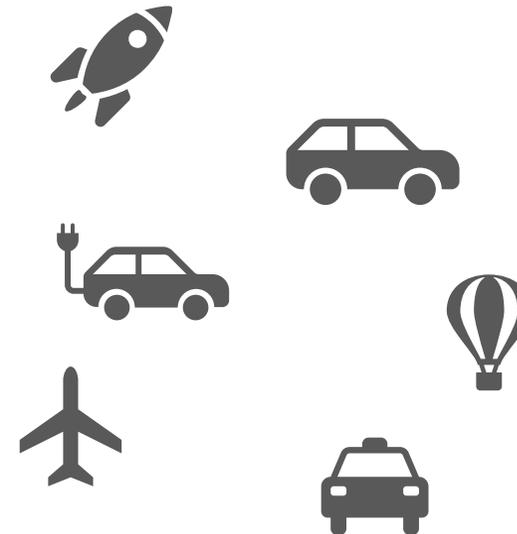
Mustererkennung in vorhandenen
Daten

Unterschied:

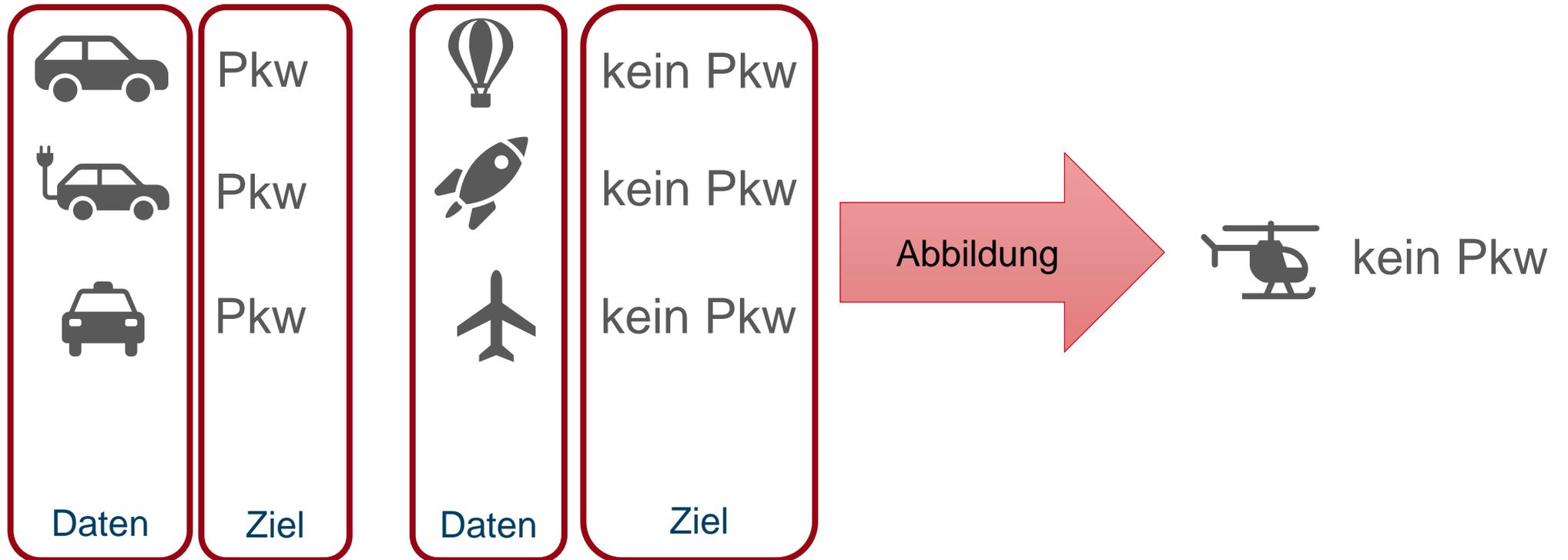
Sind Informationen über das Ziel
bekannt?

→ **JA: Überwachtes Lernen**

→ **NEIN: Unüberwachtes Lernen**



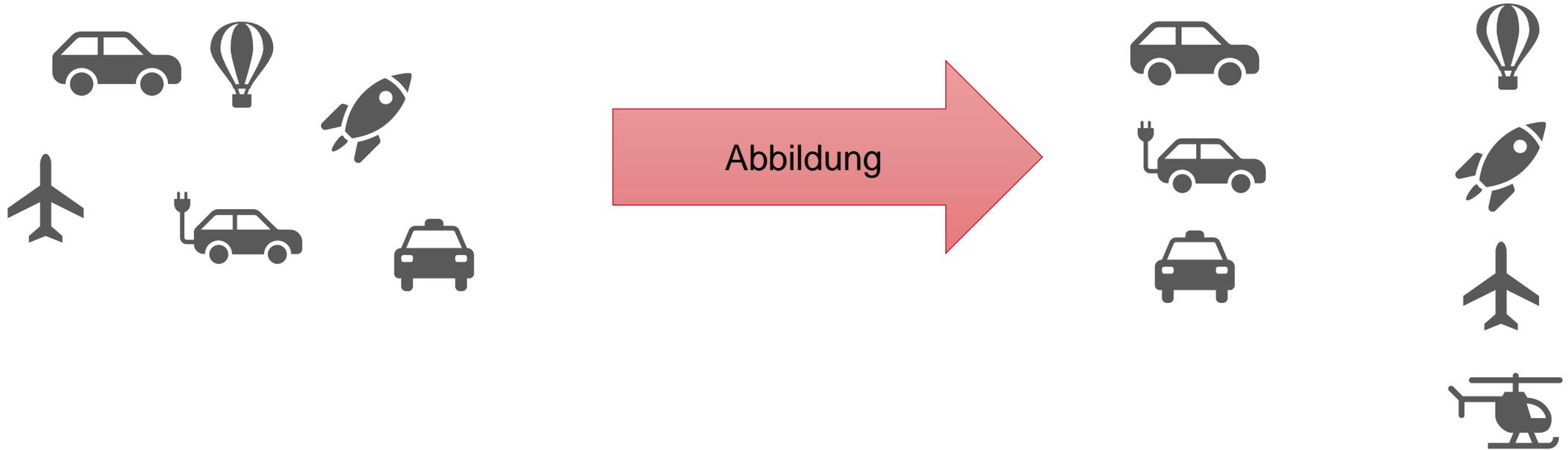
Überwachtes vs. Unüberwachtes Lernen



Überwachtes vs. Unüberwachtes Lernen



Überwachtes vs. Unüberwachtes Lernen

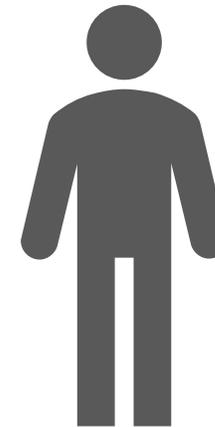


Exkurs: Bestärkendes Lernen

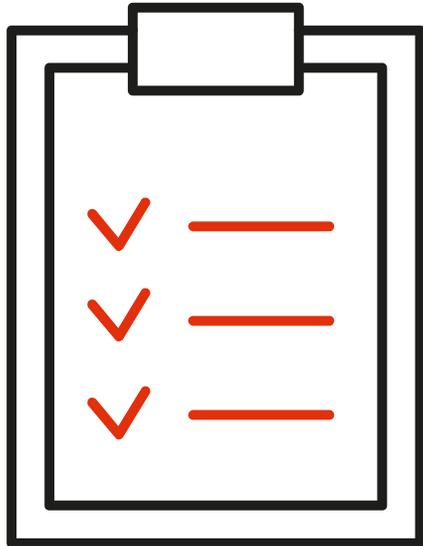
„Agent“
→ handelt



Umgebung
→ überwacht
→ belohnt konkrete Handlungen



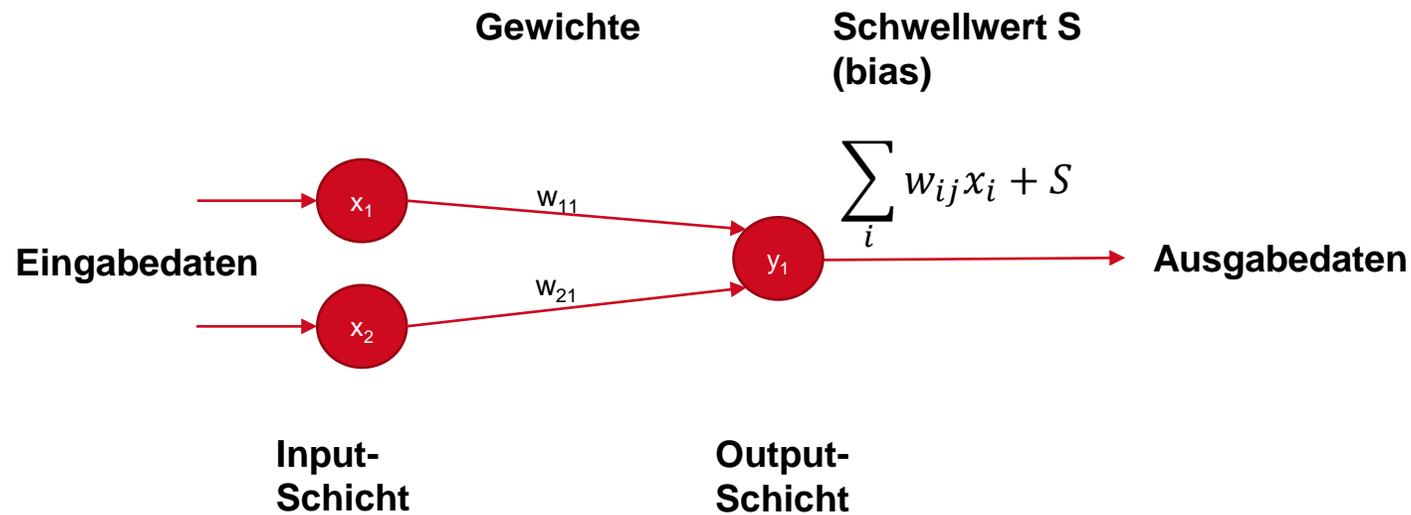
Agenda



- Vorstellung und Begrüßung
- KI in der Praxis: Bildgestützte Qualitätskontrolle
- Künstliche Intelligenz, Maschinelles Lernen und Deep Learning
- Einführung KI
- **Neuronale Netze**
 - Grundlegende Funktionsweise, Aktivierungsfunktion
 - Wie lernt ein Neuronales Netz?
 - Optimizer- und Lossfunktion
 - Netzarchitekturen
- Hyperparameter

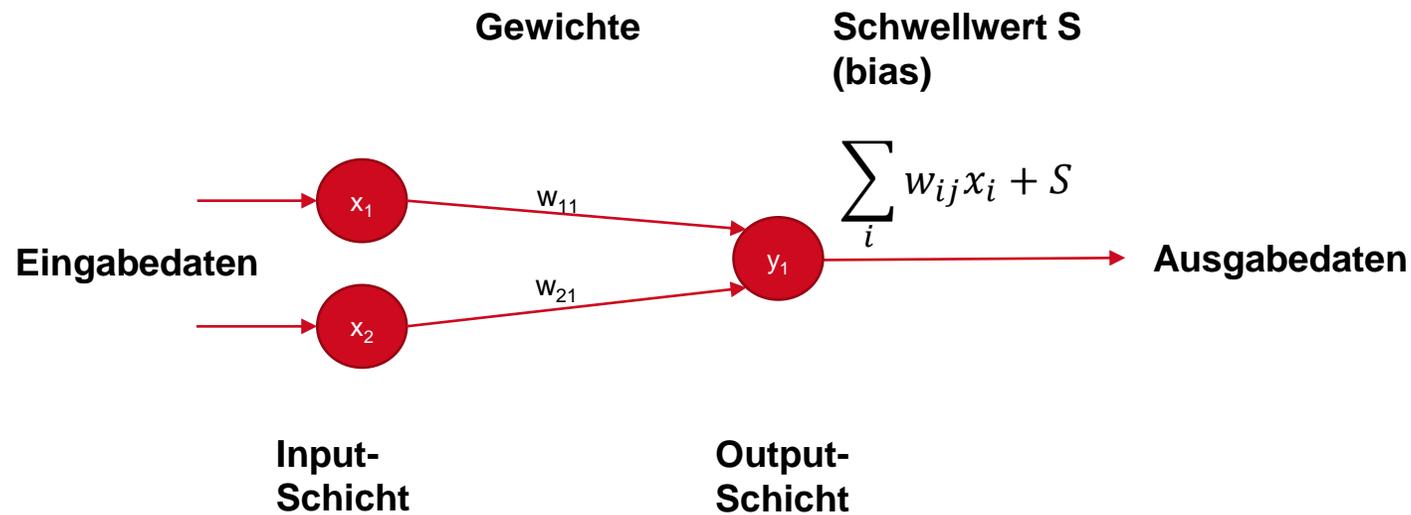
Neuronale Netze: Grundlegende Funktionsweise

Einschichtiges/einfaches Perzeptron:

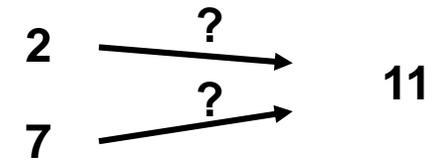


Neuronale Netze: Grundlegende Funktionsweise

Einschichtiges/einfaches Perzeptron:



Beispiel:

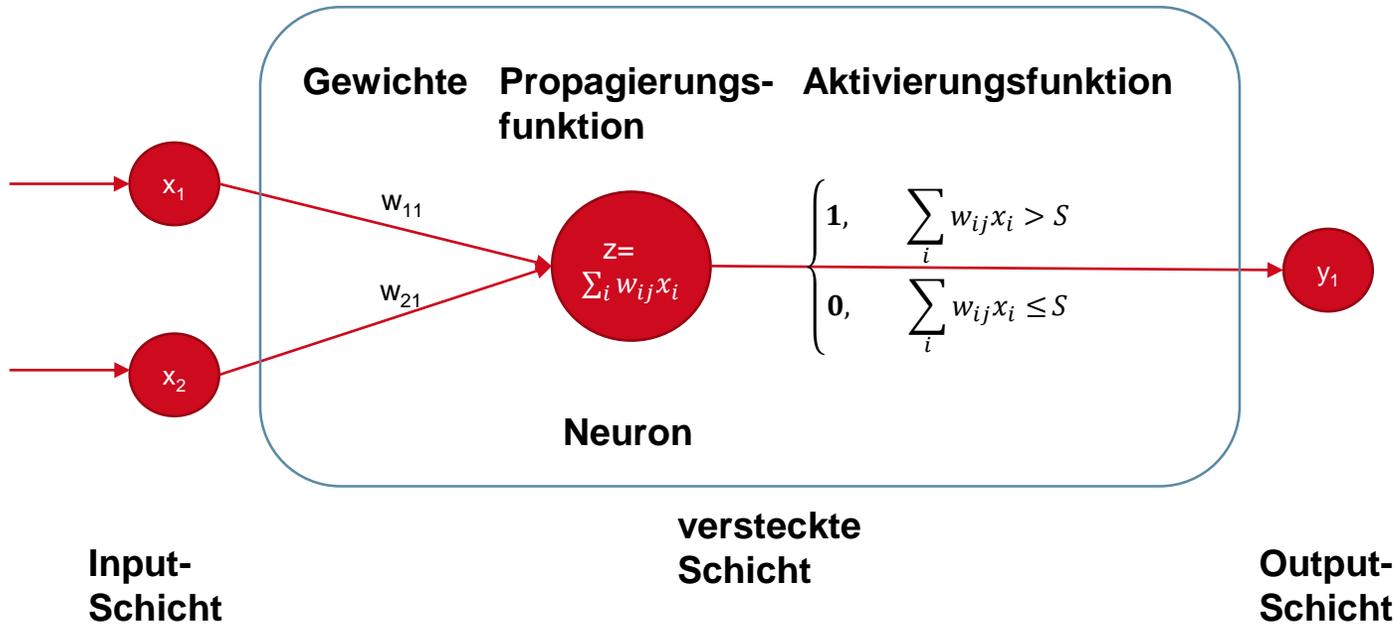


Nützliche Bibliotheken in Python

Name der Bibliothek	Funktion
numpy	enthält numerische Berechnungen und mathematische Routinen z. B. Wurzelziehen, Erzeugen randomisierter Werte
matplotlib	ermöglicht das einfache Erstellen von Grafiken
pandas	ermöglicht die einfache Arbeit mit großen Datenmengen (Verwaltung und Analyse)
sklearn, tensorflow, torch	umfasst Algorithmen des Maschinellen Lernens
torchvision	stellt Datensätze zur Auswertung bereit
os	dient der Kommunikation mit dem Betriebssystem
PIL	dient der Bearbeitung, Speicherung etc. von Bilddateien

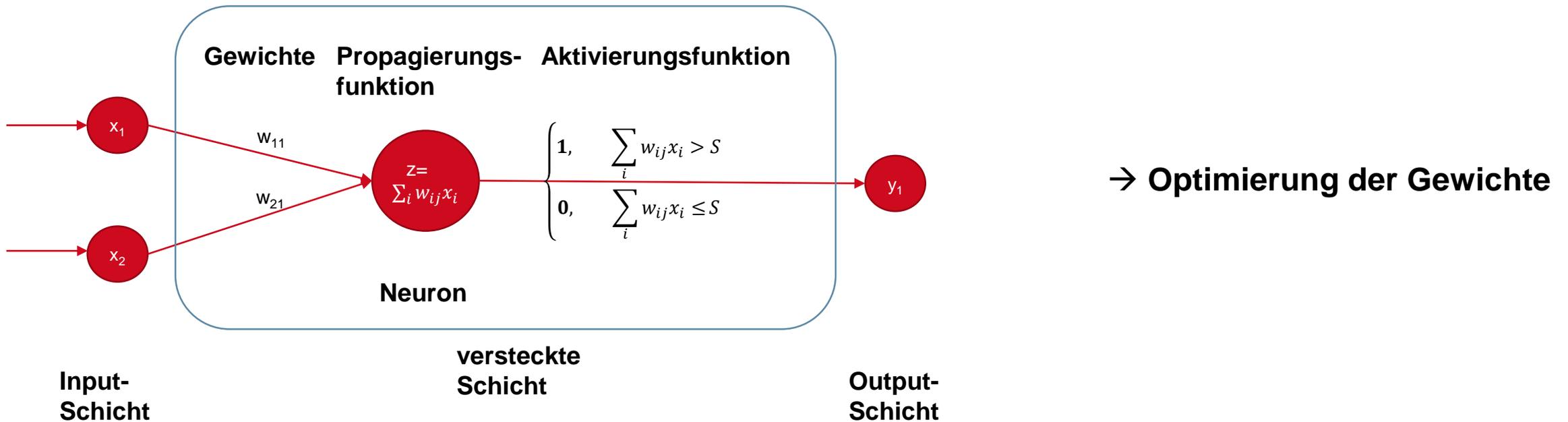
Neuronale Netze: Grundlegende Funktionsweise

Aufbau:



Neuronale Netze: Grundlegende Funktionsweise

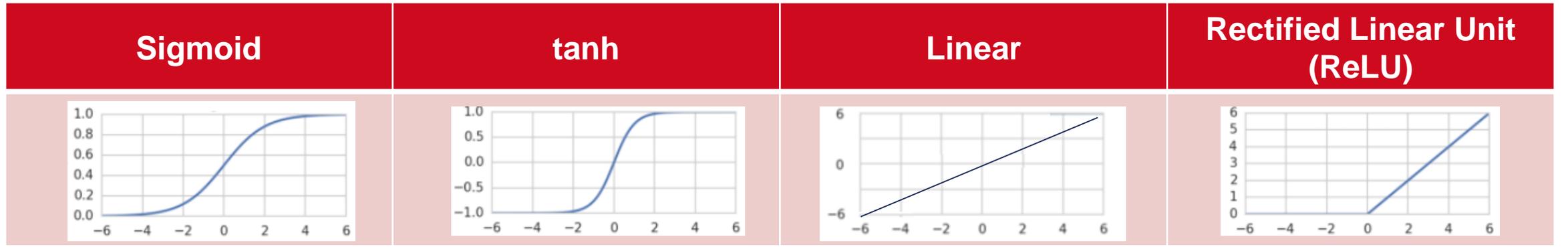
Aufbau:



Neuronale Netze: Aktivierungsfunktion

Auswertung der berechneten gewichteten Summe

- Wann „feuert“ das Neuron?
- Bestimmung des Ausgabewertes



Trainieren und Testen

DATEN

Trainings- und
Validierungsdaten

- Entwicklung der Abbildung
- Trainieren des Algorithmus

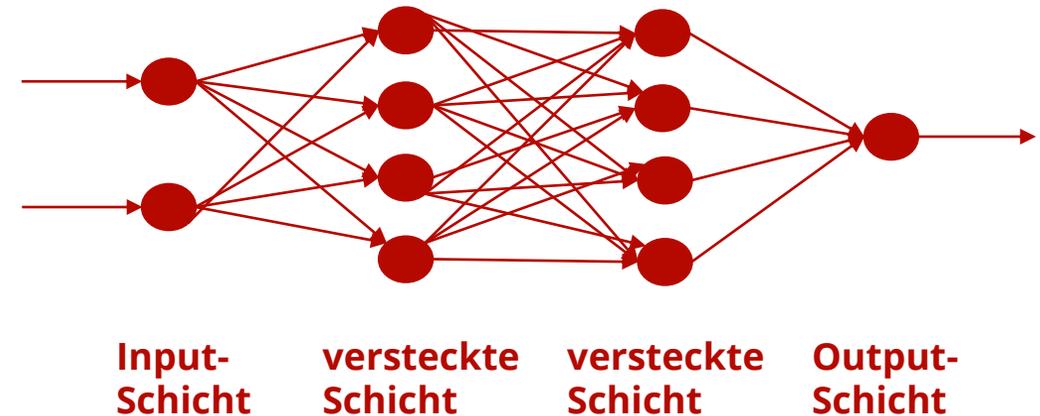
Testdaten

- Testen des Algorithmus

Neuronale Netze: Grundlegende Funktionsweise

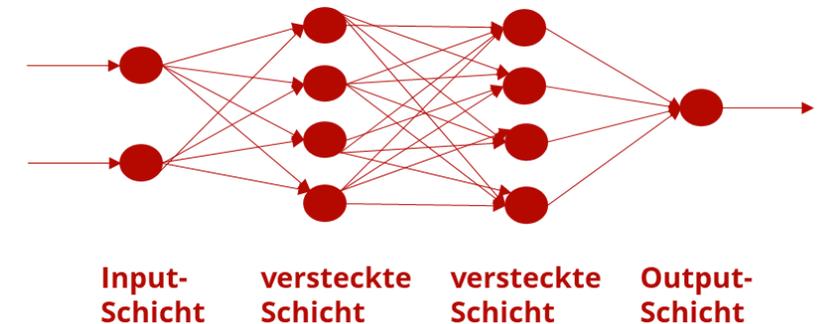
Einfaches Neuronales Netz:

- Verknüpfung der Neuronen zu einem Netz
- gerichteter, bewerteter Graph
- schichtartig aufgebaut (Schicht, „layer“)



Neuronale Netze: Wie lernt ein Neuronales Netz?

- Ziel: für die Eingabe soll das Netz eine gewünschte Ausgabe erzeugen
 - das Netz lernt, indem es sich selbst gemäß einer Lernregel (Veränderung der Gewichte) modifiziert
- Fehler zwischen berechneter und gewünschter Ausgabe soll minimiert werden



Neuronale Netze: Wie lernt ein Neuronales Netz?

Trainieren des Neuronales Netzes:

- Eingabedaten aufnehmen
- Ausgabe berechnen
- berechnete Ausgabe mit erwarteter Ausgabe vergleichen → Fehler ermitteln
- Gewichte anpassen, um den Fehler zu minimieren

Neuronale Netze: Wie lernt ein Neuronales Netz?

Trainieren des Neuronales Netzes:

- Eingabedaten aufnehmen
- Ausgabe berechnen
- berechnete Ausgabe mit erwarteter Ausgabe vergleichen → Fehler ermitteln
- Gewichte anpassen, um den Fehler zu minimieren

Testen des Neuronales Netzes:

- dem Netz ihm unbekannt Daten bereitstellen, deren Ergebnis man selbst jedoch kennt

Neuronale Netze: Wie lernt ein Neuronales Netz?

Trainieren des Neuronales Netzes:

- Eingabedaten aufnehmen
- Ausgabe berechnen
- berechnete Ausgabe mit erwarteter Ausgabe vergleichen → Fehler ermitteln
- Gewichte anpassen, um den Fehler zu minimieren

Testen des Neuronales Netzes:

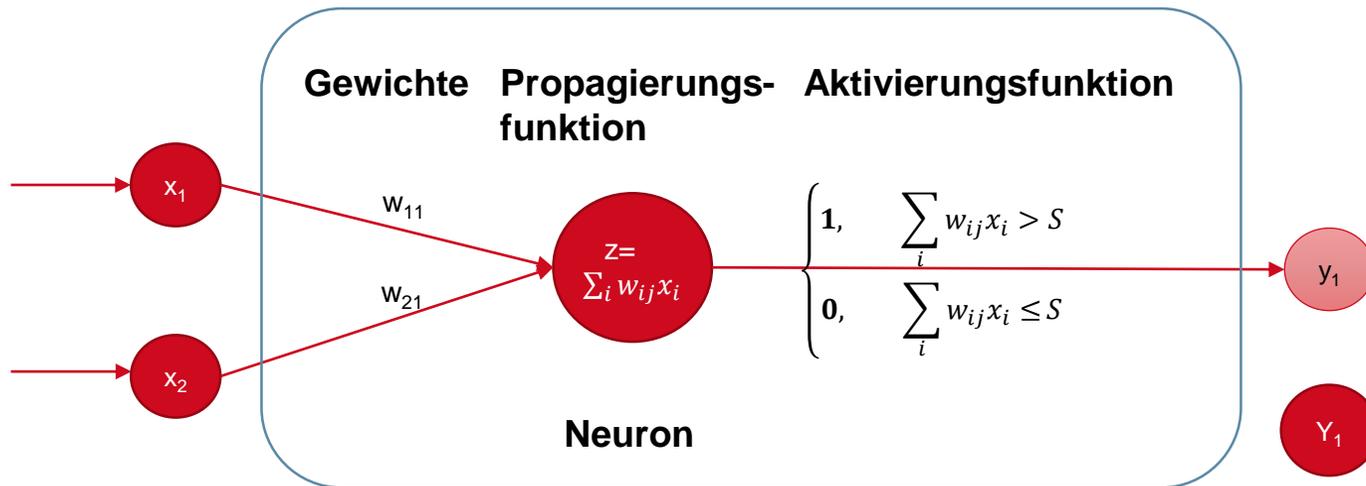
- dem Netz ihm unbekannt Daten bereitstellen, deren Ergebnis man selbst jedoch kennt

Nutzen des Neuronales Netzes:

- das trainierte Neuronale Netz berechnet für neue, dem Netz bisher unbekannte Daten eine Ausgabe

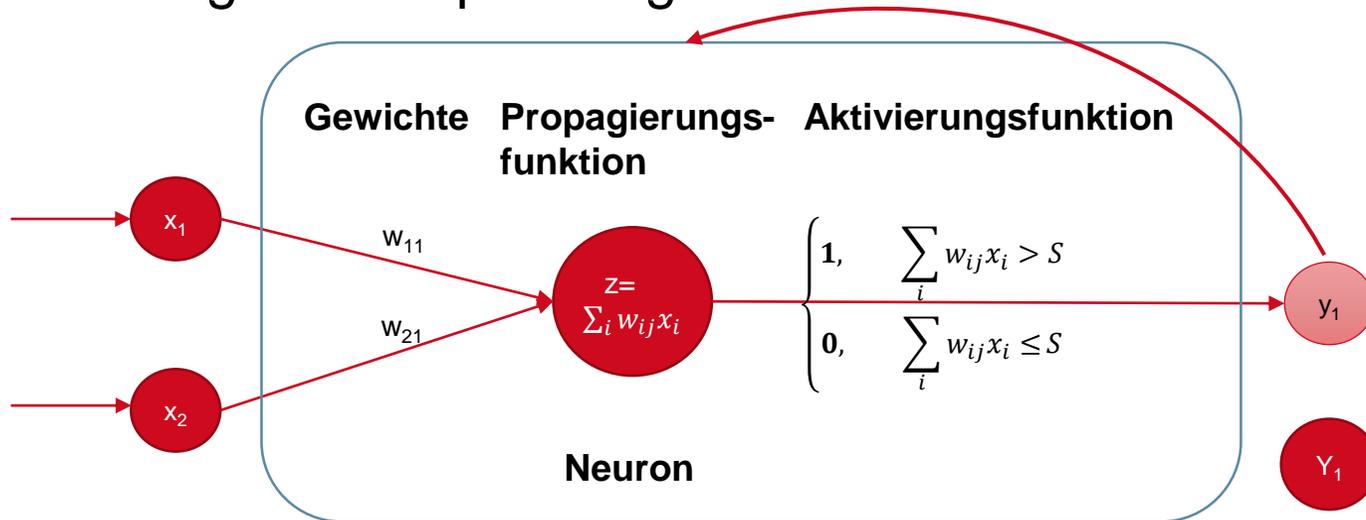
Neuronale Netze: Verlustfunktion/Lossfunktion

Fehler: Differenz zwischen berechnetem y_1 und gewünschtem Y_1 Wert



Neuronale Netze: Verlustfunktion/Lossfunktion

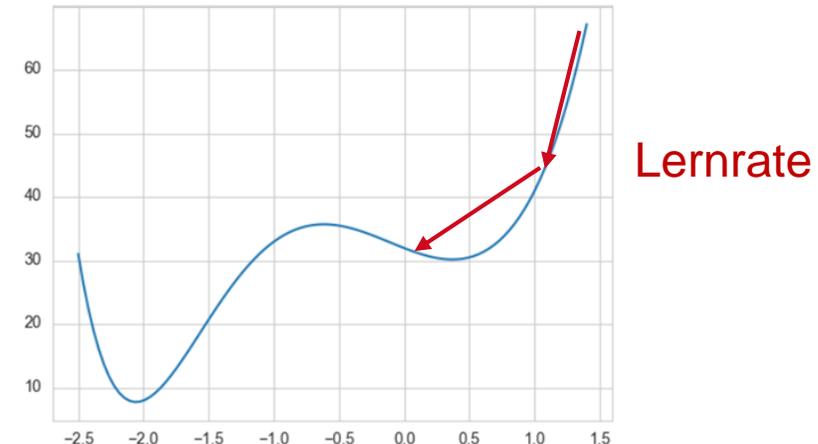
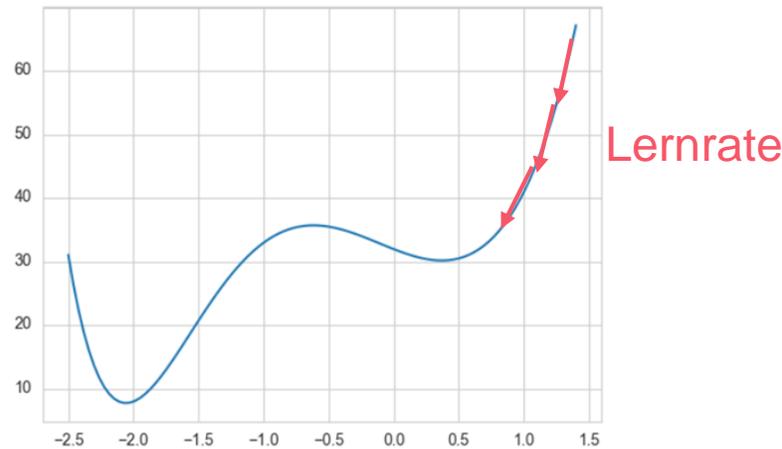
Fehler: Differenz zwischen berechnetem y_1 und gewünschtem Y_1 Wert
Änderung über Anpassung der Gewichte



Neuronale Netze: Optimizer

Ziel: schnelles Lernen, also gezielte Anpassung der Gewichte, sodass das Neuronale Netz eine möglichst gute Vorhersage ermöglicht

Fehlerwert:



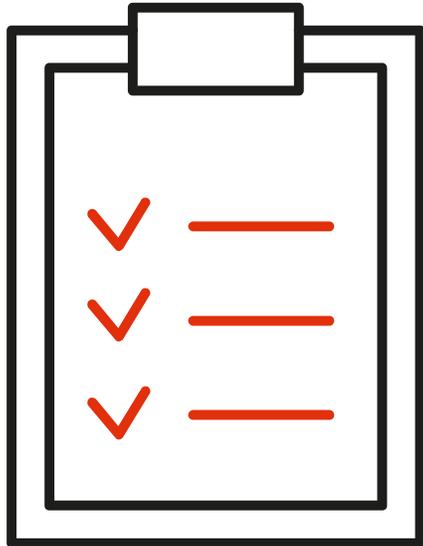
Neuronale Netze: Netzarchitekturen

	convolutional neural net (CNN)	recurrent neural net (RNN)	long short-term memory (LSTM)
Merkmale	Convolutional-Schicht → Erkennen einzelner Merkmale Pooling-Schicht → Verdichtung Verknüpfung	Rückkopplung zwischen den Neuronen Gewinnung von zeitlich codierten Informationen	spezielle Art von RNN „Kurzzeitgedächtnis, das lange anhält“ Speicherung von Abhängigkeiten und früheren Erfahrungen
Einsatzgebiete	Bildererkennung	Bildererkennung Schrifterkennung Spracherkennung	Übersetzungen Sprachverarbeitung: Chat-Bots, sprachgesteuerte Assistenten

Ablauf beim Programmieren



Agenda



- Vorstellung und Begrüßung
- KI in der Praxis: Bildgestützte Qualitätskontrolle
- Künstliche Intelligenz, Maschinelles Lernen und Deep Learning
- Einführung KI
- Neuronale Netze
 - Grundlegende Funktionsweise, Aktivierungsfunktion
 - Wie lernt ein Neuronales Netz?
 - Optimizer- und Lossfunktion
 - Netzarchitekturen
- **Hyperparameter**

Hyperparameter

Zwei Arten von Parametern:

Parameter, die während des Trainierens angepasst/gelernt werden

→ Gewichte

Hyperparameter

Parameter, die vor dem Trainieren des Neuronalen Netzes festgelegt werden müssen

→ Aktivierungsfunktion

→ Anzahl der versteckten Schichten

Ihre Ansprechpartner

Martin Folz
KI-Trainer

 0371 69001211

 martin.folz@digitalzentrum-chemnitz.de

Dr. Susanne Franke
Geschäftsführerin d-opt GmbH

 037600 760683

 s.franke@d-opt.de

 www.d-opt.de

Quellennachweise

- Python: <https://www.python.org/static/img/python-logo@2x.png>
- CNN: https://cdn-images-1.medium.com/fit/t/1600/480/1*vkQ0hXDaQv57sALXAJquxA.jpeg
- RNN: <https://nerdthecoder.files.wordpress.com/2019/02/731df-0mrhhgabskajpbt21.png>

Vielen Dank
für Ihre Aufmerksamkeit!