LARGE LANGUAGE MODELS & CHAT-GPT

Technische Grundlagen von LLMs & Use Cases

FORUM V

Vorstellung des Referenten

Dr. Felix Happach

Senior Consultant bei Q_PERIOR im Bereich Insurance

Experte für Process Intelligence

Branchenkenntnisse

- 4 Jahre in mathematischer Forschung tätig
- > 3 Jahre im Projektgeschäft als Berater im Bereich Data Analytics & KI

Beratungsschwerpunkte

- Konzeption und Anwendung KI gestützter Systeme
- Digitalisierung und Automation von Prozessen

Sonstiges

- Gastvorlesungen u.a. an der Universität Regensburg, Fachhochschule Wedel, Technische Hochschule Ingolstadt
- Interne Trainings für Module rund um KI & BPM

Agenda

Large Language Models & Chat-GPT – Technische Grundlagen von LLMs & Use Cases

01 Technische Grundlagen

02 Chancen & Risiken

03 Use Cases

04 Vorstellung Q_PERIOR

Agenda

Large Language Models & Chat-GPT – Technische Grundlagen von LLMs & Use Cases

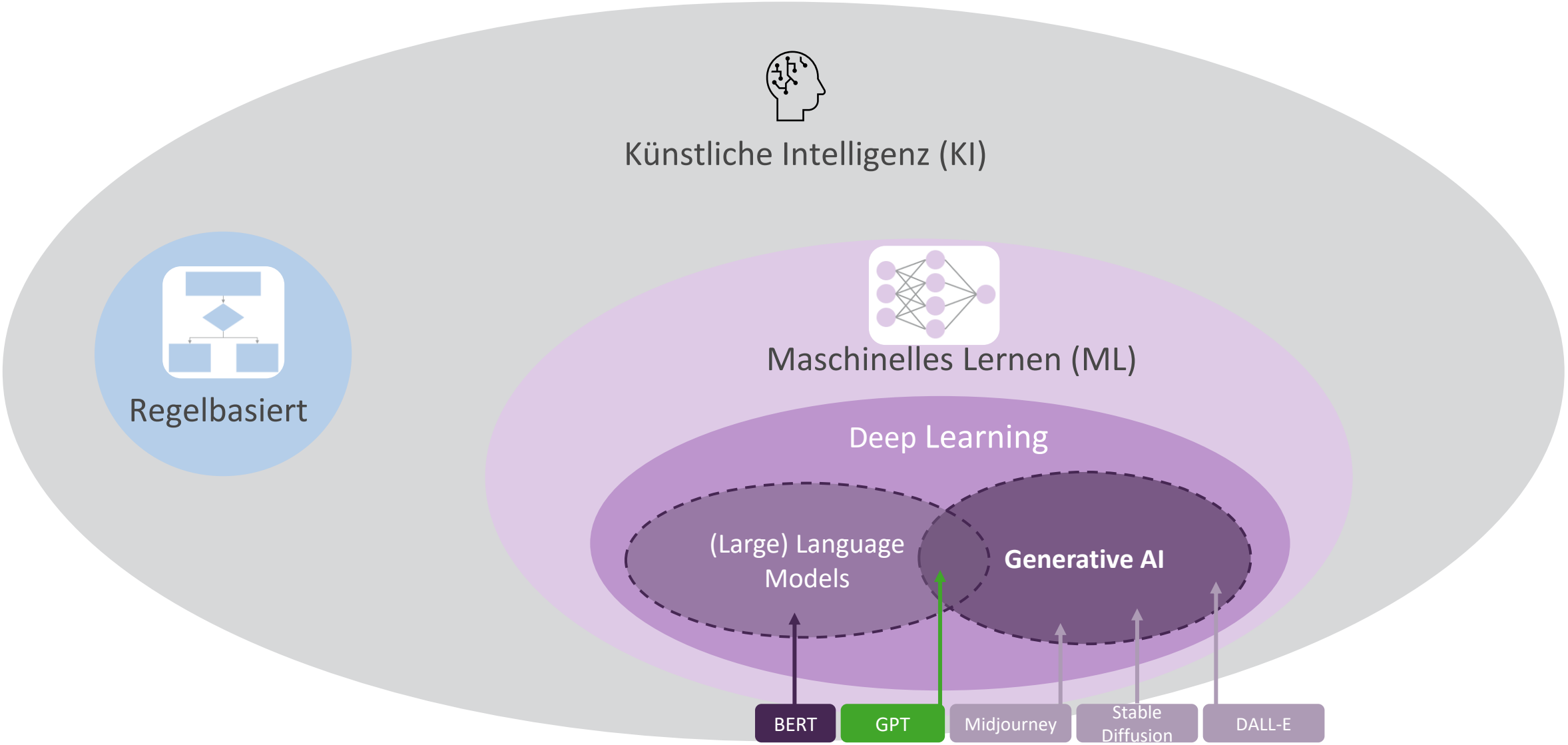
01 Technische Grundlagen

02 Chancen & Risiken

03 Use Cases

04 Vorstellung Q_PERIOR

Large Language Models und Generative AI sind spezielle Klassen des Deep Learnings



Neuronale Netze sind vom menschlichen Gehirn inspirierte ML-Algorithmen

Definition

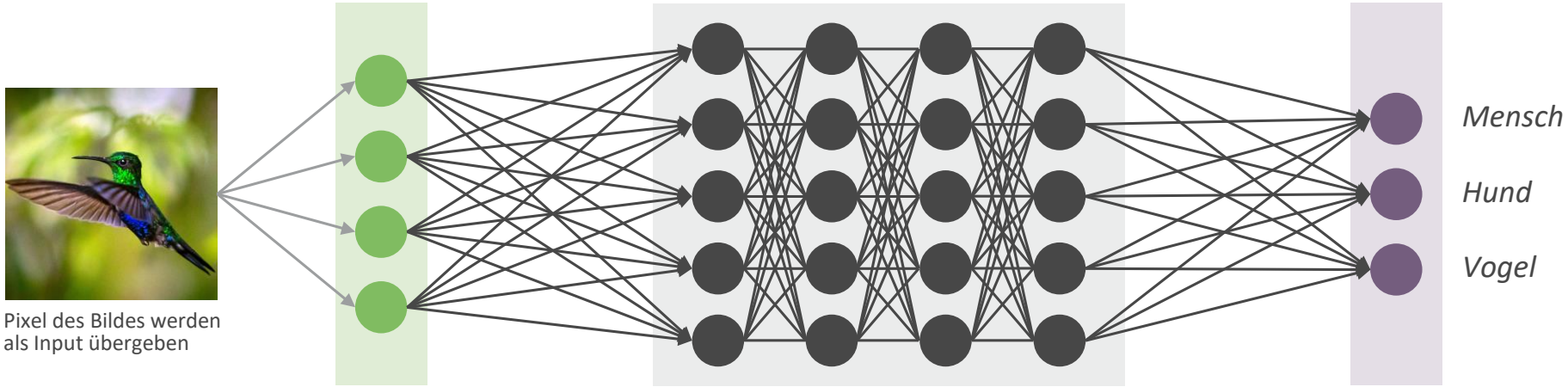
- Neuronale Netze sind Algorithmen des maschinellen Lernens, die an die Struktur und Funktion des menschlichen Gehirns angelehnt sind
- Sie bestehen aus miteinander verbundenen Knoten („Neuronen“), die in Schichten organisiert sind: Input, Hidden und Output

Aufbau von Neuronalen Netzen:

Der **Input-Layer** empfängt Daten und übergibt sie an die Hidden Layer.

Hidden Layer verarbeiten Daten durch (gewichtete) Verbindungen und Aktivierungs-funktionen und extrahieren Merkmale & Muster.

Der **Output-Layer** erzeugt das Endergebnis, z. B. eine Klassifizierung oder Vorhersage.



Neuronale Netze werden für verschiedene Aufgaben verwendet, unter anderem Bilderkennung und Verarbeitung natürlicher Sprache.

Neuronale Netze lernen durch die Anpassung der Gewichte während des Trainings

Durch die Anpassung der Gewichte wird die Differenz zwischen vorhergesagten und tatsächlichen Ergebnissen minimiert

Initialisierung

Gewichte werden mit Zufallswerten initialisiert

Forward Propagation

Inputdaten werden durch das neuronale Netz geleitet, um einen Output zu erzeugen

Fehler-berechnung

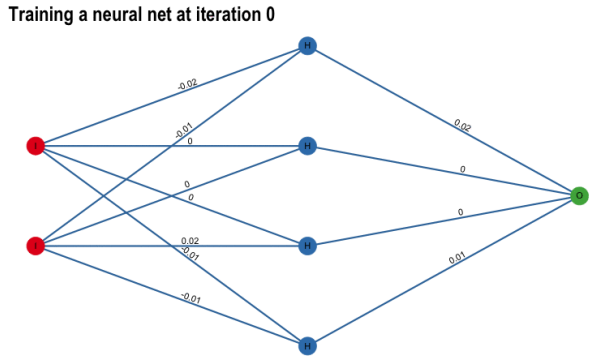
Prognostizierte und tatsächliche Ergebnisse werden mittels einer Verlustfunktion verglichen

Back-propagation

Optimierungsalgorithmen aktualisieren Gewichte, um den Verlust zu minimieren

Iteration

Parameter werden bei jeder Iteration fine-getuned, um Leistung auf Trainingsdaten zu verbessern



Validierung

In regelmäßigen Abständen wird die Leistung des Modells auf einem separaten Validierungsdatensatz ausgewertet.

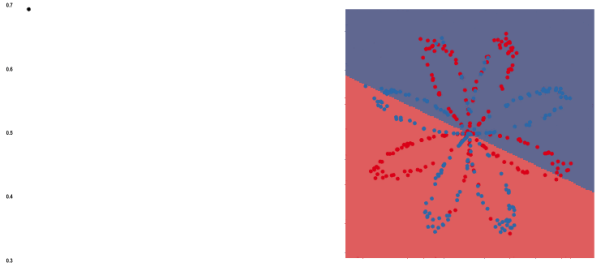
Verschlechtert sich das Modell auf dem Validierungsdatensatz, kann das Training beendet werden.

Testen

Nach Abschluss des Trainings, wird das Modell auf einem unabhängigen Testdatensatz getestet, um die Generalisierung auf neue, ungesehene Daten zu bewerten.

Deployment

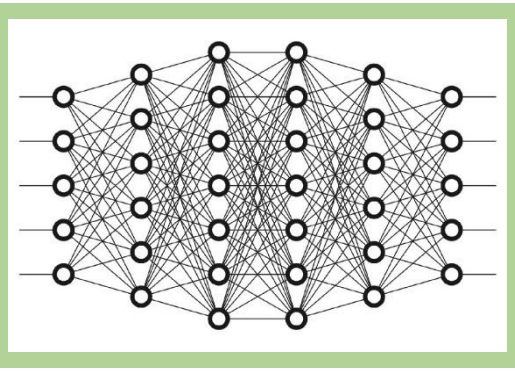
Das Modell ist bereit für die Verwendung mit realen Daten.



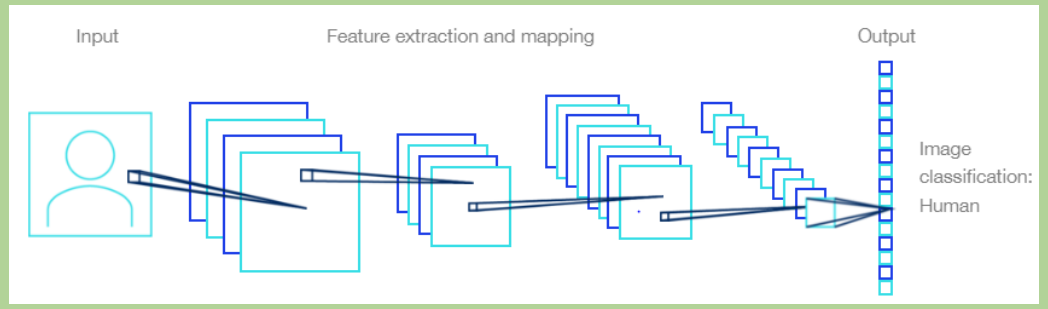
Source: http://tamaszilagyi.com/blog/2017/2017-11-11-animated_net/

Die wichtigsten Erweiterungen Neuronaler Netze sind CNNs und RNNs

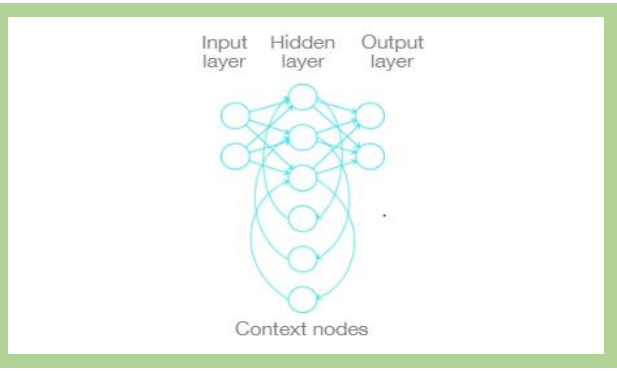
Deep Neural Network



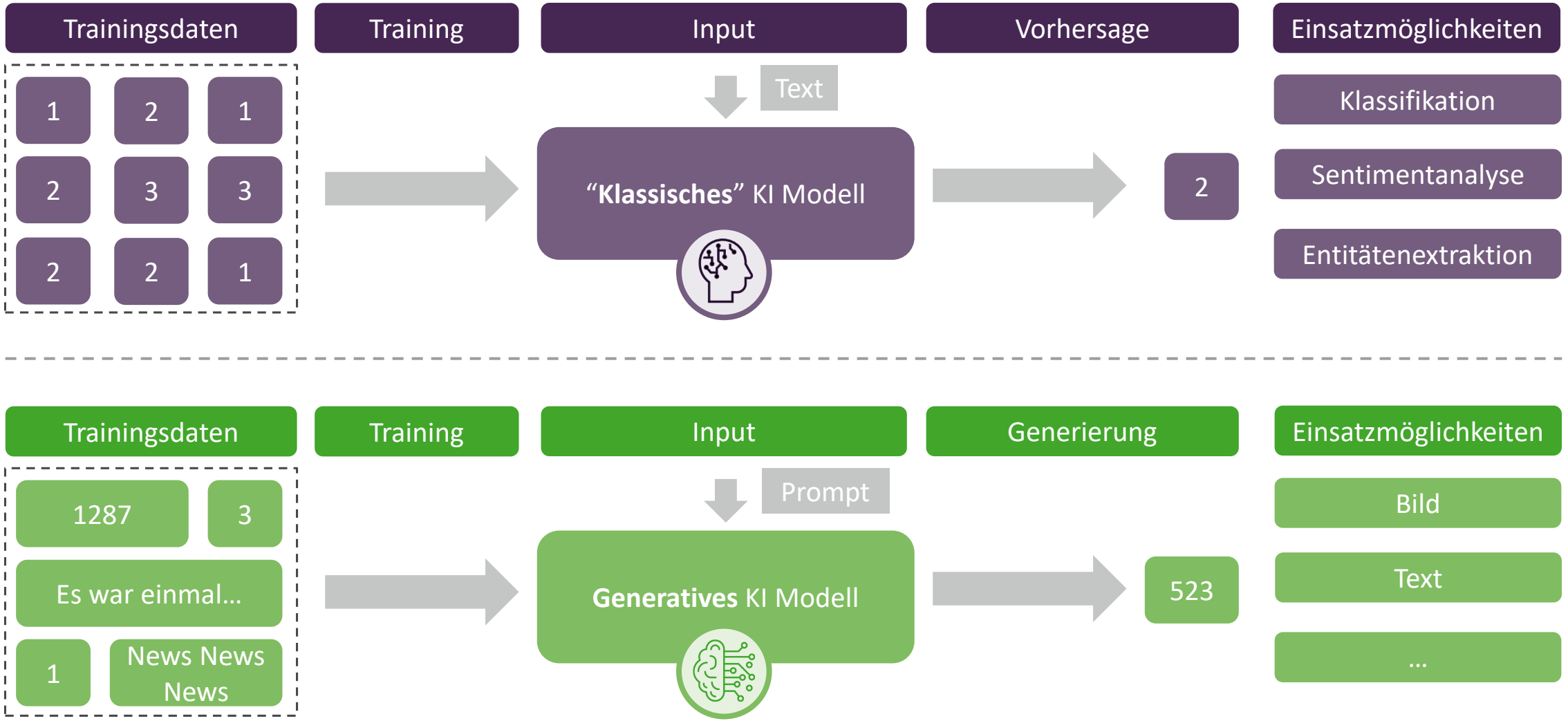
Convolutional Neural Network



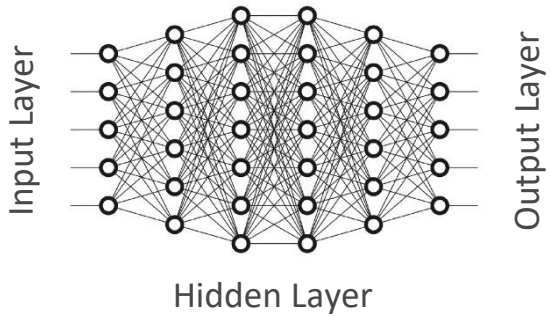
Recurrent Neural Network



Generative AI unterscheidet sich von “klassischer” KI in der Vorgehensweise



Die Basis für Large Language Models (LLMs) bilden riesige neuronale Netze



Recap: Deep Neural Networks

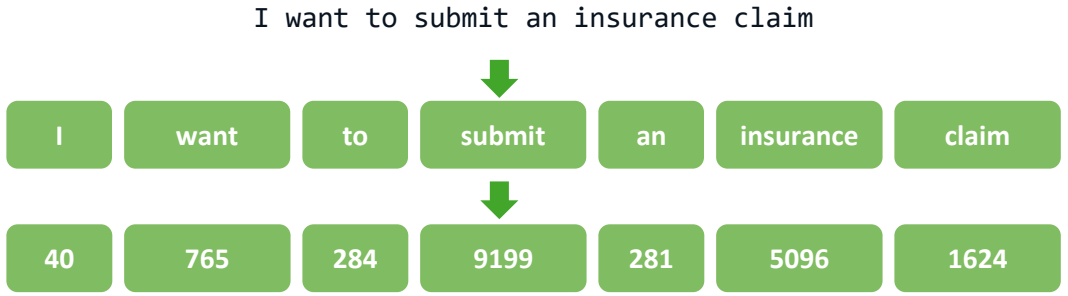
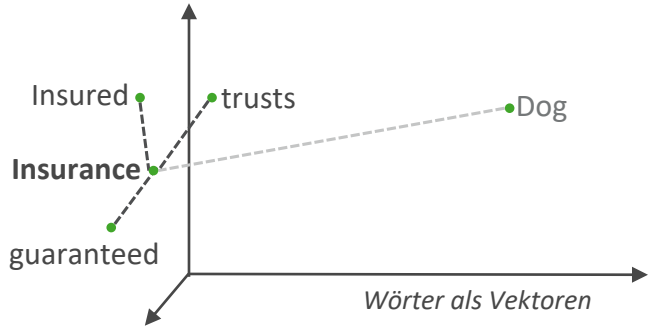
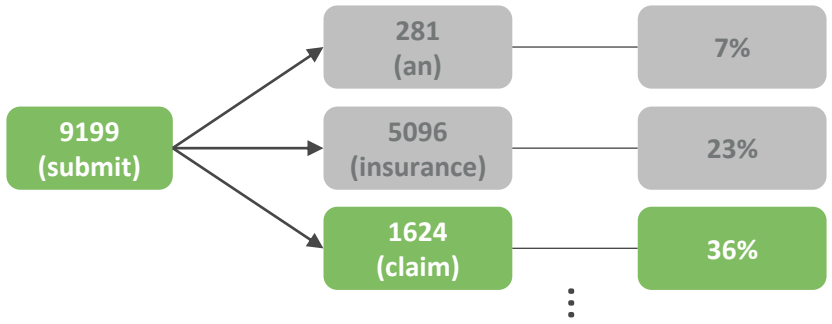
- Jedes Neuron erhält Input von Vorgängern und berechnet Output unter Beachtung der Aktivierungsfunktion
- Das Modell lernt durch die Anpassung der Gewichte auf den Verbindungen
- Je tiefer das Netz (Hidden Layer), desto komplexere Berechnungen sind möglich

Beispiele zu (Large) Language Models

		Anzahl Parameter	Vortrainiert auf	Kontext-erkennung	Input	Output
OpenAI GPT-3	generativ	ca. 175 x 10^9	499 x 10^9 Tokens („Internet“)	Unidirektional (alle Wörter vorher)	unimodal	Next-word probability auf User Prompts
OpenAI GPT-4	generativ	ca. 176 x 10^{10}	13 x 10^{12} Tokens („Internet“ + X)	Unidirektional (alle Wörter vorher)	multimodal	Next-word probability auf User Prompts

GPT = **G**enerative **P**re-Trained **T**ransformer GPT-3 Zahlen aus 2020 GPT-4 Zahlen aus 2023 **Token =** Wort, Silbe oder Zeichen

LLMs „verstehen“ Sprache indem Wörter als Vektoren in Bezug gesetzt werden

Komponente	Funktionsweise	Beispiel
Tokenization	Unstrukturierte Daten werden in numerische Daten (Vektoren) transformiert.	
Embedding	Der Abstand zwischen Vektoren repräsentiert die Korrelation zwischen Wörtern.	
Attention	Ein Token wird mit weiteren Tokens in Bezug gesetzt und gewichtet.	

Agenda

Large Language Models & Chat-GPT – Technische Grundlagen von LLMs & Use Cases

01 Technische Grundlagen

02 Chancen & Risiken

03 Use Cases

04 Vorstellung Q_PERIOR

Mit der Nutzung von GPT im Business Kontext ergeben sich einige (neue) Use Cases

- Kundenservice: semi-standardisierte Antworten auf Kundenanfragen
- Intelligente FAQ
- Kundenservice: Voice-/Chatbot zu Beratung
- Übersetzungen: Fremdsprache & Fachsprache
- Automatisierung: Generieren von Testdaten
- Automatisierung: Testen von Software
- Automatisierung: Erstellen von Dokumentationen

Dargestellte Use Cases sind Beispiele und keine umfängliche Auflistung

Durch die Verwendung von GPT im Business Kontext ergeben sich auch einige Risiken

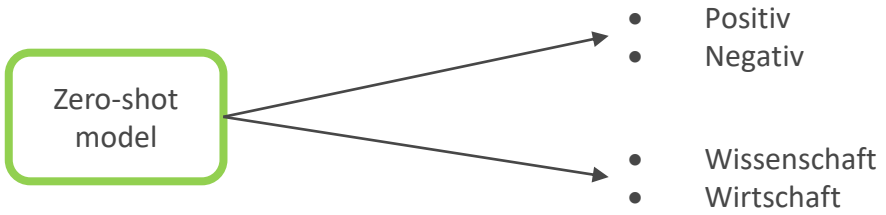
	Erklärung	Lösungsoptionen
Datenschutz	Technische Infrastruktur von API-Aufrufen in USA	Use-Cases ohne personenbezogene Daten verwenden
	Kein gleichwertiges Datenschutzniveau	Eigene Modelle hosten
Halluzination	Sequenzvorhersage basierend auf statistischen Verteilungen	Anfragen auf mehrere Prompts aufteilen
	Vorhersage ggf. thematisch abweichend (insb. bei langen Passagen)	Prompts mit spezifischen Informationen anreichern (Fokus)
Generische Antwort	Training auf breiter (generischer) Datenbasis	Fach-/Domänenwissen in Prompts einfließen lassen (Fokus)
	Spezialisierung auf Verwendung natürlicher Sprache	Domänenspezifische LLMs trainieren & verwenden
Kosten	Preismodell basierend auf Tokens (Wortbestandteile / Zeichen)	Kurze Prompts verwenden & wiederkehrende Fragen speichern
	Unterschiedliche Preismodelle für verschiedene GPT-Versionen	Preise vergleichen & ggf. Open-Source-Alternativen verwenden

Dargestellte Risiken sind Beispiele und keine umfängliche Auflistung

Verschiedene Ansätze können die Antwortgenauigkeit verbessern (Beispiel: Klassifikation)

Zero-shot classification

Klassifizierung neuer ungesehener Beispiele, die zu Klassen gehören, die in den Trainingsdaten nicht vorhanden waren.



One-shot classification

Jede neue Klasse hat ein trainiertes Beispiel. Das Erlernte soll auf nicht gesehene Beispiele angewendet werden.

Few-shot classification

Jede neue Klasse hat eine geringe Anzahl trainierter Beispiele. Das Erlernte soll auf nicht gesehene Beispiele angewendet werden.

Ergebnisse können mit Kontextanreicherung verbessert werden (Beispiel: Zero-Shot)

Simple Zero-Shot Classification

Eingabe

Frage:
Wie lautet der Name des abgebildeten Spechts? Flaumspecht oder Dreizehenspecht?

Antwort:

Ausgabe

Flaumspecht

Zero-Shot Classification mit Beschreibung

Eingabe

Beschreibung eines Dreizehenspechts:
Sein Körper ist schwarz-weiß gestreift und er hat einen gelb gefärbten Oberkopf.
Beschreibung eines Flaumspechts:
Seine Flügel sind schwarz
und mit weißen Punkten versehen. Sein Gefieder weist zudem rote Anteile auf.
Frage:
Wie lautet der Name des abgebildeten Spechts?

Antwort:

Ausgabe

Dreizehenspecht

Quelle: Language Is Not All You Need: Aligning Perception with Language Models

Ergebnisse können mit Chain-of-Thoughts verbessert werden (Beispiel: Zero-Shot)

Simple Zero-Shot

Eingabe

Frage:
Felix kann 16 Bälle jonglieren. Die Hälfte der Bälle sind Golfbälle, von denen wiederum die Hälfte blau ist. Wie viele blaue Golfbälle gibt es?

A: Die Antwort (als arabische Zahl) ist

Ausgabe

8

Zero-Shot mit Chain of Thoughts

Eingabe

Frage:
Felix kann 16 Bälle jonglieren. Die Hälfte der Bälle sind Golfbälle, von denen wiederum die Hälfte blau ist. Wie viele blaue Golfbälle gibt es?

A: *Lass uns Schritt für Schritt denken.*

Ausgabe

Es gibt insgesamt 16 Bälle. Die Hälfte der Bälle sind Golfbälle. Das heißt, dass es 8 Golfbälle gibt. Die Hälfte der Golfbälle ist blau. Das heißt, es gibt 4 blaue Golfbälle.

Quelle: Chain-of-Thought Prompting Elicits Reasoning in Large Language Models

Ein algorithmischer Denksansatz kann helfen komplexe Probleme korrekt zu lösen

Beobachtungen

LLMs sind sehr anfällig für arithmetische Berechnungsfehler, besonders im Bereich der großen Zahlen

LLMs können keine komplexen mathematischen Ausdrücke wie zum Beispiel Polynomgleichungen oder sogar Differentialgleichungen lösen

LLMs sind sehr ineffizient bei der Darstellung von Iterationen, insbesondere wenn die Anzahl der Iterationsschritte groß ist

In der Fibonacci-Folge gilt die Regel, dass jede Zahl gleich der Summe der beiden vorangehenden Zahlen ist. Angenommen, die ersten beiden Zahlen sind 0 und 1, was ist die 50. Zahl der Fibonacci-Folge?

Chain of Thought

Die erste Ziffer ist 0, die zweite 1, deshalb ist die dritte Ziffer 0+1=1. Die vierte Nummer ist 1+1=2. Die fünfte Nummer ist 1+2=3.
…….. (Skip Tokens)

Die fünfzigste Ziffer ist 32.432.268.459

32.432.268.459

Program of Thought

```
Fibonacci_length=51
Fibonacci_sequence=np.zeros(Fibonacci_ length)
Fibonacci_sequence[0]=0
Fibonacci_sequence[1]=1
For i in range(2, Fibonacci_ length):
                Fibonacci_sequence[i]=
                Fibonacci_sequence[i-1]+
                Fibonacci_sequence[i-2]
Ans= Fibonacci_sequence[-1]
```

12.586.269.025

Quelle: Program of Thoughts Prompting: Disentangling Computation from Reasoning for Numerical Reasoning Tasks

Agenda

Large Language Models & Chat-GPT – Technische Grundlagen von LLMs & Use Cases

01 Technische Grundlagen

02 Chancen & Risiken

03 Use Cases

04 Vorstellung Q_PERIOR

Mit der Nutzung von GPT im Business Kontext ergeben sich einige (neue) Use Cases

- Kundenservice: semi-standardisierte Antworten auf Kundenanfragen
- Intelligente FAQ
- Kundenservice: Voice-/Chatbot zu Beratung
- Übersetzungen: Fremdsprache & Fachsprache

Fokus

- Automatisierung: Generieren von Testdaten
- Automatisierung: Testen von Software
- Automatisierung: Erstellen von Dokumentationen

Dargestellte Use Cases sind Beispiele und keine umfängliche Auflistung

Unstrukturierte Trainings- und Testdaten zu generieren war bisher schwierig

Training- und Testdaten sollten …

- … in hoher Zahl verfügbar sein
- … von hoher Qualität sein

Datenschutz limitiert die Verwendung echter Daten

Da Kundendaten strengen Datenschutzbestimmungen unterliegen ist es oft nicht möglich diese zum Testen und Trainieren von KI-Modellen zu verwenden.

Manuelle Generierung benötigt Zeit & verursacht Kosten

Das (manuelle) Suchen, Verwalten und Generieren von Trainings- und Testdaten erfordert einen enormen Zeit- und Kostenaufwand.

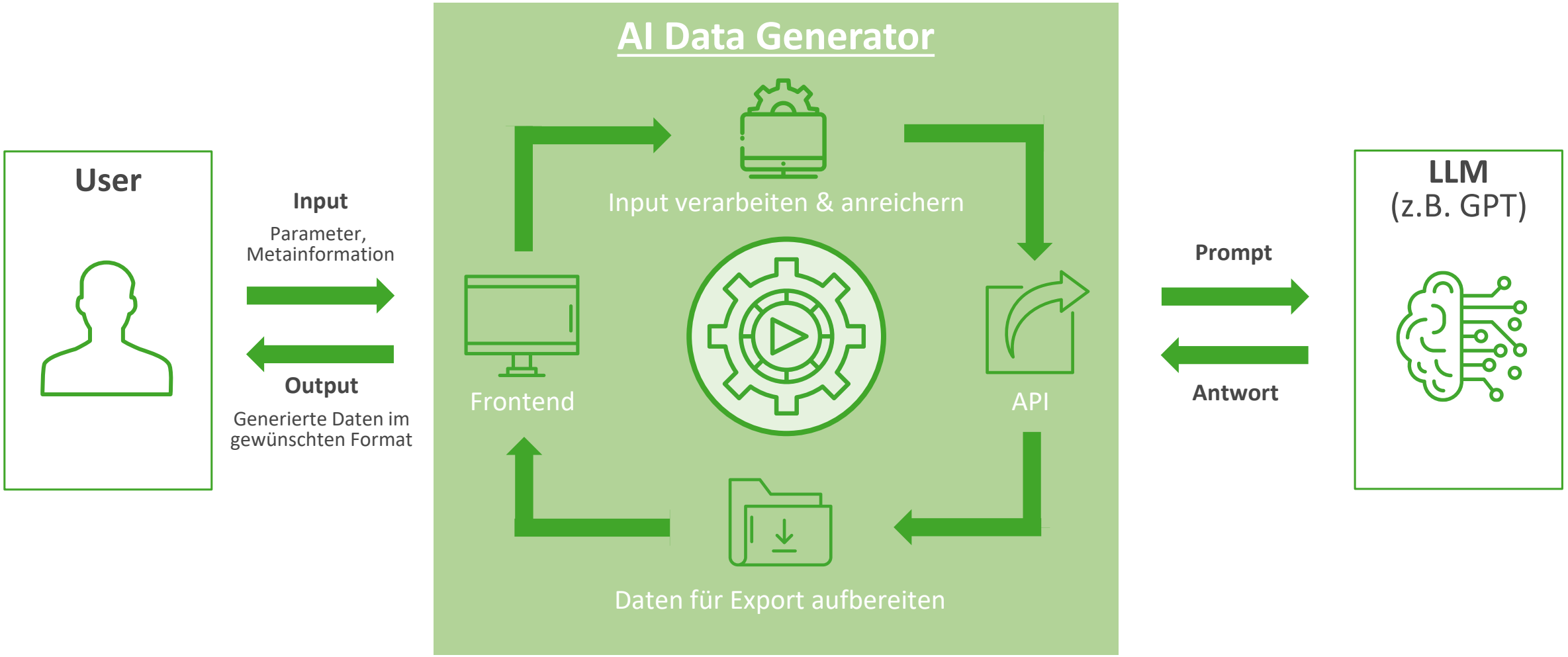
Daten sind nur in geringer Anzahl verfügbar

Auch wenn Daten verfügbar sind, ist die Menge of unzureichend oder Daten sind unvollständig, sodass es insgesamt an qualitativ hochwertigen Daten mangelt.

Unausgeglichene Datensätze erschweren das Modelltraining

Fehlende oder verzerrte Daten können zu ungenauen Ergebnissen oder Vorhersagen von KI-Modellen führen.

Der User-Input wird transformiert und Daten werden im gewünschten Format generiert



Durch die Datengenerierung können Projekte massiv beschleunigt werden

Generierung großer Mengen an hochwertigen Daten

Über eine Benutzeroberfläche lässt sich die gewünschte Datenmenge einfach festlegen und der **AI Data Generator** generiert einen hochwertigen Datensatz.

Generierung synthetischer aber realitätsnaher Daten

Die vom **AI Data Generator** erzeugten synthetischen Daten sind von echten Daten nicht zu unterscheiden, ohne auf (geschützte) Kundendaten zurückgreifen zu müssen.

Reduzierung manueller Aufwände & Kosten

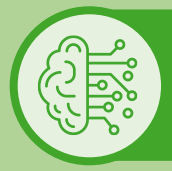
Da der **AI Data Generator** nur wenige Eingaben erfordert, werden manuelle Aufwände und damit verbundene Kosten für die Datenerstellung erheblich reduziert.

Automatisierung von Generierung & Testen

Der **AI Data Generator** ermöglicht eine Vollautomatisierung des gesamten Zyklus von Datengenerierung bis zum Testen (auch Human-in-the-Loop Ansätze möglich).

Vorteile / Nutzen

Projektbeschleuniger



Aufbau domänen-spezifischer LLMs

Beschleunigung der Prototyp-Entwicklung

Agenda

Large Language Models & Chat-GPT – Technische Grundlagen von LLMs & Use Cases

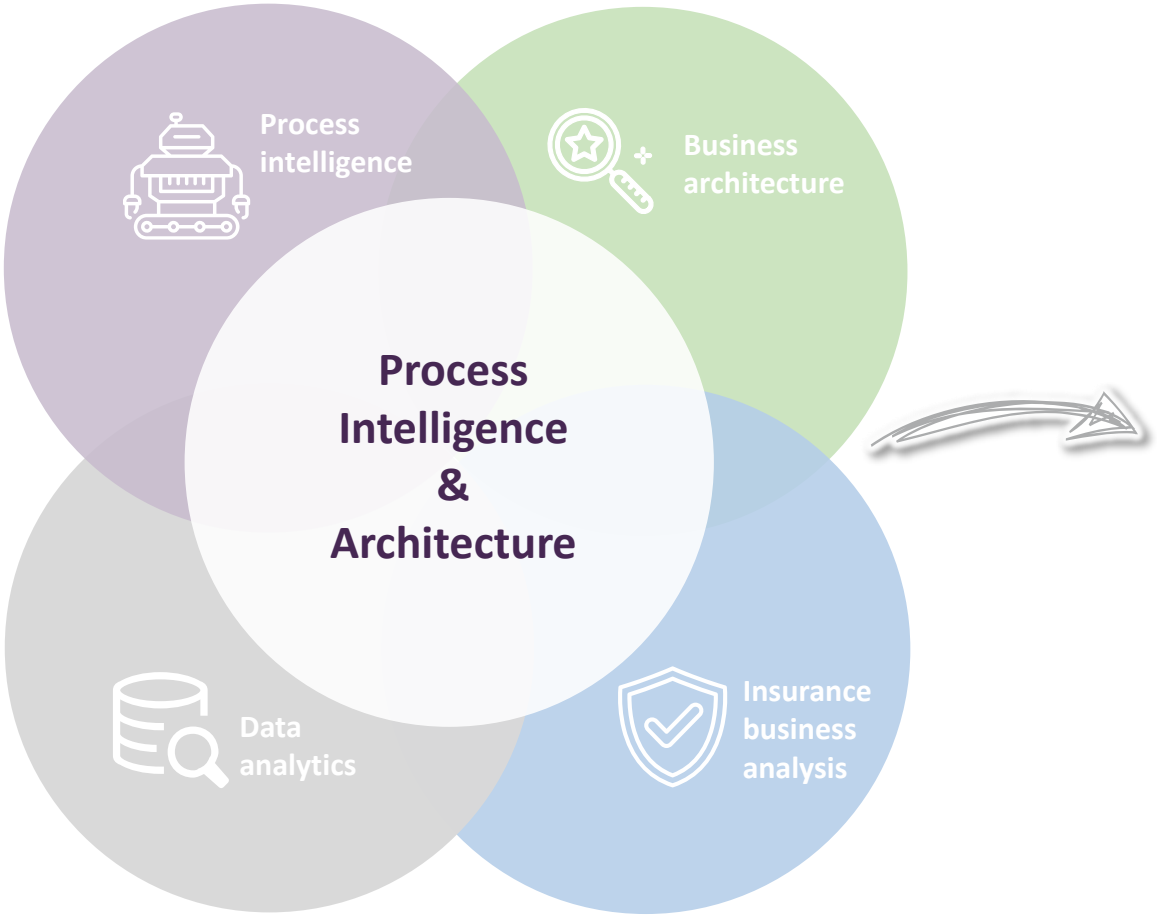
01 Technische Grundlagen

02 Chancen & Risiken

03 Use Cases

04 Vorstellung Q_PERIOR

Wir beschäftigen uns mit vielen Themen rund um KI in Versicherungsprozessen



Prototypen im Bereich Process Intelligence

Automatisierte Schadenanalyse: Bewertung der Schadensdeckung der Privathaftpflichtversicherung und automatische Auszahlung

Erweitertes medizinisches Underwriting: NLP extrahiert Krankheitsdiagnosen aus medizinischen Berichten und gleicht diese mit der Risikobereitschaft des Rückversicherers ab

NLP-basierte Unterstützung der Vertragsverwaltung: NLP extrahiert wichtige Versicherungsinformationen, z.B. Prämie und Risiko, aus Verträgen, um die Administration zu entlasten

Biometrische Sprachbasierte Assistenten: Am Kunden-Touchpoint des Umzugs kann eine Sprachassistent die Kundenbetreuung entlasten. Die Kundenidentifikation erfolgt durch Stimmbiometrie.

Optimierung von Gutachtenzeitpunkten: Der optimale Zeitpunkt eines Gutachtens zur Arbeitsunfähigkeit wird anhand von Personen- und Diagnosedaten mittels KI vorhergesagt, um Gutachten und Krankentagegeld einzusparen.

Wavestone & Q_PERIOR | Ein neues globales Beratungsunternehmen

17

LÄNDER WELTWEIT

DEUTSCHLAND/ FRANKREICH / SCHWEIZ

ASIEN / NORD AMERIKA
VEREINIGTES KÖNIGREICH

BELGIEN / BOSNIEN-HERZEGOWINA
ITALIEN / LUXEMBURG / MARROKO
ÖSTERREICH / POLEN / RUMÄNIEN
SPANIEN

>800 MIO.

KONSOLIDIERTER UMSATZ 2023 [€]

1 MRD. UMSATZ IN 2025 [€]

>5.500

MITARBEITENDE

Top 2024 Company kununu

brandeins /thema 2023 Best Consultants Germany

Great Place To Work. Certified OKT 2023-OKT 2024 DE

15 %

UMSATZSTEIGERUNG AUF LÄNGERE SICHT

WAVESTONE + Q_PERIOR

Thank you ! – Let´s get in touch

Dr. Felix Happach
Senior Consultant
felix.happach@q-perior.com

SCAN ME

Together With You Q_PERIOR

www.q-perior.com