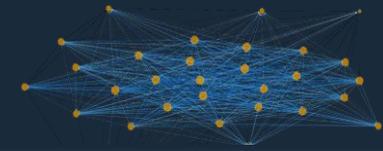


# Sprachmodelle: Wie neuronale Netze mit Texten rechnen

# Über das Unternehmen



Strukturen in Daten erkennen und Zusammenhänge visualisieren

Kundenspezifische Angebote auswählen

Bilder klassifizieren und Objekte lokalisieren

Punktwolken analysieren und 3D-Objekte erkennen

In Videos Bewegungen analysieren

Zeitreihen fortsetzen

Sensordaten analysieren und Geräteausfälle prognostizieren

Prognosen mit Bayes'schen neuronalen Netzen statistisch bewerten

Texte übersetzen und zusammenfassen

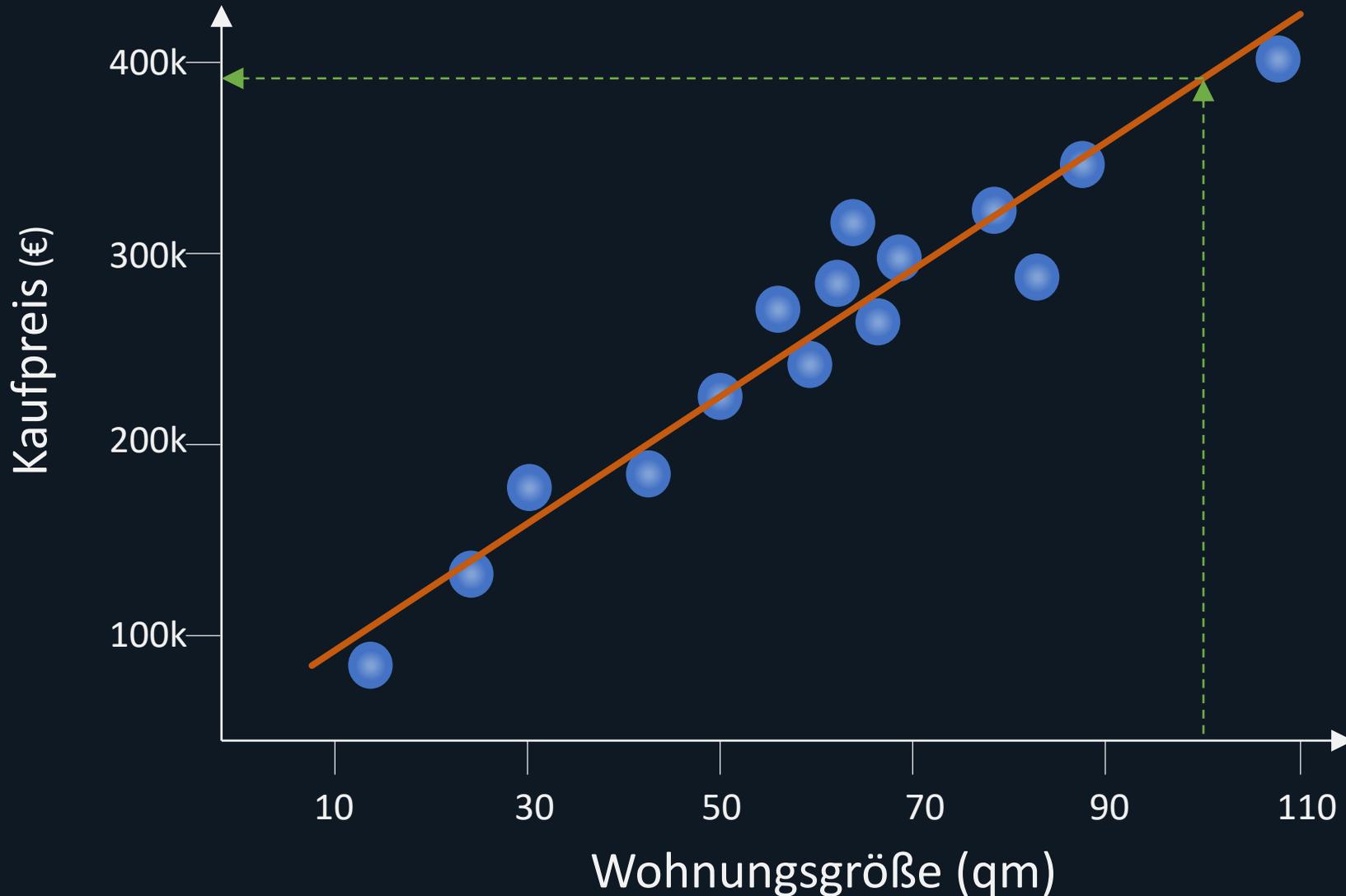
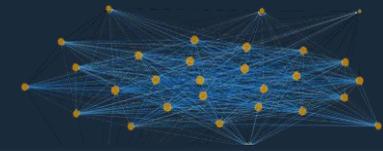
Sprachmodelle durch Transfer Learning anwenden

Dynamische Systeme mit Reinforcement Learning steuern

Und hier ist Raum für Ihr individuelles Projekt

- Seit 06/2020: Selbständiger Dozent für Machine Learning und Deep Learning
- 2016 – 2020: Postdoc am Karlsruher Institut für Technologie
- 2011 – 2015: Promotion und Postdoc an der University of St Andrews, UK
- 2009 – 2011: Masterstudium Photonik an der Hochschule München
- 2008 – 2010: Ingenieur bei der PNDetector GmbH in München
- 2003 – 2007: Diplomstudium Allgemeine Physik an der nta in Isny

# Intro: Machine Learning



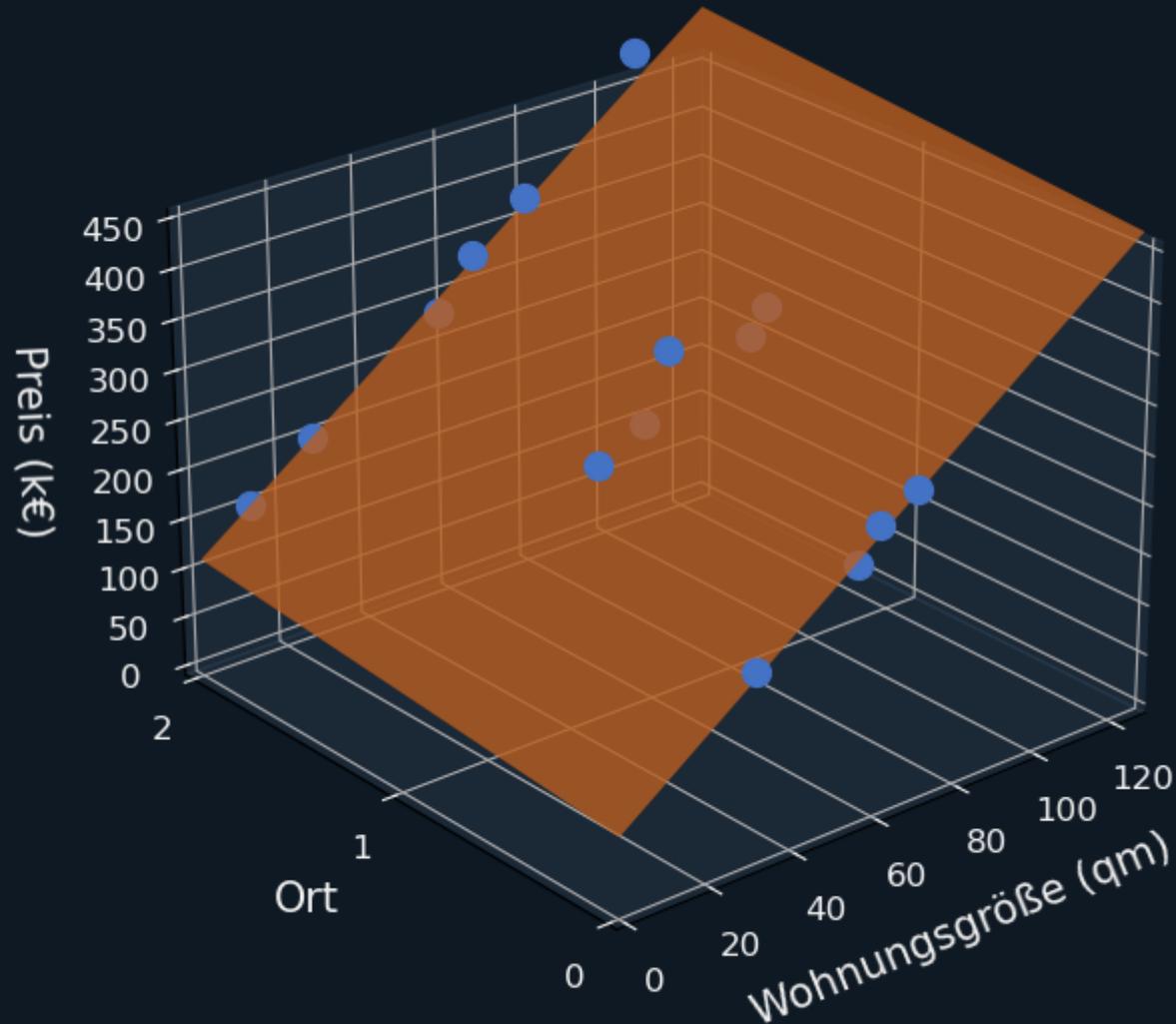
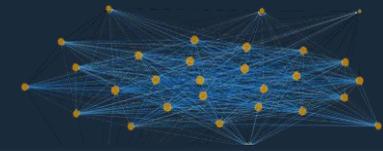
Wie viel kostet eine  
100 qm große  
Wohnung?

Modellauswahl:  
 $y = wx + b$

Modell fitten,  
 $w$  und  $b$  optimieren

Prognose treffen

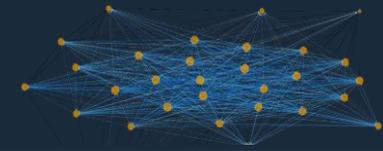
# Mehrere Merkmale



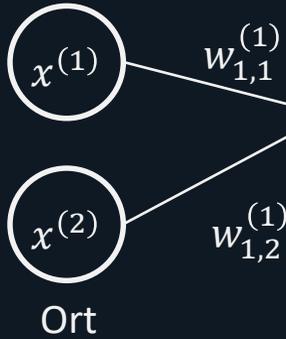
Die Prognose verbessern,  
indem mehrere Merkmale  
berücksichtigt werden

$$y = w_1x^{(1)} + w_2x^{(2)} + b$$

# Künstliche neuronale Netze

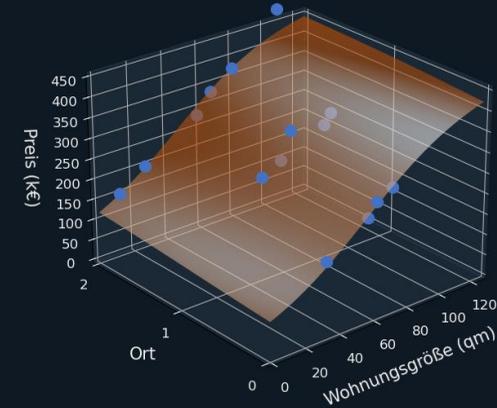
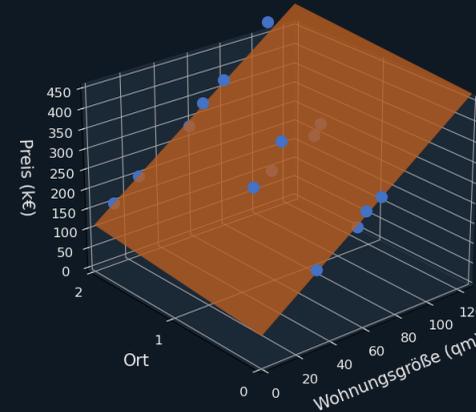


Wohnungsfläche

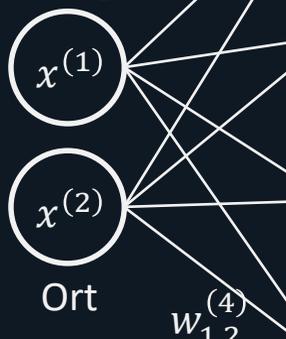


Prognostiziertes Preis

$$y = (w_1 x^{(1)} + w_2 x^{(2)} + b)$$



Wohnungsfläche



$$y_1^{(1)} = g_1 (w_{1,1}^{(1)} x^{(1)} + w_{1,2}^{(1)} x^{(2)} + b_1^{(1)})$$

$$y_1^{(2)} = g_1 (w_{1,1}^{(2)} x^{(1)} + w_{1,2}^{(2)} x^{(2)} + b_1^{(2)})$$

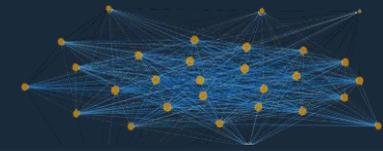
$$y_1^{(3)} = g_1 (w_{1,1}^{(3)} x^{(1)} + w_{1,2}^{(3)} x^{(2)} + b_1^{(3)})$$

$$y_1^{(4)} = g_1 (w_{1,1}^{(4)} x^{(1)} + w_{1,2}^{(4)} x^{(2)} + b_1^{(4)})$$

Prognostiziertes Preis

$$y_2^{(1)} = g_2 \left( w_{2,1}^{(1)} y_1^{(1)} + w_{2,2}^{(1)} y_1^{(2)} + w_{2,3}^{(1)} y_1^{(3)} + w_{2,4}^{(1)} y_1^{(4)} + b_2^{(1)} \right)$$





*"Wir lieben das Allgäu", "Wir fördern Künstliche Intelligenz im Allgäu"*

1. Texte in einzelne Wörter zerlegen und ein Wörterbuch erstellen:

["Allgäu", "das", "fördern", "im", "Intelligenz", "Künstliche", "lieben", "wir"]

2. Jedes Wort in eine Ganzzahl umwandeln

{"Allgäu": 1, "das": 2, "fördern": 3, ...}

[9, 8, 2, 1], [9, 3, 7, 6, 5, 1]

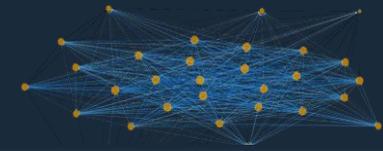
3. Eine Embedding-Schicht weist jedem Wort einen niedrigdimensionalen Vektor zu

1  $\rightarrow$  [0.25, -0.12, 0.56], 2  $\rightarrow$  [0.37, 0.72, -0.18], 3  $\rightarrow$  [-0.61, 0.22, -0.72]

Die Wort-Vektoren werden erlernt und definieren wie Wörter in Bezug stehen

Wörter mit ähnlicher Bedeutung führen zu ähnlichen Wort-Vektoren

# Ein Beispiel für Wort-Vektoren



'Das schöne Allgäu',

'So toll hier im Allgäu',

'Uns gefallen die Berge und Seen',

'Wir mögen die Landschaft und die Berge',

'Ganz toll im Allgäu',

'Wir mögen das Allgäu',



'Uns gefiel es leider nicht',

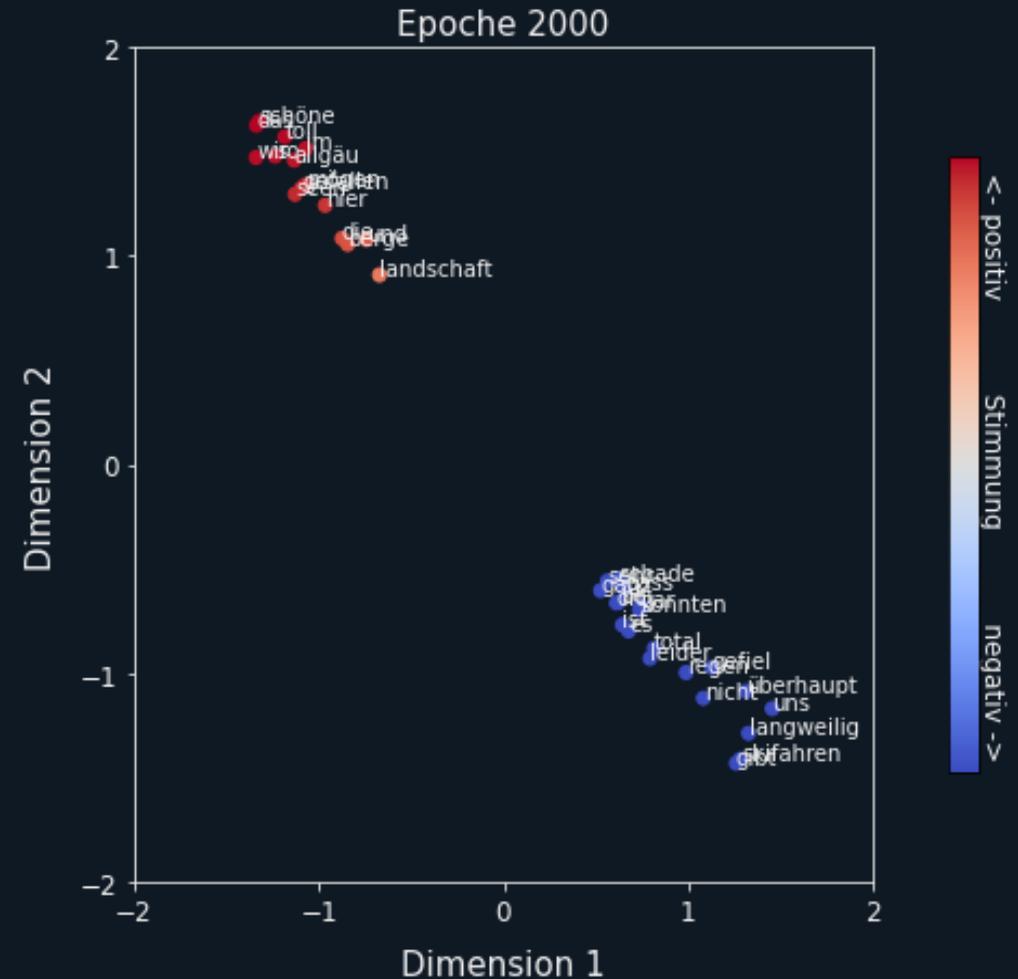
'Bei Regen ist es total langweilig',

'Ganz langweilig!',

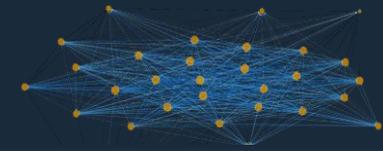
'So schade, dass es oft Regen gibt',

'Sehr schade, wir konnten gar nicht skifahren',

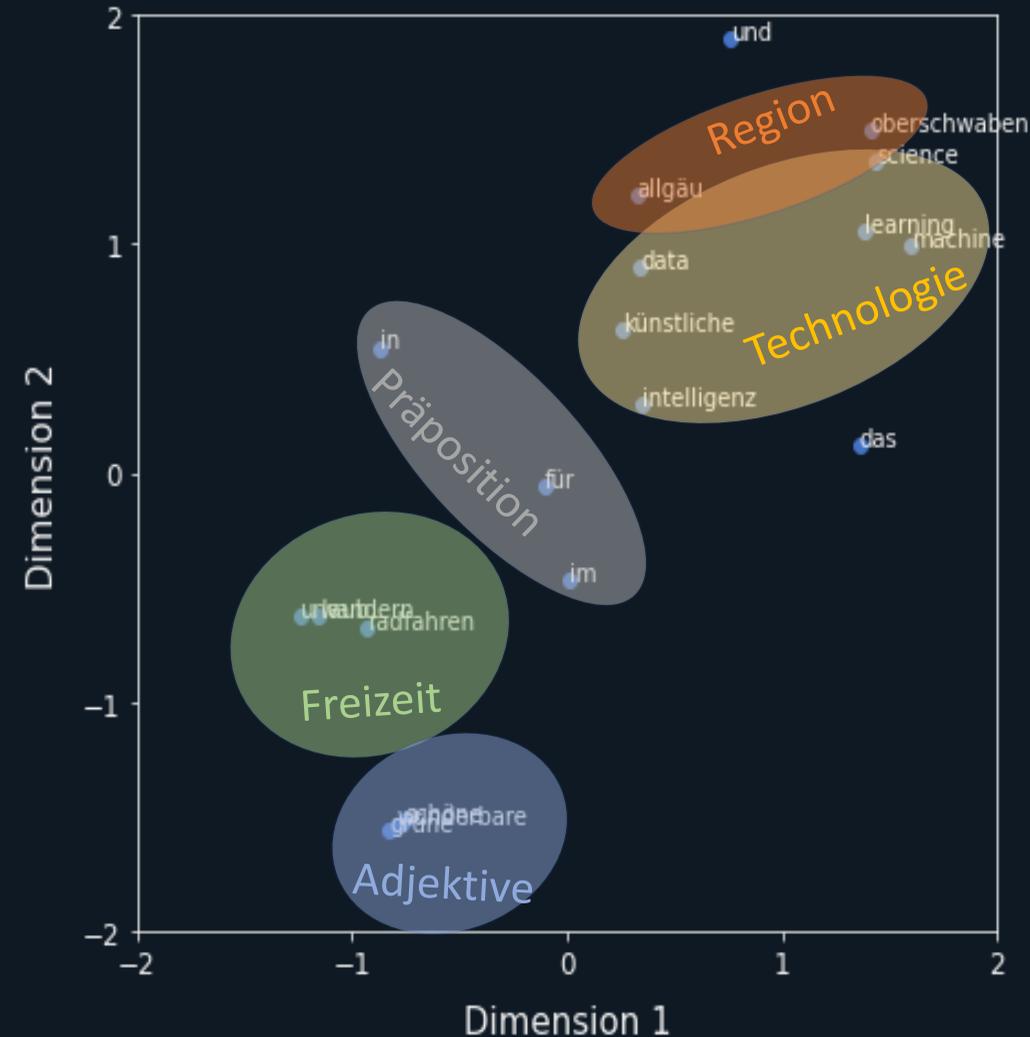
'Das gefiel uns überhaupt nicht'



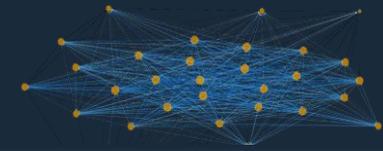
# Ein Wort maskieren und prognostizieren



- 'Das <mask> Allgäu' → 'schöne',
- 'Das <mask> Allgäu' → 'wunderbare',
- 'Das grüne <mask>' → 'Allgäu',
- 'Radfahren <mask> Allgäu' → 'im',
- '<mask> im Allgäu' → 'Wandern',
- 'Radfahren <mask> Oberschwaben' → 'in',
- 'Urlaub in <mask>' → 'Oberschwaben',
- 'Künstliche Intelligenz für <mask> Allgäu' → 'das',
- 'Künstliche <mask> für Oberschwaben' → 'Intelligenz',
- 'Data Science für <mask>' → 'Oberschwaben',
- 'Data Science und <mask> Learning' → 'Machine',
- 'Machine <mask> für das Allgäu' → 'Learning'



# Sprachmodelle und Transfer-Learning



Basis-Modell:

Große neuronale Netze mit bis zu 175 Milliarden Parametern

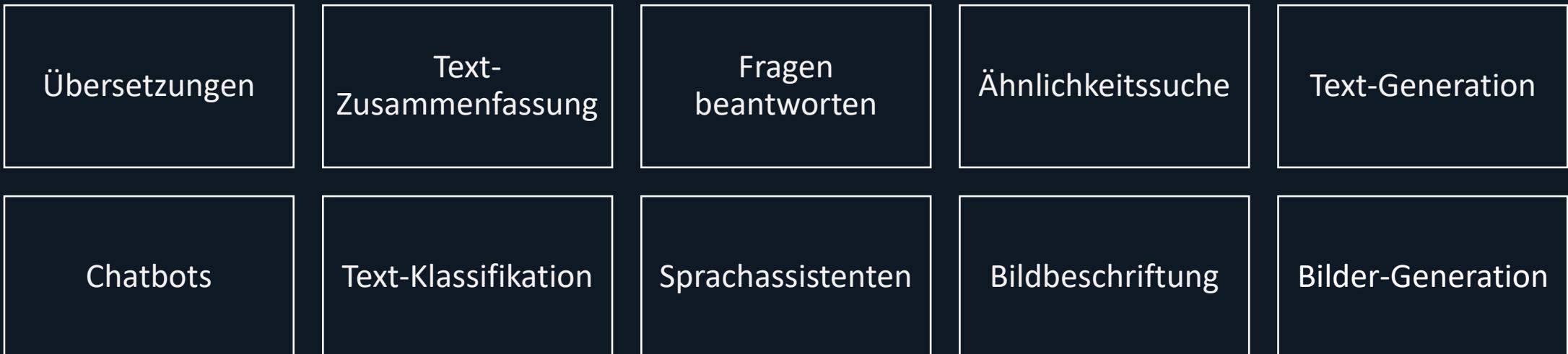
Trainiert durch Next-Word-Prediction

Trainiert auf großen Text Korpora wie Wikipedia, Blogs, Büchern, etc.

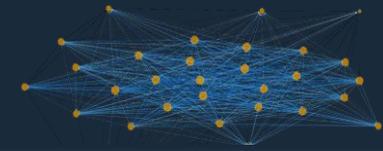
Transfer-Learning:

Durch weiteres Training das Modell an individuelle Aufgaben anpassen

Typische Textverarbeitungs Pipelines:



# Online Ressourcen und Demos



Chatbots, kreatives Schreiben:

<https://you.com/>, <https://openai.com/blog/chatgpt>

Rechtschreibung korrigieren:

<https://languagetool.org/de> , <https://www.deepl.com/write>

Übersetzungen:

<https://www.deepl.com/translator>, <https://translate.google.com/>

Bild aus Text erstellen:

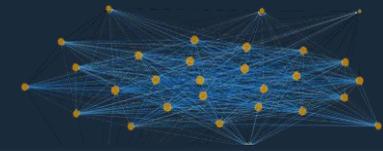
<https://openai.com/product/dall-e-2>, <https://stablediffusionweb.com/#demo>

Video Ansprache erstellen:

<https://www.synthesia.io>

Sprache in Text umwandeln:

<https://speechnotes.co/>, <https://www.spechtexer.com/>



HuggingFace Transformers Bibliothek: [https://huggingface.co/docs/transformers/main\\_classes/pipelines](https://huggingface.co/docs/transformers/main_classes/pipelines)

Pipelines für diverse Anwendungen der Textverarbeitung

Fertig trainierte Modelle. Meist lizenzfrei verwendbar

Anwendung in 4 Zeilen Code (Python)

```
from transformers import pipeline  
en_de_translator = pipeline("translation_de_to_en", model='google/bert2bert_L-24_wmt_de_en')  
result = en_de_translator("Wir unterstützen Unternehmen bei der Datenanalyse durch individuelle Beratung und Projekte.")  
print(result) # we want to support you in data analysis through individual consultation and projects.
```

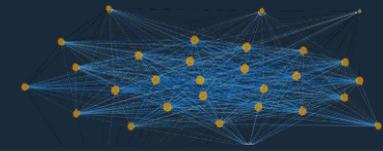
Von:  
Sascha Rothe  
Shashi Narayan  
Aliaksei Severyn

OpenAI GPT3.5: <https://platform.openai.com/docs/api-reference>

Basiert auf Textergänzung. Für kreatives Schreiben. Übersetzungen. Abfragen von Allgemeinwissen

Benutzeraccount notwendig. Kosten pro Wort nach einer Probephase

```
import openai  
openai.api_key = "...your_personal_openai_key..."  
result = openai.Completion.create(model="text-davinci-003",  
                                prompt="Tim ist 17, Paul ist 12. Wie viele Jahre ist Tim älter als Paul?")  
print(result) # Tim ist 5 Jahre älter als Paul.
```



1. Die Verarbeitung von natürlicher Sprache durch neuronale Netze eröffnet Möglichkeiten zur Automation von alltäglichen Bürotätigkeiten
2. Die Sprachmodelle basieren auf schlicht sehr großen Trainingsdaten und statistischen, nachvollziehbaren Berechnungen
3. Für viele Anwendungen gibt es bereits fertige Lösungen

**Bleiben Sie interessiert!**

**Nutzen Sie die Tools!**

**Ihr Arbeitsalltag wird sich erleichtern!**