

# KI-Projekte programmieren mit Python

Dein einfacher Einstieg

# DAS INHALTS- VERZEICHNIS

» Hier geht's  
direkt  
zum Buch

# Inhaltsverzeichnis

	<b>Einleitung</b> . . . . .	9
	Wie ist das Buch aufgebaut? . . . . .	9
	Programmbeispiele zum Download und Colab Notebooks . . . . .	10
<b>1</b>	<b>Denkende Maschinen</b> . . . . .	<b>11</b>
1.1	Was ist Intelligenz? . . . . .	11
1.1.1	Rechnen mit Zahlen. . . . .	12
1.1.2	Gedächtnis . . . . .	15
1.1.3	Räumliches Vorstellungsvermögen . . . . .	19
1.1.4	Sprachverständnis . . . . .	20
1.1.5	Schlussfolgerndes Denken . . . . .	20
1.1.6	Wortflüssigkeit . . . . .	22
1.1.7	Wahrnehmungsgeschwindigkeit . . . . .	25
1.1.8	Starke und schwache KI. . . . .	26
1.2	Projekt: Wie schnell kann ein Computer rechnen? . . . . .	27
1.3	Maschinen als Ersatzmensch – Der Turing-Test . . . . .	31
1.4	Projekt: Eine freundliche Maschine. . . . .	33
1.4.1	Den Computer sprechen lassen . . . . .	35
1.5	Projekt: Ein Stichwort-Chatbot . . . . .	35
1.6	Projekt: Eine KI als Spielgegner . . . . .	42
1.6.1	Das Nim-Spiel. . . . .	42
1.6.2	Die Programmierung der Basisversion . . . . .	43
1.7	Rückblick . . . . .	49
1.8	Lösungen . . . . .	49
<b>2</b>	<b>Sprache und Denken – Natural Language Processing (NLP)</b> . . . . .	<b>53</b>
2.1	Trainingscamp: Strings. . . . .	53
2.1.1	Station 1: Strings erstellen . . . . .	53
2.1.2	Station 2: Unicode-Zeichen. . . . .	55
2.1.3	Station 3: Strings durchsuchen . . . . .	56
2.1.4	Station 4: Neue Strings aus alten Strings . . . . .	57
2.1.5	Station 5: Formatierung . . . . .	58
2.2	Projekt: Storytelling . . . . .	59
2.2.1	Mad Libs . . . . .	59
2.2.2	Eine digitale Version . . . . .	60
2.3	Sprachmodelle nutzen – Natural Language Processing (NLP) . . . . .	61
2.3.1	spaCy in einer virtuellen Umgebung installieren . . . . .	62
2.4	NLP mit spaCy ausprobieren . . . . .	64
2.4.1	Tokenisierung. . . . .	64

2.4.2	Stoppwörter . . . . .	68
2.4.3	Einen Text in Sätze zerlegen . . . . .	69
2.5	Projekt: Mit NLP einen Text automatisch zusammenfassen . . . . .	70
2.5.1	Vorüberlegung: Wie erkennt man, was wichtig ist? . . . . .	70
2.5.2	Programmierung . . . . .	72
2.6	Was ist ein Apfel? Die Bedeutung von Wörtern . . . . .	76
2.6.1	Semantisches Netz . . . . .	76
2.6.2	Embeddings und Ähnlichkeit – Experimente mit spaCy. . . . .	77
2.6.3	Ähnlichkeit . . . . .	81
2.7	Projekt: Schlag nach bei Goethe! Ähnliche Wörter finden . . . . .	82
2.8	Projekt: Fitness-Chatbot für Reisende . . . . .	85
2.9	Rückblick . . . . .	88
2.10	Lösungen . . . . .	89
<b>3</b>	<b>Lernende Programme . . . . .</b>	<b>93</b>
3.1	Wie Computer »intelligent« werden . . . . .	93
3.1.1	Überwachtes Lernen: Regression . . . . .	93
3.1.2	Überwachtes Lernen: Klassifizieren . . . . .	102
3.1.3	Unüberwachtes Lernen . . . . .	104
3.1.4	Verstärkungslernen . . . . .	106
3.2	Projekt: Lernen durch Versuch und Irrtum – Wegesuche auf dem Mond . . . . .	107
3.2.1	Turtle-Grafik . . . . .	108
3.2.2	Das Simulationsprogramm . . . . .	109
3.2.3	Fazit . . . . .	120
3.3	Rückblick . . . . .	121
3.4	Lösungen . . . . .	121
<b>4</b>	<b>Inspiziert durch das Gehirn – Künstliche neuronale Netze . . . . .</b>	<b>125</b>
4.1	Das Gehirn – Ein natürliches neuronales Netz . . . . .	125
4.2	Künstliches neuronales Netz . . . . .	127
4.2.1	Aktive Knoten . . . . .	129
4.2.2	Aktivierungsfunktionen . . . . .	131
4.3	Ein neuronales Netz entwerfen und trainieren . . . . .	135
4.3.1	Uneinigkeit erkennen . . . . .	136
4.3.2	Vorhersagen berechnen . . . . .	137
4.3.3	Trainingsprozess und Batchgröße . . . . .	138
4.3.4	Verlustfunktion (Loss) . . . . .	139
4.3.5	Gradientenabstieg . . . . .	141
4.3.6	Klassischer Gradientenabstieg und Stochastischer Gradientenabstieg . . . . .	144
4.3.7	Adam-Optimierer . . . . .	144
4.4	Trainingscamp: Arrays verarbeiten mit NumPy . . . . .	146
4.4.1	Vorbereitung . . . . .	147

4.4.2	Station 1: Arrays erzeugen . . . . .	147
4.4.3	Station 2: Rechnen mit Arrays. . . . .	150
4.4.4	Station 3: Arrays in Form bringen . . . . .	152
4.5	Projekt: Uneinigkeit erkennen . . . . .	155
4.5.1	Vorbereitung . . . . .	155
4.5.2	Google Colab verwenden . . . . .	156
4.5.3	Programmierung . . . . .	158
4.6	Projekt: Ziffern erkennen . . . . .	163
4.6.1	Programmierung . . . . .	164
4.7	Rückblick . . . . .	171
4.8	Lösungen . . . . .	171
<b>5</b>	<b>Der Computer als Künstler – Bilder generieren mit einem GAN . . . . .</b>	<b>175</b>
5.1	Was ist generative KI? . . . . .	175
5.2	Wie funktioniert ein GAN? . . . . .	177
5.3	Trainingscamp: Tensoren . . . . .	178
5.3.1	Station 1: Eigenschaften eines Tensors. . . . .	178
5.3.2	Station 2: Tensoren erzeugen . . . . .	182
5.3.3	Station 3: Tensoren in Form bringen . . . . .	184
5.3.4	Station 4: Zugriff auf Teile eines Tensors . . . . .	186
5.3.5	Station 5: Tensoren als Bilder anzeigen . . . . .	187
5.4	Projekt: Kreative Ziffern . . . . .	191
5.4.1	Programmierung . . . . .	192
5.4.2	Weiterentwicklung: Modelle speichern und trainierte Modelle nutzen. . . . .	204
5.5	Projekt: Künstliche Gesichter . . . . .	206
5.5.1	Vorbereitung . . . . .	207
5.5.2	Programmierung . . . . .	208
5.6	Rückblick . . . . .	210
5.7	Lösungen . . . . .	211
<b>6</b>	<b>Textproduktion mit rekurrenten neuronalen Netzen (RNN) . . . . .</b>	<b>217</b>
6.1	»Es war einmal« – Ein Sprachmodell mit Bleistift und Papier . . . . .	217
6.2	Wie funktioniert ein RNN? . . . . .	220
6.2.1	Hat ein künstliches neuronales Netz ein Gedächtnis? . . . . .	220
6.2.2	Neuronen mit Gedächtnis – Rekurrente neuronale Netze (RNNs) . . . . .	221
6.2.3	Gradientenverschwinden und Gradientenexplosionen. . . . .	223
6.2.4	Die Arbeitsweise von LSTM-Zellen. . . . .	225
6.3	Projekt: In sechs Schritten zum Textgenerator . . . . .	229
6.3.1	Die Grundidee . . . . .	230
6.3.2	Die Programmierung . . . . .	232

6.4	Das Projekt weiterentwickeln . . . . .	242
6.4.1	Mehr Zufall . . . . .	243
6.4.2	Mehr Komplexität durch zusätzliche Schichten . . . . .	244
6.4.3	Eine moderne Weiterentwicklung: Gated Recurrent Unit (GRU) . . . . .	245
6.4.4	Preprocessing: Satzzeichen verarbeiten . . . . .	246
6.5	Eine Alternative zu RNN: Transformer . . . . .	248
6.5.1	Programmierung . . . . .	249
6.6	Rückblick . . . . .	251
6.7	Lösungen . . . . .	252
<b>7</b>	<b>Musik komponieren mit KI . . . . .</b>	<b>255</b>
7.1	Trainingscamp: Noten verarbeiten mit music21 . . . . .	255
7.1.1	Station 1: Vorbereitung . . . . .	255
7.1.2	Station 2: Melodien mit Streams . . . . .	256
7.1.3	Station 3: Eine Melodie definieren und abspielen . . . . .	257
7.1.4	Station 4: Mehrstimmige Partituren mit Scores . . . . .	260
7.1.5	Station 5: Aus einer MIDI-Datei die Melodie gewinnen . . . . .	263
7.1.6	Station 6: Eine Melodie transponieren . . . . .	266
7.2	Projekt: Melodien komponieren . . . . .	267
7.2.1	Erweiterung: Mehr Trainingsdaten . . . . .	275
7.2.2	Anregungen für Experimente und Entwicklungen . . . . .	278
7.3	Rückblick . . . . .	279
7.4	Lösungen . . . . .	280
<b>8</b>	<b>Diffusionsmodelle – Bilder aus Rauschen . . . . .</b>	<b>283</b>
8.1	Die Grundidee . . . . .	283
8.2	Faltung (Konvolution) – Ein Bild durch Features beschreiben . . . . .	285
8.2.1	Features im Alltag . . . . .	285
8.2.2	Mit Filtern Features erkennen . . . . .	286
8.2.3	Padding und Stride . . . . .	288
8.2.4	Lernende Filter in einem Convolutional Neural Network (CNN) . . . . .	289
8.2.5	Faltung als neuronales Netz . . . . .	290
8.3	Wie entrauscht man ein verrauschtes Bild? . . . . .	291
8.3.1	Das U-Net und Bildsegmentierung in der Medizin . . . . .	291
8.3.2	Faltung, Pooling und Skip-Connections – Die Magie der U-Nets . . . . .	293
8.3.3	Veranschaulichung: Ein beschädigtes Bild rekonstruieren . . . . .	299
8.4	Projekt: Künstliche Ziffern . . . . .	300
8.5	Rückblick . . . . .	312
8.6	Lösungen . . . . .	313
	<b>Stichwortverzeichnis . . . . .</b>	<b>315</b>