

# PyTorch für KI und ML

Das Praxisbuch für generative KI und  
Machine Learning

# DAS INHALTS- VERZEICHNIS

» Hier geht's  
direkt  
zum Buch

<b>Vorwort</b> .....	<b>15</b>
<b>Einleitung</b> .....	<b>17</b>
<b>Teil I Mit PyTorch ML-Modelle erstellen</b>	
<hr/>	
<b>1 Einführung in PyTorch</b> .....	<b>25</b>
Was ist Machine Learning? .....	25
Grenzen der traditionellen Programmierung .....	27
Vom Programmieren zum Lernen .....	29
Was ist PyTorch? .....	31
PyTorch verwenden .....	33
Installation von PyTorch auf der Kommandozeile .....	33
PyTorch in PyCharm verwenden .....	34
PyTorch in Google Colab verwenden .....	36
Einstieg ins Machine Learning .....	38
Sehen, was das Netzwerk gelernt hat .....	44
Zusammenfassung .....	45
<b>2 Einführung in Computer Vision</b> .....	<b>47</b>
Die Funktionsweise der Computer Vision .....	47
Der Fashion-MNIST-Datensatz .....	48
Neuronen für Computer Vision .....	50
Das neuronale Netzwerk entwerfen .....	51
Der vollständige Code .....	53
Das Netzwerk trainieren .....	58
Die Modellausgaben genauer untersuchen .....	61
Überanpassung .....	63
Early Stopping (»vorzeitiger Abbruch«) .....	64
Zusammenfassung .....	65

<b>3</b>	<b>Über die Grundlagen hinaus: Merkmale in Bildern erkennen</b>	<b>67</b>
	Faltung	68
	Pooling	69
	Convolutional Neural Networks implementieren	71
	Das Convolutional Neural Network untersuchen	74
	Ein CNN für die Unterscheidung zwischen Pferden und Menschen	76
	Der Datensatz »Horses or Humans«	76
	Mit den Daten umgehen	77
	CNN-Architektur für »Horses or Humans«	79
	Validierung für den Datensatz »Horses or Humans«	82
	Die »Horses or Humans«-Bilder testen	84
	Image Augmentation	86
	Transferlernen	90
	Mehrfachklassifizierung	95
	Dropout-Regularisierung	98
	Zusammenfassung	101
<b>4</b>	<b>Daten mit PyTorch verwenden</b>	<b>103</b>
	Einstieg in Datasets	104
	Die FashionMNIST-Klasse erforschen	106
	Generische Dataset-Klassen	106
	ImageFolder	106
	DatasetFolder	107
	FakeData	107
	Custom Splits verwenden	108
	Der ETL-Prozess zur Datenverwaltung im Machine Learning	109
	Die Ladephase optimieren	111
	Die DataLoader-Klasse verwenden	112
	Batching	112
	Daten mischen	112
	Daten parallel laden	113
	Benutzerdefiniertes Daten-Sampling	113
	ETL parallelisieren, um die Trainingsleistung zu steigern	113
	Zusammenfassung	115
<b>5</b>	<b>Einführung in die Verarbeitung natürlicher Sprache</b>	<b>117</b>
	Sprache in Zahlen codieren	117
	Einstieg in die Tokenisierung	118
	Sätze in Sequenzen umwandeln	121
	Stoppwörter entfernen und Text aufräumen	124
	HTML-Tags entfernen	125
	Stoppwörter entfernen	125
	Interpunktionszeichen entfernen	125

Mit echten Datenquellen arbeiten .....	126
Textdatensätze erhalten .....	126
Text aus CSV-Dateien lesen .....	130
Text aus JSON-Dateien auslesen .....	133
Zusammenfassung .....	135
<b>6 Stimmungen mit Embeddings programmierbar machen .....</b>	<b>137</b>
Bedeutung aus Wörtern ableiten .....	137
Ein einfaches Beispiel: Positives und Negatives .....	137
Tiefer eintauchen: Vektoren .....	138
Embedding in PyTorch .....	139
Embeddings für die Erstellung eines Sarkasmus-Detektors nutzen .....	140
Überanpassung in Sprachmodellen verringern .....	143
Die Einzelteile zusammensetzen .....	155
Das Modell für die Klassifizierung eines Satzes nutzen .....	156
Embeddings visualisieren .....	158
Vortrainierte Embeddings einsetzen .....	161
Zusammenfassung .....	162
<b>7 Rekurrente neuronale Netzwerke für das Natural Language Processing .....</b>	<b>165</b>
Grundlagen der Rekurrenz .....	165
Rekurrenz für die Verarbeitung natürlicher Sprachen erweitern .....	168
Erstellung eines Text-Classifiers mit RNNs .....	170
LSTMs stapeln (Stacking) .....	173
Vortrainierte Embeddings mit RNNs verwenden .....	181
Zusammenfassung .....	186
<b>8 Mit Machine Learning Texte erzeugen .....</b>	<b>187</b>
Sequenzen in Eingabesequenzen umwandeln .....	188
Das Modell erstellen .....	193
Text erzeugen .....	195
Das nächste Wort vorhersagen .....	196
Vorhersagen kombinieren, um Text zu erzeugen .....	197
Den Datensatz erweitern .....	200
Die Modellarchitektur verbessern .....	202
Embedding-Dimensionen .....	202
Die LSTMs initialisieren .....	202
Variable Lernrate .....	203
Die Daten verbessern .....	204
Zeichenbasierte Encodierung .....	206
Zusammenfassung .....	208

<b>9</b>	<b>Sequenz- und Zeitreihendaten verstehen</b> .....	<b>209</b>
	Häufige Attribute von Zeitreihen .....	210
	Trend .....	210
	Saisonalität .....	211
	Autokorrelation .....	211
	Rauschen .....	212
	Techniken für die Vorhersage von Zeitreihen .....	213
	Naive Vorhersage zur Ermittlung einer Grundlinie .....	213
	Vorhersagegenauigkeit messen .....	215
	Weniger naive Vorhersagen: Verwendung eines gleitenden Mittelwerts (Moving Average) .....	215
	Verbesserung der Moving-Average-Analyse .....	216
	Zusammenfassung .....	217
<b>10</b>	<b>Erstellung von ML-Modellen für die Vorhersage von Sequenzen</b> .....	<b>219</b>
	Ein »Windowed Dataset« erzeugen .....	220
	Eine »Sliding Window«-Version des Zeitreihendatensatzes erstellen .....	223
	Erstellung und Training eines DNN für die Vorhersage aus Sequenzdaten .....	226
	Die Ergebnisse des DNN auswerten .....	228
	Die Lernrate anpassen .....	231
	Zusammenfassung .....	231
<b>11</b>	<b>Faltungsbasierte und rekurrente Verfahren für Sequenzmodelle</b> .....	<b>233</b>
	Faltung für Sequenzdaten .....	233
	Faltungen programmieren .....	234
	Mit den Conv1-D-Hyperparametern experimentieren .....	239
	NASA-Wetterdaten verwenden .....	242
	GIS-Daten in Python importieren .....	243
	Verwendung von RNNs für die Sequenzmodellierung .....	246
	Einen größeren Datensatz untersuchen .....	248
	Andere rekurrente Verfahren einsetzen .....	251
	Dropouts verwenden .....	252
	Bidirektionale RNNs verwenden .....	254
	Zusammenfassung .....	256

## Teil II Generative KI einsetzen

---

<b>12</b>	<b>Konzepte der Inferenz</b> .....	<b>259</b>
	Tensoren .....	259
	Bilddaten .....	260
	Textdaten .....	262
	Tensoren als Ausgabe eines Modells .....	264
	Zusammenfassung .....	266

<b>13</b>	<b>PyTorch-Modelle für den produktiven Betrieb bereitstellen</b>	<b>267</b>
	Einführung in TorchServe	268
	TorchServe einrichten	270
	Die Umgebung vorbereiten	270
	Die config.properties-Datei anlegen	270
	Das Modell definieren	271
	Die Handler-Datei anlegen	272
	Das Modellarchiv anlegen	274
	Den Server starten	275
	Inferenz testen	277
	Weitere Schritte	279
	Flask als Server nutzen	279
	Eine Umgebung für Flask einrichten	280
	Einen Flask-Server in Python erstellen	280
	Zusammenfassung	281
<b>14</b>	<b>Modelle von Drittanbietern und zentrale Modellverzeichnisse</b>	<b>283</b>
	Der Hugging-Face-Hub	284
	Den Hugging-Face-Hub benutzen	285
	Ein Modell von Hugging Face nutzen	290
	Der PyTorch-Hub	292
	Die PyTorch-Vision-Modelle verwenden	292
	Verarbeitung natürlicher Sprachen (NLP)	295
	Andere Modelle	295
	Zusammenfassung	295
<b>15</b>	<b>Transformer und Transformers</b>	<b>297</b>
	Das Konzept der Transformer verstehen	297
	Encoder-Architekturen	298
	Die Decoder-Architektur	305
	Die Encoder-Decoder-Architektur	311
	Die Transformers-API	313
	Einstieg in Transformers	314
	Grundkonzepte	315
	Pipelines	315
	Tokenizer	317
	Zusammenfassung	321
<b>16</b>	<b>LLMs mit eigenen Daten verwenden</b>	<b>323</b>
	Feintuning eines LLM	323
	Einrichtung und Abhängigkeiten	324
	Die Daten laden und untersuchen	325
	Modell und Tokenizer initialisieren	325

Preprocessing der Daten .....	326
Die Daten zusammenführen .....	326
Metriken definieren .....	327
Das Training konfigurieren .....	327
Den Trainer initialisieren .....	328
Training und Auswertung .....	329
Das Modell speichern und testen .....	330
Ein LLM per Prompt-Tuning optimieren .....	331
Die Daten vorbereiten .....	332
Die DataLoader anlegen .....	333
Das Modell definieren .....	333
Das Modell trainieren .....	336
Auswertung während des Trainings .....	338
Trainingskennzahlen ausgeben .....	339
Die Prompt-Embeddings speichern .....	340
Inferenz mit dem Modell durchführen .....	340
Zusammenfassung .....	343
<b>17 LLMs mit Ollama bereitstellen .....</b>	<b>345</b>
Installation und Einstieg in die Arbeit mit Ollama .....	346
Ollama als Server betreiben .....	349
Applikationsentwicklung mit einem Ollama-LLM .....	351
Das Szenario .....	352
Ein Python-Skript als Machbarkeitsstudie .....	353
Eine Web-App für Ollama erstellen .....	356
Die Datei app.js .....	357
Die Datei index.html .....	359
Zusammenfassung .....	360
<b>18 Einführung in RAG .....</b>	<b>363</b>
Was ist RAG? .....	365
Einstieg in die Arbeit mit RAG .....	366
Ähnlichkeit verstehen .....	367
Die Datenbank anlegen .....	368
Eine Ähnlichkeitssuche durchführen .....	370
Die Einzelteile zusammensetzen .....	371
RAG-Inhalte mit einem LLM nutzen .....	372
Gehostete Modelle nutzen .....	376
Zusammenfassung .....	377

<b>19 Einsatz generativer Modelle mit Hugging-Face-Diffusers</b> .....	<b>379</b>
Was sind Diffusion-Modelle? .....	379
Die Hugging-Face-Diffusers-Bibliothek .....	382
Bild-zu-Bild mit Diffusers .....	385
Inpainting mit Diffusers .....	387
Zusammenfassung .....	390
<b>20 Generative Bildmodelle mit LoRA und Diffusers optimieren</b> .....	<b>391</b>
LoRA mit Diffusers trainieren .....	392
Diffusers laden .....	392
Daten zur Feinabstimmung von LoRA laden .....	393
Ein Modell mit Diffusers optimieren .....	396
Das Modell veröffentlichen .....	398
Mit optimierter LoRA ein Bild erzeugen .....	400
Zusammenfassung .....	403
<b>Index</b> .....	<b>405</b>