

<b>Vorwort</b> .....	<b>XI</b>
<b>1 Einstieg in PyTorch</b> .....	<b>1</b>
Zusammenbau eines maßgeschneiderten Deep-Learning-Rechners .....	1
Grafikprozessor (GPU) .....	2
Hauptprozessor (CPU) und Motherboard .....	2
Arbeitsspeicher (RAM) .....	2
Speicher .....	3
Deep Learning in der Cloud .....	3
Google Colaboratory .....	4
Cloud-Anbieter .....	5
Welchen Cloud-Anbieter sollte ich wählen? .....	8
Verwendung von Jupyter Notebook .....	8
PyTorch selbst installieren .....	9
CUDA downloaden .....	9
Anaconda .....	10
Zu guter Letzt – PyTorch (und Jupyter Notebook) .....	10
Tensoren .....	11
Tensoroperationen .....	12
Tensor-Broadcasting .....	14
Zusammenfassung .....	15
Weiterführende Literatur .....	15
<b>2 Bildklassifizierung mit PyTorch</b> .....	<b>17</b>
Unsere Klassifizierungsaufgabe .....	17
Traditionelle Herausforderungen .....	18
Zunächst erst mal Daten .....	19
Daten mit PyTorch einspielen .....	20
Einen Trainingsdatensatz erstellen .....	21
Erstellen eines Validierungs- und eines Testdatensatzes .....	22

Endlich, ein neuronales Netzwerk! . . . . .	24
Aktivierungsfunktionen . . . . .	25
Ein Netzwerk erstellen . . . . .	25
Verlustfunktionen . . . . .	26
Optimierung. . . . .	27
Training . . . . .	29
Validierung. . . . .	30
Ein Modell auf der GPU zum Laufen bringen . . . . .	31
Alles in einem . . . . .	32
Vorhersagen treffen . . . . .	34
Speichern von Modellen. . . . .	35
Zusammenfassung . . . . .	36
Weiterführende Literatur . . . . .	37
<b>3 Neuronale Konvolutionsnetze (CNNs) . . . . .</b>	<b>39</b>
Unser erstes Konvolutionsnetz . . . . .	39
Konvolutionen . . . . .	40
Pooling . . . . .	43
Die Dropout-Schicht . . . . .	45
Die Geschichte der CNN-Architekturen. . . . .	45
AlexNet . . . . .	46
Inception/GoogLeNet . . . . .	46
VGG . . . . .	47
ResNet . . . . .	49
Weitere Architekturen . . . . .	49
Vortrainierte Modelle in PyTorch nutzen . . . . .	50
Die Struktur eines Modells untersuchen . . . . .	51
Die Batch-Normalisierungs-Schicht. . . . .	53
Welches Modell sollten Sie verwenden? . . . . .	54
One-Stop-Shopping für Modelle: PyTorch Hub. . . . .	54
Zusammenfassung . . . . .	55
Weiterführende Literatur . . . . .	55
<b>4 Transfer Learning und andere Kniffe . . . . .</b>	<b>57</b>
Transfer Learning mit ResNet. . . . .	57
Die optimale Lernrate finden . . . . .	59
Differenzielle Lernraten . . . . .	63
Datenaugmentation . . . . .	64
Transformationen in Torchvision . . . . .	66
Farbräume und Lambda-Transformationen . . . . .	71
Benutzerdefinierte Transformationsklassen. . . . .	72
Klein anfangen und schrittweise vergrößern!. . . . .	73

Ensemble-Modelle . . . . .	74
Zusammenfassung . . . . .	75
Weiterführende Literatur . . . . .	75
<b>5 Textklassifizierung . . . . .</b>	<b>77</b>
Rekurrente neuronale Netzwerke . . . . .	77
Long-Short-Term-Memory-(LSTM-)Netzwerke . . . . .	79
Gated Recurrent Units (GRUs) . . . . .	80
BiLSTM-Netzwerke . . . . .	81
Einbettungen . . . . .	82
Torchtext . . . . .	84
Ein Twitter-Datensatz . . . . .	85
Field-Objekte definieren . . . . .	87
Einen Wortschatz aufbauen . . . . .	89
Erstellung unseres Modells . . . . .	91
Die Trainingsschleife modifizieren . . . . .	92
Tweets klassifizieren . . . . .	93
Datenaugmentation . . . . .	94
Zufälliges Einfügen . . . . .	94
Zufälliges Löschen . . . . .	95
Zufälliges Austauschen . . . . .	95
Rückübersetzung . . . . .	96
Datenaugmentation und Torchtext . . . . .	97
Transfer Learning? . . . . .	98
Zusammenfassung . . . . .	98
Weiterführende Literatur . . . . .	99
<b>6 Eine Reise in die Welt der Klänge . . . . .</b>	<b>101</b>
Töne . . . . .	101
Der ESC-50-Datensatz . . . . .	102
Den Datensatz beschaffen . . . . .	102
Audiowiedergabe in Jupyter . . . . .	103
Den ESC-50-Datensatz erkunden . . . . .	103
SoX und LibROSA . . . . .	105
torchaudio . . . . .	105
Einrichten eines eigenen ESC-50-Datensatzes . . . . .	106
Ein CNN-Modell für den ESC-50-Datensatz . . . . .	108
Frequenzbereich . . . . .	111
Mel-Spektrogramme . . . . .	111
Ein neuer Datensatz . . . . .	113
Ein vortrainiertes ResNet-Modell . . . . .	117
Lernrate finden . . . . .	118

Datenaugmentation für Audiodaten . . . . .	120
Transformationen mit torchaudio . . . . .	120
SoX-Effektketten . . . . .	121
SpecAugment . . . . .	122
Weitere Experimente . . . . .	127
Zusammenfassung . . . . .	127
Weiterführende Literatur . . . . .	128
<b>7 PyTorch-Modelle debuggen . . . . .</b>	<b>129</b>
3 Uhr morgens. Wie steht es um Ihre Daten? . . . . .	129
TensorBoard . . . . .	130
TensorBoard installieren . . . . .	130
Daten an TensorBoard übermitteln . . . . .	131
Hooks in PyTorch . . . . .	134
Mittelwert und Standardabweichung visualisieren . . . . .	135
Class Activation Mapping . . . . .	137
Flammendiagramme . . . . .	140
py-spy installieren . . . . .	143
Flammendiagramme interpretieren . . . . .	143
Eine langsame Transformation beheben . . . . .	145
Debuggen von GPU-Problemen . . . . .	149
Die GPU überwachen. . . . .	149
Gradient-Checkpointing . . . . .	151
Zusammenfassung . . . . .	153
Weiterführende Literatur . . . . .	153
<b>8 PyTorch im Produktiveinsatz . . . . .</b>	<b>155</b>
Bereitstellen eines Modells . . . . .	155
Einrichten eines Flask-Webdiensts . . . . .	156
Modellparameter laden . . . . .	159
Erstellen eines Docker-Containers . . . . .	160
Unterschiede zwischen lokalem und Cloud-Speicher . . . . .	163
Logging und Telemetrie . . . . .	165
Deployment mit Kubernetes . . . . .	166
Einrichten der Google Kubernetes Engine . . . . .	166
Aufsetzen eines Kubernetes-Clusters . . . . .	167
Dienste skalieren . . . . .	168
Aktualisierungen und Bereinigungen . . . . .	169
TorchScript . . . . .	169
Tracing . . . . .	170
Scripting . . . . .	172
Einschränkungen in TorchScript . . . . .	174

Mit libTorch arbeiten . . . . .	175
libTorch einrichten . . . . .	176
Ein TorchScript-Modell importieren . . . . .	177
Quantisierung . . . . .	179
Dynamische Quantisierung . . . . .	182
Weitere Quantisierungsmöglichkeiten . . . . .	183
Lohnt sich das alles? . . . . .	184
Zusammenfassung . . . . .	184
Weiterführende Literatur. . . . .	185
<b>9 Praxiserprobte PyTorch-Modelle in Aktion . . . . .</b>	<b>187</b>
Datenaugmentation: Vermischen und Glätten . . . . .	187
Mixup . . . . .	187
Label-Glättung . . . . .	192
Computer, einmal in scharf bitte! . . . . .	193
Einführung in die Super-Resolution . . . . .	194
Einführung in Generative Adversarial Networks (GANs) . . . . .	196
Der Fälscher und sein Kritiker . . . . .	197
Trainieren eines GAN . . . . .	197
Die Gefahr des Mode Collapse . . . . .	199
ESRGAN. . . . .	200
Weitere Einblicke in die Bilderkennung . . . . .	200
Objekterkennung . . . . .	201
Faster R-CNN und Mask R-CNN . . . . .	203
Adversarial Samples. . . . .	205
Black-Box-Angriffe . . . . .	208
Abwehr adversarialer Angriffe . . . . .	208
Die Transformer-Architektur. . . . .	209
Aufmerksamkeitsmechanismus . . . . .	209
Attention Is All You Need. . . . .	210
BERT. . . . .	211
FastBERT . . . . .	212
GPT-2 . . . . .	214
GPT-2 vorbereiten . . . . .	215
Texte mit GPT-2 erzeugen . . . . .	220
Beispielhafte Ausgabe . . . . .	222
ULMFiT . . . . .	223
Welches Modell verwenden? . . . . .	225
Selbstüberwachtes Training mit PyTorch Lightning auf Basis von Bildern . . . . .	226
Rekonstruieren und Erweitern der Eingabe . . . . .	226
Daten automatisch labeln. . . . .	228

PyTorch Lightning .....	229
Der Imagenette-Datensatz .....	230
Einen selbstüberwachten Datensatz erstellen .....	230
Ein Modell mit PyTorch Lightning erstellen .....	231
Weitere Möglichkeiten zur Selbstüberwachung (und darüber hinaus) .....	235
Zusammenfassung .....	235
Weiterführende Literatur .....	236
<b>Index</b> .....	<b>239</b>