

<b>Vorwort</b> .....	<b>9</b>
<b>1 Eine Einführung in PyTorch</b> .....	<b>15</b>
Was ist PyTorch? .....	15
Warum PyTorch verwenden? .....	16
Erste Schritte .....	18
Ausführen in Google Colaboratory .....	18
Ausführen auf einem lokalen Computer .....	21
Ausführen auf Cloud-Plattformen .....	23
Die PyTorch-Umgebung überprüfen .....	24
Ein unterhaltsames Beispiel .....	25
<b>2 Tensoren</b> .....	<b>31</b>
Was ist ein Tensor? .....	31
Ein einfaches CPU-Beispiel .....	32
Ein einfaches GPU-Beispiel .....	33
Tensoren zwischen CPUs und GPUs verschieben .....	33
Tensoren erstellen .....	34
Tensorattribute .....	37
Datentypen .....	37
Tensoren aus zufälligen Stichproben erstellen .....	39
Tensoren wie andere Tensoren erstellen .....	40
Tensoroperationen .....	40
Tensoren indizieren, slicen, kombinieren und aufteilen .....	41
Tensoroperationen für die Mathematik .....	44
Automatische Differentiation (Autograd) .....	49

<b>3</b>	<b>Deep-Learning-Entwicklung mit PyTorch</b>	<b>51</b>
	Der Gesamtprozess	51
	Datenvorbereitung	54
	Laden von Daten	54
	Datentransformationen	57
	Daten auf Stapel verteilen	60
	Allgemeine Datenvorbereitung (torch.utils.data)	61
	Modellentwicklung	64
	Modellentwurf	64
	Training	76
	Validierung	82
	Testen	85
	Modellbereitstellung (Deployment)	86
	Modelle speichern	86
	In PyTorch Hub bereitstellen	87
	In der Produktion bereitstellen	88
<b>4</b>	<b>Referenzentwürfe für die Entwicklung neuronaler Netze</b>	<b>89</b>
	Bildklassifizierung mit Transfer Learning	90
	Datenverarbeitung	90
	Modellentwurf	92
	Training und Validierung	93
	Testen und bereitstellen	95
	Stimmungsanalyse mit Torchtext	96
	Datenverarbeitung	96
	Modellentwurf	100
	Training und Validierung	101
	Testen und bereitstellen	103
	Generatives Lernen – Fashion-MNIST-Bilder mit DCGAN generieren	105
	Datenverarbeitung	105
	Modellentwurf	107
	Training	109
	Testen und bereitstellen	113
<b>5</b>	<b>PyTorch anpassen</b>	<b>115</b>
	Benutzerdefinierte Schichten und Aktivierungsfunktionen	115
	Beispiel für eine benutzerdefinierte Schicht (ComplexLinear)	117
	Beispiel für eine benutzerdefinierte Aktivierungsfunktion (ComplexReLU)	120

Benutzerdefinierte Modellarchitekturen . . . . .	121
Benutzerdefinierte Verlustfunktionen . . . . .	123
Benutzerdefinierte Algorithmen für Optimierer . . . . .	125
Benutzerdefinierte Trainings-, Validierungs- und Testschleifen . . . . .	128
<b>6 PyTorch beschleunigen und optimieren . . . . .</b>	<b>131</b>
PyTorch auf einer TPU . . . . .	132
PyTorch auf mehreren GPUs (Einzelcomputer) . . . . .	134
Datenparallelverarbeitung . . . . .	135
Modellparallelverarbeitung . . . . .	138
Kombinierte Daten- und Modellparallelverarbeitung . . . . .	140
Verteiltes Training (mehrere Computer) . . . . .	142
Modelloptimierung . . . . .	143
Hyperparameter-Tuning . . . . .	143
Quantisierung . . . . .	149
Pruning . . . . .	152
<b>7 PyTorch in die Produktion überführen . . . . .</b>	<b>157</b>
Tools und Bibliotheken für die PyTorch-Bereitstellung . . . . .	158
Gemeinsames Beispielmodell . . . . .	159
Python-API . . . . .	159
TorchScript . . . . .	160
TorchServe . . . . .	163
TorchServe und das Modellarchivierungstool installieren . . . . .	164
TorchServe starten . . . . .	166
ONNX . . . . .	169
Mobile Bibliotheken . . . . .	171
Eine Flask-App bereitstellen . . . . .	172
Colab-Flask-App . . . . .	175
Bereitstellen in der Cloud mit TorchServe . . . . .	176
Schneller Start mit Docker . . . . .	177
Bereitstellen auf mobilen und Edge-Geräten . . . . .	178
iOS . . . . .	178
Android . . . . .	181
Andere Edge-Geräte . . . . .	183
<b>8 Das PyTorch-Ökosystem und zusätzliche Ressourcen . . . . .</b>	<b>187</b>
Das PyTorch-Ökosystem . . . . .	188
Torchvision für Bild- und Videodaten . . . . .	194
Datensätze und Ein-/Ausgabe . . . . .	194
Modelle . . . . .	196
Transformationen, Operationen und Utilitys . . . . .	197

Torchtext für NLP .....	202
Ein Datensatzobjekt erstellen.....	203
Daten vorverarbeiten .....	203
Einen Datenlader für die Stapelverarbeitung erstellen.....	204
Daten (torchtext.data) .....	205
Datensätze (torchtext.datasets) .....	205
Vokabulare (torchtext.vocab) .....	207
TensorBoard für die Visualisierung .....	208
Lernkurven mit SCALARS .....	211
Modellarchitekturen mit GRAPHS .....	212
Daten mit IMAGES, TEXT und PROJECTOR .....	212
Gewichtsverteilungen mit DISTRIBUTIONS und HISTOGRAMS .....	213
Hyperparameter mit HPARAMS .....	214
Die TensorBoard-API.....	215
Papers with Code.....	217
Zusätzliche PyTorch-Ressourcen .....	218
Tutorials.....	218
Bücher .....	219
Onlinekurse und Live-Schulungen.....	220
<b>Index.....</b>	<b>223</b>