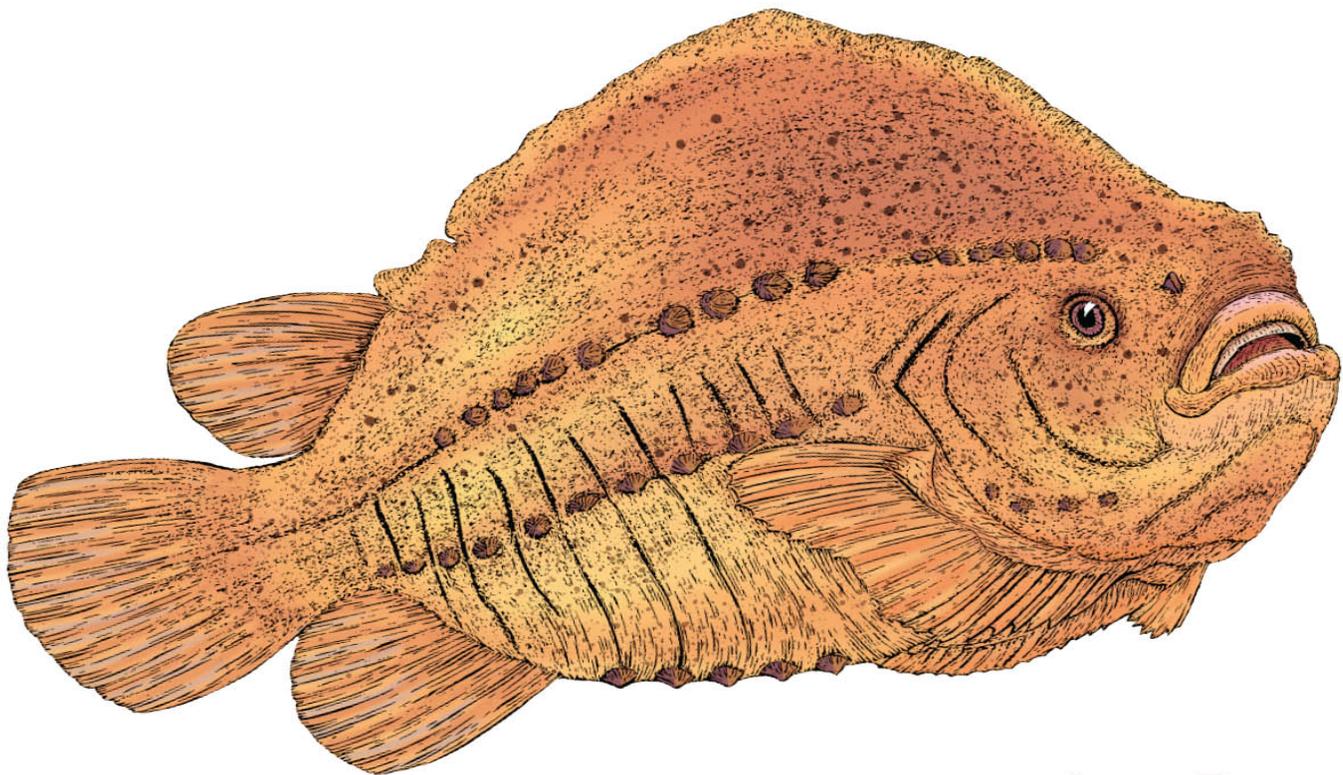


O'REILLY®

Mit PyTorch-
Schnelleinstieg

PyTorch kompakt

Syntax, Design Patterns und Codebeispiele
für Deep-Learning-Modelle



Joe Papa

Übersetzung von Frank Langenau

Papier
plus⁺
PDF.

Zu diesem Buch – sowie zu vielen weiteren O’Reilly-Büchern – können Sie auch das entsprechende E-Book im PDF-Format herunterladen. Werden Sie dazu einfach Mitglied bei oreilly.plus⁺:

www.oreilly.plus

PyTorch kompakt

Syntax, Design Patterns und Codebeispiele für Deep-Learning-Modelle

Joe Papa

Deutsche Übersetzung von Frank Langenau

O'REILLY®

Joe Papa

Lektorat: Alexandra Follenius

Übersetzung: Frank Langenau

Korrektorat: Sibylle Feldmann, www.richtiger-text.de

Satz: III-satz, www.drei-satz.de

Herstellung: Stefanie Weidner

Umschlaggestaltung: Karen Montgomery, Michael Oréal, www.oreal.de

Bibliografische Information der Deutschen Nationalbibliothek

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <http://dnb.d-nb.de> abrufbar.

ISBN:

Print 978-3-96009-185-1

PDF 978-3-96010-600-5

ePub 978-3-96010-601-2

mobi 978-3-96010-602-9

1. Auflage 2022

Translation Copyright für die deutschsprachige Ausgabe © 2022 dpunkt.verlag GmbH

Wieblinger Weg 17

69123 Heidelberg

Authorized German translation of the English edition of *PyTorch Pocket Reference*, ISBN 9781492090007

© 2021 Mobile Insights Technology Group, LLC. This translation is published and sold by permission of O'Reilly Media, Inc., which owns or controls all rights to publish and sell the same.

Dieses Buch erscheint in Kooperation mit O'Reilly Media, Inc. unter dem Imprint »O'REILLY«.

O'REILLY ist ein Markenzeichen und eine eingetragene Marke von O'Reilly Media, Inc. und wird mit Einwilligung des Eigentümers verwendet.

Hinweis:

Dieses Buch wurde auf PEFC-zertifiziertem Papier aus nachhaltiger Waldwirtschaft gedruckt. Der Umwelt zuliebe verzichten wir zusätzlich auf die Einschweißfolie.



Schreiben Sie uns:

Falls Sie Anregungen, Wünsche und Kommentare haben, lassen Sie es uns wissen: komentar@oreilly.de.

Die vorliegende Publikation ist urheberrechtlich geschützt. Alle Rechte vorbehalten. Die Verwendung der Texte und Abbildungen, auch auszugsweise, ist ohne die schriftliche Zustimmung des Verlags urheberrechtswidrig und daher strafbar. Dies gilt insbesondere für die Vervielfältigung, Übersetzung oder die Verwendung in elektronischen Systemen.

Es wird darauf hingewiesen, dass die im Buch verwendeten Soft- und Hardware-Bezeichnungen sowie Markennamen und Produktbezeichnungen der jeweiligen Firmen im Allgemeinen warenzeichen-, marken- oder patentrechtlichem Schutz unterliegen.

Alle Angaben und Programme in diesem Buch wurden mit größter Sorgfalt kontrolliert. Weder Autor noch Verlag noch Übersetzer können jedoch für Schäden haftbar gemacht werden, die in Zusammenhang mit der Verwendung dieses Buches stehen.

5 4 3 2 1 0

Inhalt

Vorwort

1 Eine Einführung in PyTorch

Was ist PyTorch?

Warum PyTorch verwenden?

Erste Schritte

- Ausführen in Google Colaboratory

- Ausführen auf einem lokalen Computer

- Ausführen auf Cloud-Plattformen

- Die PyTorch-Umgebung überprüfen

Ein unterhaltsames Beispiel

2 Tensoren

Was ist ein Tensor?

- Ein einfaches CPU-Beispiel

- Ein einfaches GPU-Beispiel

- Tensoren zwischen CPUs und GPUs verschieben

Tensoren erstellen

- Tensorattribute

- Datentypen

- Tensoren aus zufälligen Stichproben erstellen

- Tensoren wie andere Tensoren erstellen

Tensoroperationen

- Tensoren indizieren, slicen, kombinieren und aufteilen

Tensoroperationen für die Mathematik
Automatische Differentiation (Autograd)

3 Deep-Learning-Entwicklung mit PyTorch

Der Gesamtprozess

Datenvorbereitung

Laden von Daten

Datentransformationen

Daten auf Stapel verteilen

Allgemeine Datenvorbereitung (`torch.utils.data`)

Modellentwicklung

Modellentwurf

Training

Validierung

Testen

Modellbereitstellung (Deployment)

Modelle speichern

In PyTorch Hub bereitstellen

In der Produktion bereitstellen

4 Referenzentwürfe für die Entwicklung neuronaler Netze

Bildklassifizierung mit Transfer Learning

Datenverarbeitung

Modellentwurf

Training und Validierung

Testen und bereitstellen

Stimmungsanalyse mit Torchtext

Datenverarbeitung

Modellentwurf

Training und Validierung

Testen und bereitstellen

Generatives Lernen - Fashion-MNIST-Bilder mit DCGAN generieren

Datenverarbeitung

Modellentwurf
Training
Testen und bereitstellen

5 PyTorch anpassen

Benutzerdefinierte Schichten und
Aktivierungsfunktionen

Beispiel für eine benutzerdefinierte Schicht
(ComplexLinear)

Beispiel für eine benutzerdefinierte
Aktivierungsfunktion (ComplexReLU)

Benutzerdefinierte Modellarchitekturen

Benutzerdefinierte Verlustfunktionen

Benutzerdefinierte Algorithmen für Optimierer

Benutzerdefinierte Trainings-, Validierungs- und
Testschleifen

6 PyTorch beschleunigen und optimieren

PyTorch auf einer TPU

PyTorch auf mehreren GPUs (Einzelcomputer)

Datenparallelverarbeitung

Modellparallelverarbeitung

Kombinierte Daten- und Modellparallelverarbeitung

Verteiltes Training (mehrere Computer)

Modelloptimierung

Hyperparameter-Tuning

Quantisierung

Pruning

7 PyTorch in die Produktion überführen

Tools und Bibliotheken für die PyTorch-Bereitstellung

Gemeinsames Beispielmmodell

Python-API

TorchScript

TorchServe

- TorchServe und das Modellarchivierungstool installieren
- TorchServe starten
- ONNX
- Mobile Bibliotheken
- Eine Flask-App bereitstellen
- Colab-Flask-App
- Bereitstellen in der Cloud mit TorchServe
- Schneller Start mit Docker
- Bereitstellen auf mobilen und Edge-Geräten
 - iOS
 - Android
 - Andere Edge-Geräte

8 Das PyTorch-Ökosystem und zusätzliche Ressourcen

- Das PyTorch-Ökosystem
- Torchvision für Bild- und Videodaten
 - Datensätze und Ein-/Ausgabe
 - Modelle
 - Transformationen, Operationen und Utilitys
- Torchtext für NLP
 - Ein Datensatzobjekt erstellen
 - Daten vorverarbeiten
 - Einen Datenlader für die Stapelverarbeitung erstellen
 - Daten (torchtext.data)
 - Datensätze (torchtext.datasets)
 - Vokabulare (torchtext.vocab)
- TensorBoard für die Visualisierung
 - Lernkurven mit SCALARS
 - Modellarchitekturen mit GRAPHS
 - Daten mit IMAGES, TEXT und PROJECTOR
 - Gewichtsverteilungen mit DISTRIBUTIONS und HISTOGRAMS

[Hyperparameter mit HPARAMS](#)
[Die TensorBoard-API](#)
[Papers with Code](#)
[Zusätzliche PyTorch-Ressourcen](#)
[Tutorials](#)
[Bücher](#)
[Onlinekurse und Live-Schulungen](#)

Index

Vorwort

Wir leben in aufregenden Zeiten! Einige von uns hatten das Glück, große technologische Fortschritte mitzuerleben – die Erfindung des Personal Computers, die Anfänge des Internets, die Verbreitung von Mobiltelefonen und das Aufkommen der sozialen Medien. Und jetzt gibt es große Durchbrüche in der KI!

Es ist spannend, diesen Wandel zu beobachten und an ihm teilzuhaben. Ich glaube, wir stehen erst am Anfang, und es ist toll, daran zu denken, wie sich die Welt in den nächsten zehn Jahren verändern könnte. Wie großartig ist es, dass wir in diesen Zeiten leben und an der Entwicklung der KI teilhaben können!

Zweifellos hat PyTorch einige der wichtigsten Fortschritte in den Bereichen Deep Learning und KI ermöglicht. Die Bibliothek kann kostenlos heruntergeladen und verwendet werden, und jeder, der über einen Computer oder eine Internetverbindung verfügt, kann KI-Experimente durchführen. Neben umfassenderen Referenzen wie dieser gibt es viele kostenlose und preiswerte Schulungen, Blogartikel und Tutorials, die Ihnen helfen können. Jeder kann sich mit PyTorch in maschinelles Lernen und KI einarbeiten.

Für wen dieses Buch gedacht ist

Dieses Buch richtet sich sowohl an Einsteiger:innen als auch an fortgeschrittene Benutzer:innen, die sich für maschinelles Lernen und KI interessieren. Es ist hilfreich, wenn Sie bereits Erfahrungen darin gesammelt haben, wie man Python-Code schreibt, und wenn Sie verstehen, worum es bei Data Science und maschinellem Lernen prinzipiell geht.

Wenn Sie sich dem maschinellen Lernen gerade erst zuwenden, unterstützt Sie dieses Buch dabei, die Grundlagen von PyTorch zu erlernen. Außerdem werden einige einfache Beispiele vorgestellt. Sollten Sie bereits mit einem anderen Framework wie zum Beispiel TensorFlow, Caffe2 oder MXNet arbeiten, macht das Buch Sie mit der PyTorch-API und der entsprechenden Programmierweise vertraut, sodass Sie Ihre Fähigkeiten erweitern können.

Falls Sie PyTorch schon eine Weile verwenden, hilft Ihnen dieses Buch, Ihr Wissen zu anspruchsvolleren Themen wie Beschleunigung und Optimierung zu erweitern. Zudem dient es Ihnen als Schnellreferenz, während Sie PyTorch für Ihre täglichen Entwicklungsarbeiten einsetzen.

Warum ich dieses Buch geschrieben habe

Es kann sehr spannend sein, PyTorch zu erlernen und zu beherrschen. Es gibt so viel zu entdecken! Als ich anfing, mich mit PyTorch zu beschäftigen, wünschte ich mir eine einzige Ressource, die mir alles beibringen würde. Ich wollte etwas haben, das mir einen guten Überblick über die Möglichkeiten von PyTorch verschafft, aber auch Beispiele

und genügend Details bietet, wenn ich tiefer eindringen müsste.

Es gibt einige wirklich gute Bücher und Kurse zu PyTorch, doch sie konzentrieren sich oftmals auf Tensoren und das Training für Deep-Learning-Modelle. Die Onlinedokumentation von PyTorch ist zwar ebenfalls sehr gut und bietet eine Fülle von Details und Beispielen, allerdings fand ich sie oft recht umständlich. Ständig musste ich herumklicken, um zu lernen oder zu googeln, was ich wissen musste. Ich brauchte ein Buch auf meinem Schreibtisch, das ich mit Markierungen versehen und beim Programmieren zurate ziehen konnte.

Mein Ziel ist es, dass dieses Buch die ultimative PyTorch-Referenz für Sie wird. Ich hoffe, dass Sie es nicht nur durchlesen, um ein umfassendes Verständnis der verfügbaren Ressourcen zu bekommen, sondern dass Sie sich die wichtigsten Abschnitte für Ihre Entwicklungsarbeit markieren und auf Ihrem Schreibtisch griffbereit halten. Wenn Sie etwas vergessen, können Sie die Antwort sofort nachschlagen. Sollten Sie E-Books oder Onlinebücher bevorzugen, können Sie dieses Buch online mit Lesezeichen versehen. Wie auch immer Sie es verwenden, ich hoffe, das Buch hilft Ihnen dabei, mit PyTorch eine spannende neue Technologie zu erschaffen!

Wegweiser durch das Buch

Als Neuling in PyTorch sollten Sie mit [Kapitel 1](#) beginnen und die Kapitel nacheinander lesen. Die Kapitel befassen sich zunächst mit Einsteigerthemen und gehen dann zu anspruchsvolleren Themen über. Wenn Sie bereits etwas Erfahrung mit PyTorch haben, könnten Sie auch direkt zu den Themen springen, die Sie am meisten interessieren.

Lassen Sie aber [Kapitel 8](#) zum PyTorch-Ökosystem nicht aus. Sicherlich werden Sie hier etwas Neues entdecken!

Dieses Buch ist grob wie folgt aufgebaut:

- [Kapitel 1](#) gibt eine kurze Einführung in PyTorch, hilft Ihnen beim Einrichten Ihrer Entwicklungsumgebung und bietet ein unterhaltsames Beispiel, das Sie selbst ausprobieren sollten.
- [Kapitel 2](#) befasst sich mit dem Tensor, dem grundlegenden Baustein von PyTorch. Auf ihm baut alles in PyTorch auf.
- [Kapitel 3](#) bietet Ihnen einen umfassenden Blick darauf, wie Sie PyTorch für Deep Learning einsetzen können, und [Kapitel 4](#) stellt beispielhafte Referenzkonzepte vor, sodass Sie PyTorch in Aktion sehen können.
- Die [Kapitel 5](#) und [6](#) behandeln fortgeschrittenere Themen. [Kapitel 5](#) zeigt Ihnen, wie Sie PyTorch-Komponenten für Ihre eigene Arbeit anpassen, und in [Kapitel 6](#) erfahren Sie, wie Sie das Training beschleunigen und Ihre Modelle optimieren.
- In [Kapitel 7](#) lernen Sie, wie Sie PyTorch über lokale Computer, Cloud-Server und mobile oder Edge-Geräte in die Produktion überführen können.
- [Kapitel 8](#) dient der Orientierung, um herauszufinden, in welche Richtung Sie Ihre weitere Arbeit führen kann. Es führt das PyTorch-Ökosystem ein, beschreibt bekannte Pakete und listet zusätzliche Schulungsquellen auf.

Konventionen, die in diesem Buch verwendet werden

In diesem Buch werden die folgenden typografischen Konventionen verwendet:

Kursiv

Kennzeichnet neue Begriffe, URLs, E-Mail-Adressen, Dateinamen und Dateierweiterungen.

Schreibmaschinenschrift

Wird in Programmlistings verwendet und im Fließtext für Programmelemente wie zum Beispiel Variablen- oder Funktionsnamen, Datenbanken, Datentypen, Umgebungsvariablen, Anweisungen und Schlüsselwörter.

Schreibmaschinenschrift fett

Kennzeichnet Befehle oder andere Texte, die vom Benutzer buchstäblich eingegeben werden sollen.

Schreibmaschinenschrift kursiv

Zeigt Text, der ersetzt werden soll durch Werte, die der Benutzer bereitstellt, oder Werte, die sich aus dem Kontext ergeben. Außerdem werden die in Tabellen kursiv gedruckten Transformationen derzeit von PyTorch nicht unterstützt.



Dieses Element kennzeichnet einen Tipp oder Vorschlag.



Dieses Element kennzeichnet eine allgemeine Anmerkung.



Dieses Element kennzeichnet eine Warnung.

Codebeispiele

Ergänzendes Material (Codebeispiele, Übungen usw.) stehen Ihnen unter <https://github.com/joe-papa/pytorch-book> zum Download zur Verfügung.

Dieses Buch soll Ihnen bei Ihrer täglichen Arbeit helfen. Falls Beispielcode zum Buch angeboten wird, dürfen Sie ihn im Allgemeinen in Ihren Programmen und für Dokumentationen verwenden. Sie müssen uns nicht um Erlaubnis bitten, es sei denn, Sie kopieren einen erheblichen Teil des Codes. Wenn Sie zum Beispiel ein Programm schreiben, das einige Codeblöcke aus diesem Buch verwendet, benötigen Sie keine Erlaubnis. Sollten Sie aber Beispiele aus O'Reilly-Büchern verkaufen oder verbreiten, ist eine Erlaubnis erforderlich. Wenn Sie eine Frage beantworten und dabei dieses Buch oder Beispielcode aus diesem Buch zitieren, brauchen Sie wiederum keine Erlaubnis. Aber wenn Sie erhebliche Teile des Beispielcodes aus diesem Buch in die Dokumentation Ihres Produkts einfließen lassen, ist eine Erlaubnis einzuholen.

Wir schätzen eine Quellenangabe, verlangen sie aber nicht. Eine Quellenangabe umfasst in der Regel Titel, Autor, Verlag und ISBN, zum Beispiel: »*PyTorch kompakt*« von Joe Papa (O'Reilly). Copyright 2022 dpunkt.verlag, ISBN 978-3-96009-185-1.« Wenn Sie der Meinung sind, dass Sie die Codebeispiele in einer Weise verwenden, die über die oben erteilte Erlaubnis hinausgeht, kontaktieren Sie uns bitte unter komentar@oreilly.de.

Danksagungen

Als Leser bin ich oftmals beeindruckt, wenn ich die Danksagungen von anderen Autoren lese. Ein Buch zu schreiben, ist kein Pappenstiel, und um ein gutes Buch zu schreiben, braucht man die Unterstützung von vielen Menschen. Die Danksagungen erinnern uns ständig daran, dass wir es nicht allein schaffen können.

Ich bin dankbar für die Unterstützung und die Ermutigung durch meinen Freund Matt Kirk, den ich vor Jahren auf einer O'Reilly-Konferenz kennengelernt habe. Unsere gemeinsame Leidenschaft für Persönlichkeitsentwicklung inspiriert dazu, Bücher und Kurse zu verfassen sowie anderen zu helfen, ihr volles Potenzial sowohl persönlich als auch beruflich auszuschöpfen. Wöchentliche Zoomchats und Selbsthilfeprojekte während der Pandemie haben mir zweifellos geholfen, bei Verstand zu bleiben. Ohne Matt wäre dieses Buch nicht möglich gewesen.

Bei Rebecca Novack möchte ich mich dafür bedanken, dass sie das Projekt vorgeschlagen und mir sowie dem Team bei O'Reilly die Chance eingeräumt hat, dieses Projekt zu realisieren.

Es ist nicht einfach, ein Buch zu schreiben, aber ein gutes Buch zu schreiben, erfordert engagierte Reviewer, denen die Leser am Herzen liegen. Bedanken möchte ich mich bei Mike Drob, Axel Sirota und Jeff Bleiel dafür, dass sie sich die Zeit genommen haben, das Buch durchzusehen und unzählige Vorschläge zu unterbreiten. Durch Mikes Vorschläge sind viele praktische Ressourcen hinzugekommen, die ich sonst übersehen hätte. Er hat dafür gesorgt, dass wir modernste Tools und bewährte Verfahren verwenden, die man in Onlinedokumentationen nicht findet.

Axels Detailgenauigkeit ist unglaublich. Ich bin dankbar für seine Ermutigungen und seine Bemühungen, den Code und die technischen Details in diesem Buch zu überprüfen. Jeff ist ein hervorragender Lektor. Für seine Vorschläge zum thematischen Aufbau des Buchs danke ich ihm. Er hat mir sehr geholfen, ein besserer Autor zu werden.

PyTorch ist ein echtes Gemeinschaftsprojekt. Ich bin den Mitarbeiter:innen bei Facebook und den über 1.700 Mitwirkenden, die dieses Framework für maschinelles Lernen entwickelt haben, sehr dankbar. Vor allem möchte ich denjenigen danken, die die Dokumentationen und Tutorials erstellt haben, die anderen - genau wie mir - helfen, PyTorch schnell zu erlernen.

Einige Personen, deren Arbeit mir am meisten geholfen hat, sind Suraj Subramanian, Seth Juarez, Cassie Breviu, Dmitry Soshnikov, Ari Bornstein, Soumith Chintala, Justin Johnson, Jeremy Howard, Rachel Thomas, Francisco Ingham, Sasank Chilamkurthy, Nathan Inkawhich, Sean Robertson, Ben Trevett, Avinash Sajjanshetty, James Reed, Michael Suo, Michela Paganini, Shen Li, Séb Arnold, Rohan Varma, Pritam Damania und Jeff Tang sowie die unzähligen Blogger und YouTuber zum Thema PyTorch.

Ich danke Manbir Gulati dafür, dass er mich in PyTorch eingeführt hat, und Rob Miller dafür, dass er mir die Möglichkeit gegeben hat, KI-Projekte mit PyTorch zu leiten. Auch den Gedankenaustausch zum Deep Learning für dieses Buch mit meinem Freund Isaac Privitera habe ich sehr geschätzt.

Natürlich hätte ich all das nicht erreichen können ohne die harte Arbeit und das Engagement meiner Mutter Grace, die mir und meinem Bruder aus bescheidenen Verhältnissen heraus den Weg ins Leben geebnet hat. Ich vermisse sie jeden Tag.

Ein besonderer Dank gilt meinem Bruder Vinnie, der eine große Hilfe war bei der Fertigstellung meiner Hausprojekte und mir so mehr Zeit zum Schreiben gab. Ich danke meinem Stiefvater Lou für seine Aufmunterung, während ich das Buch geschrieben habe. Danken möchte ich auch meinen Kindern Savannah, Caroline und George. Sie waren geduldig und verständnisvoll, als ihr Vater arbeiten musste.

Schließlich möchte ich meiner Frau Emily danken. Sie hat meine Ideen und Träume in meinem Leben immer unterstützt. Als ich die Aufgabe, dieses Buch zu schreiben, in Angriff genommen habe, konnte ich mich natürlich wieder auf sie verlassen. Es war schon eine große Herausforderung, sich fast im Alleingang um unsere drei Kinder zu kümmern und während der Pandemie noch neue Aufgaben zu übernehmen.

Dennoch war sie die Unterstützung, die ich brauchte, um mein Buch fertigzustellen. Und während ich am Buch schrieb, erfuhren wir, dass sie schwanger ist und wir unser viertes Kind erwarten! Meine Frau macht das alles mit einem Lächeln und einem Scherz (der oft auf meine Kosten geht), und dafür liebe ich sie.

Eine Einführung in PyTorch

PyTorch ist eine der beliebtesten Python-Bibliotheken für Deep Learning und wird von der KI-Forschungsgemeinde häufig eingesetzt. Viele Entwickler:innen und Forscher:innen verwenden PyTorch, um Experimente und Prototypenerstellung im Bereich Deep Learning zu beschleunigen.

Dieses Kapitel gibt Ihnen eine kurze Einführung in das Wesen von PyTorch und in einige der Features, die es so beliebt machen. Hier erfahren Sie auch, wie Sie Ihre PyTorch-Entwicklungsumgebung auf Ihrem lokalen Computer und in der Cloud installieren und einrichten. Am Ende dieses Kapitels können Sie dann schon überprüfen, ob PyTorch richtig installiert ist, und ein einfaches PyTorch-Programm ausführen.

Was ist PyTorch?

Die PyTorch-Bibliothek wird hauptsächlich vom *AI Research Lab von Facebook* (FAIR) entwickelt und ist eine

frei verfügbare Open-Source-Software mit über 1.700 Mitwirkenden. Damit lassen sich in Python mit starker GPU-Beschleunigung Array-basierte Berechnungen ganz einfach durchführen, dynamische neuronale Netze erstellen und automatisches Differenzieren realisieren - alles Features, die für die Deep-Learning-Forschung unabdingbar sind. Obwohl einige die Bibliothek für beschleunigte Tensorberechnungen heranziehen, nutzt man sie in der Regel für die Deep-Learning-Entwicklung.

Die einfache und flexible Schnittstelle von PyTorch erlaubt schnelles Experimentieren. Mit nur wenigen Codezeilen lassen sich Daten laden, Transformationen anwenden und Modelle erstellen. Zudem können Sie mühelos und flexibel benutzerdefinierte Trainings-, Validierungs- und Testschleifen schreiben sowie trainierte Modelle bereitstellen.

PyTorch verfügt über ein starkes Ökosystem und eine große Nutzergemeinde, darunter Universitäten wie Stanford und Firmen wie Uber, NVIDIA und Salesforce. Im Jahr 2019 dominierte PyTorch die Konferenzberichte über maschinelles Lernen und Deep Learning: 69% der Konferenzberichte der *Conference on Computer Vision and Pattern Recognition* (CVPR) verwendeten PyTorch, über 75% der Konferenzberichte der *Association for Computational Linguistics* (ACL) und des *North American Chapter of the ACL* (NAACL) arbeiteten damit, und über 50% der Konferenzberichte der *International Conference on Learning Representations* (ICLR) und der *International Conference on Machine Learning* (ICML) verwendeten PyTorch ebenfalls. Es gibt über 60.000 Repositorys auf GitHub, die sich auf PyTorch beziehen.

Viele Entwickler:innen und Forscher:innen nutzen PyTorch, um das Experimentieren und die Prototypenerstellung im

Bereich Deep Learning zu beschleunigen. Die einfache Python-API, die GPU-Unterstützung und die Flexibilität machen diese Bibliothek zu einer beliebten Wahl bei akademischen und kommerziellen Forschungseinrichtungen. Seit seiner Veröffentlichung als Open-Source-Software im Jahr 2018 hat PyTorch inzwischen eine stabile Version erreicht und kann problemlos auf Windows-, Mac- und Linux-Betriebssystemen installiert werden. Das Framework wird weiterhin schnell erweitert und ermöglicht nun die Bereitstellung in Produktionsumgebungen in der Cloud und auf mobilen Plattformen.

Warum PyTorch verwenden?

Wenn Sie maschinelles Lernen studieren, Deep-Learning-Forschung betreiben oder KI-Systeme entwickeln, werden Sie wahrscheinlich auf ein Framework für Deep Learning zurückgreifen müssen. Ein solches Framework für Deep Learning erweist sich besonders bei gängigen Aufgaben als hilfreich, beispielsweise um Daten zu laden und vorzuerarbeiten oder Modelle zu entwerfen, zu trainieren und bereitzustellen. PyTorch ist gerade wegen seiner Einfachheit, Flexibilität und der Python-Schnittstelle bei akademischen und Forschungsteams sehr beliebt geworden. Es gibt viele Gründe, PyTorch zu lernen und zu verwenden, zum Beispiel:

PyTorch ist beliebt.

Viele Unternehmen und Forschungseinrichtungen verwenden PyTorch als ihr wichtigstes Deep-Learning-Framework. Einige Unternehmen setzen sogar auf PyTorch auf, um ihre speziellen Tools für maschinelles

Lernen zu entwickeln. Infolgedessen sind PyTorch-Kenntnisse sehr gefragt.

PyTorch wird von allen großen Cloud-Plattformen unterstützt, beispielsweise von Amazon Web Services (AWS), Google Cloud Platform (GCP), Microsoft Azure und Alibaba Cloud.

Für eine reibungslose Entwicklung können Sie einen virtuellen Computer mit vorinstalliertem PyTorch aufsetzen. Es ist möglich, vorgefertigte Docker-Images zu verwenden, groß angelegte Trainings auf GPU-Plattformen in der Cloud durchzuführen und Modelle im Produktionsmaßstab auszuführen.

PyTorch wird von Google Colaboratory und Kaggle-Kerneln unterstützt.

PyTorch-Code lässt sich in einem Browser ausführen, ohne dass eine Installation oder Konfiguration erforderlich ist. An Kaggle-Wettbewerben können Sie teilnehmen, indem Sie PyTorch direkt in Ihrem Kernel ausführen.

PyTorch ist ausgereift und stabil.

PyTorch wird regelmäßig gepflegt und ist seit Juni 2021 in der Version 1.9 verfügbar.

PyTorch unterstützt CPU-, GPU-, TPU- und Parallelverarbeitung.

Training und Inferenz lassen sich mit GPUs und TPUs beschleunigen. TPUs (*Tensor Processing Units*) sind anwendungsspezifische integrierte Schaltungen (ASICs), die darauf ausgelegt sind, KI-Anwendungen zu beschleunigen. Diese Chips wurden von Google entwickelt, um eine Alternative zu GPUs für die Hardwarebeschleunigung von neuronalen Netzen zu bieten. Mit Parallelverarbeitung können Sie

Vorverarbeitungsschritte auf Ihrer CPU realisieren, während ein Modell auf der GPU oder der TPU trainiert wird.

PyTorch unterstützt verteiltes Training.

Neuronale Netze lassen sich über mehrere GPUs auf mehreren Computern trainieren.

PyTorch unterstützt die Überführung in die Produktion.

Mit den neueren TorchScript- und TorchServe-Features können Sie Modelle ganz leicht in Produktionsumgebungen – Cloud-Server eingeschlossen – überführen.

PyTorch unterstützt zunehmend die mobile Bereitstellung.

Obwohl dieses Feature derzeit noch in einer experimentellen Phase ist, können Sie Modelle jetzt auf iOS- und Android-Geräten bereitstellen.

PyTorch verfügt über ein umfangreiches Ökosystem und eine Reihe von Open-Source-Bibliotheken.

Bibliotheken wie Torchvision, fastai und PyTorch Lightning erweitern die Fähigkeiten und unterstützen spezifische Gebiete wie die Verarbeitung natürlicher Sprache (*Natural Language Processing*, NLP) und Computervision.

PyTorch besitzt auch ein C++-Frontend.

In diesem Buch geht es in erster Linie um die Python-Schnittstelle. Allerdings unterstützt PyTorch auch eine C++-Schnittstelle als Frontend. Wenn Sie hochleistungsfähige Anwendungen mit niedriger Latenz oder Bare-Metal-Anwendungen erstellen müssen, können Sie sie in C++ schreiben und dabei sowohl das Design als auch die Architektur wie bei der Python-API verwenden.

PyTorch unterstützt das ONNX-Format (Open Neural Network Exchange) von Haus aus.

Ihre Modelle können Sie ganz einfach in das ONNX-Format exportieren und sie mit ONNX-kompatiblen Plattformen, Laufzeitumgebungen oder Visualisierern verwenden.

PyTorch verfügt über eine große Gemeinschaft von Entwicklern und Benutzerforen.

Das PyTorch-Forum verzeichnet mehr als 38.000 Benutzerinnen und Benutzer, und es ist einfach, Unterstützung zu erhalten oder Fragen an die Gemeinschaft zu posten, wenn Sie das PyTorch-Diskussionsforum (<https://pytorch.tips/discuss>) besuchen.

Erste Schritte

Wenn Sie mit PyTorch vertraut sind, haben Sie dieses Framework vielleicht schon installiert und Ihre Entwicklungsumgebung eingerichtet. Falls nicht, zeige ich Ihnen in diesem Abschnitt einige Wege, wie Sie dabei vorgehen können. Am schnellsten können Sie mit Google Colaboratory (oder *Colab*) loslegen. Google Colab ist eine kostenlose Cloud-basierte Entwicklungsumgebung, die Jupyter Notebook ähnelt und bei der PyTorch bereits installiert ist. Colab bietet eine kostenlose beschränkte GPU-Unterstützung und Schnittstellen, die sich für Google Drive eignen und über die sich Notebooks speichern und teilen lassen.

Wenn Sie keinen Internetzugang haben oder den PyTorch-Code auf Ihrer eigenen Hardware ausführen möchten, erfahren Sie hier auch, wie Sie PyTorch auf einem lokalen Computer installieren. PyTorch können Sie unter den

Betriebssystemen Windows, Linux und macOS installieren. Zur Beschleunigung empfehle ich einen NVIDIA-Grafikprozessor, der aber nicht unbedingt erforderlich ist.

Schließlich kann es sein, dass Sie PyTorch-Code auf einer Cloud-Plattform wie AWS, Azure oder GCP entwickeln möchten. Wollen Sie eine Cloud-Plattform verwenden, zeige ich Ihnen die Möglichkeiten, um auf der jeweiligen Plattform schnell loslegen zu können.

Ausführen in Google Colaboratory

Mit Google Colab können Sie Python- und PyTorch-Code in Ihrem Browser schreiben und ausführen. Dateien speichern Sie direkt in Ihr Google-Drive-Konto und teilen Ihre Arbeit ganz einfach mit anderen. Besuchen Sie für den Anfang die Google-Colab-Website (<https://pytorch.tips/colab>), wie [Abbildung 1-1](#) sie zeigt.

Wenn Sie bereits bei Ihrem Google-Konto angemeldet sind, erscheint ein Pop-up-Fenster. Klicken Sie in der rechten oberen Ecke auf *Neues Notebook*. Sollte das Pop-up-Fenster nicht erscheinen, klicken Sie im Menü auf *Datei* und wählen *Neues Notebook*. Daraufhin werden Sie aufgefordert, sich bei Ihrem Google-Konto anzumelden oder ein Konto zu erstellen (siehe [Abbildung 1-2](#)).

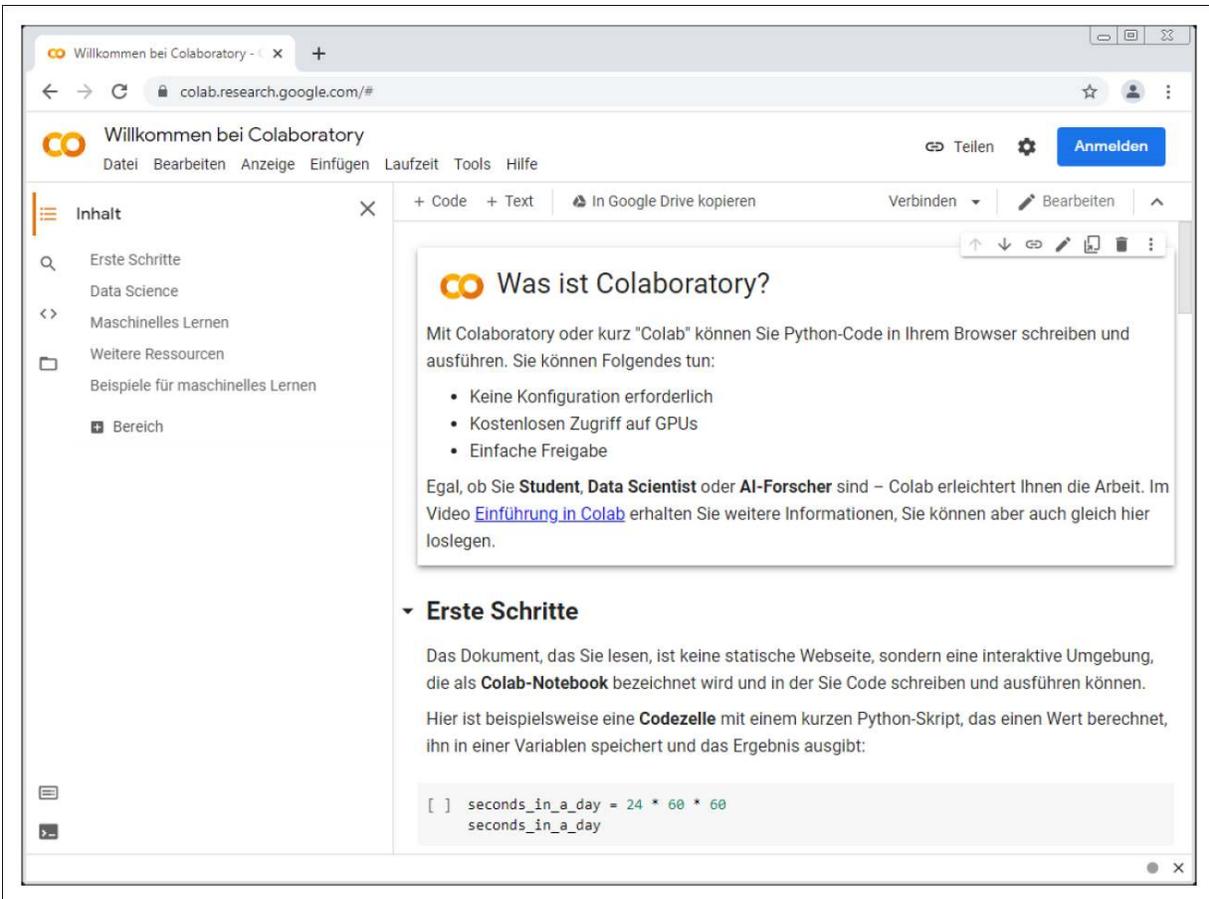


Abbildung 1-1: Begrüßungsseite von Google Colaboratory

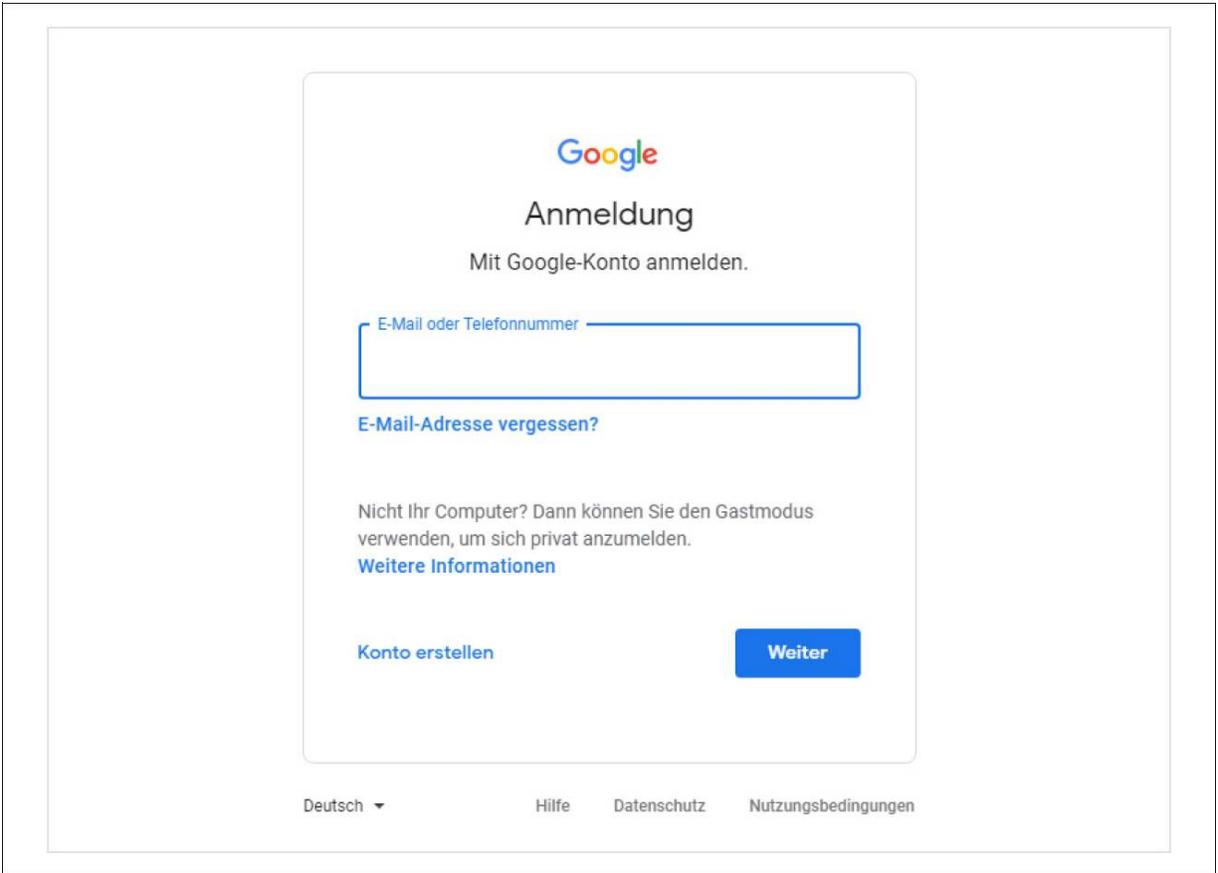


Abbildung 1-2: Anmeldeseite für Google

Um Ihre Konfiguration zu überprüfen, importieren Sie die PyTorch-Bibliothek. Lassen Sie sich die installierte Version ausgeben und kontrollieren Sie, ob Sie eine GPU verwenden, wie [Abbildung 1-3](#) zeigt.

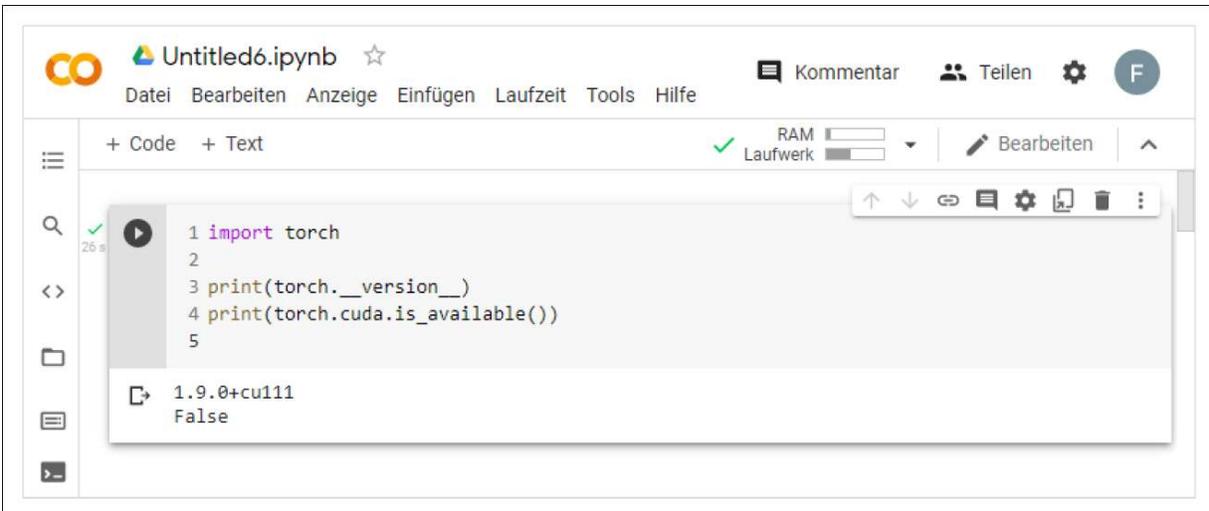


Abbildung 1-3: Die PyTorch-Installation in Google Colaboratory überprüfen

Standardmäßig verwendet unser Colab-Notebook keine GPU. Deshalb müssen Sie zuerst im Menü *Laufzeit* den Befehl *Laufzeittyp ändern* ausführen, dann im Dropdown-Menü *Hardwarebeschleuniger* den Eintrag *GPU* auswählen und auf *Speichern* klicken, wie [Abbildung 1-4](#) zeigt.

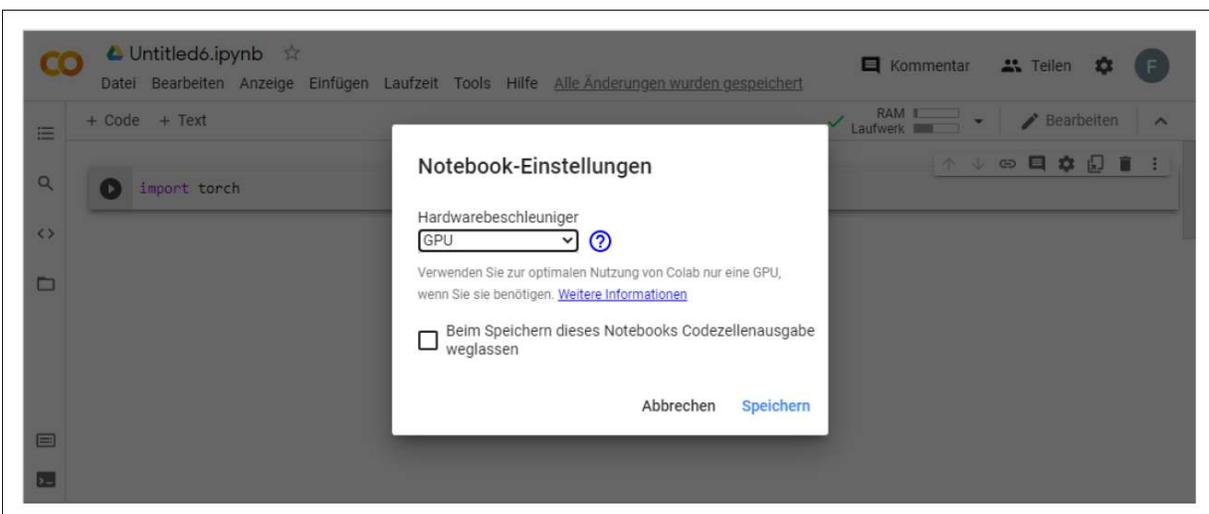


Abbildung 1-4: Eine GPU in Google Colaboratory verwenden

Führen Sie nun die Zelle erneut aus. Dazu markieren Sie die Zelle und drücken *Umschalt+Enter*. Wie in [Abbildung](#)

1-5 zu sehen, sollte True als Ausgabe von `is_available()` erscheinen.



The screenshot shows the Google Colaboratory interface. At the top, there's a header with the Colab logo, the file name 'Untitled6.ipynb', and various menu options like 'Datei', 'Bearbeiten', 'Anzeige', etc. Below the header, there's a code editor with the following code:

```
1 import torch
2
3 print(torch.__version__)
4 print(torch.cuda.is_available())
```

The output of the code cell is displayed below the code:

```
1.9.0+cu111
True
```

The interface also shows a RAM and Laufwerk (disk) usage indicator at the top right, and a 'Bearbeiten' (edit) button.

Abbildung 1-5: In Google Colaboratory überprüfen, ob eine GPU aktiv ist



Google bietet eine kostenpflichtige Version namens Colab Pro, die schnellere GPUs, längere Laufzeiten und mehr Arbeitsspeicher bereitstellt. Für die Beispiele in diesem Buch sollte aber die kostenlose Colab-Version ausreichend sein. Mit den letzten Schritten haben Sie überprüft, ob PyTorch installiert ist und um welche Version es sich handelt. Außerdem haben Sie verifiziert, dass eine GPU verfügbar ist und dass die richtigen Treiber installiert sind und korrekt funktionieren. Als Nächstes zeige ich Ihnen, wie Sie Ihr PyTorch auf einem lokalen Computer überprüfen.

Ausführen auf einem lokalen Computer