



深入浅出

张校捷 著

PyTorch

从模型到源码



中国工信出版集团



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>



深入浅出

张校捷 著

PyTorch

从模型到源码

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书从机器学习和深度学习的基础概念入手,由浅到深地详细介绍了PyTorch深度学习框架的知识,主要包含深度学习的基础知识,如神经网络的优化算法、神经网络的模块等;同时也包含了深度学习的进阶知识,如使用PyTorch构建复杂的深度学习模型,以及前沿的深度学习模型的介绍等。另外,为了加深读者对PyTorch深度学习框架的理解和掌握,本书还介绍了PyTorch的源代码结构,包括该框架的Python语言前端和C++语言后端的源代码结构。

作为一本面向初中级读者的技术类图书,本书既可以作为深度学习框架PyTorch入门的参考书籍,也可以作为PyTorch深度学习框架的结构和源代码的阅读指南使用。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有,侵权必究。

图书在版编目(CIP)数据

深入浅出PyTorch:从模型到源码 / 张校捷著. —北京:电子工业出版社,2020.4
ISBN 978-7-121-38641-1

I. ①深… II. ①张… III. ①机器学习 IV. ①TP181

中国版本图书馆CIP数据核字(2020)第036265号

责任编辑:李利健

文字编辑:孔祥飞

印 刷:三河市良远印务有限公司

装 订:三河市良远印务有限公司

出版发行:电子工业出版社

北京市海淀区万寿路173信箱 邮编:100036

开 本:787×980 1/16 印张:26.75 字数:462千字

版 次:2020年4月第1版

印 次:2020年4月第1次印刷

定 价:89.90元

凡所购买电子工业出版社图书有缺损问题,请向购买书店调换。若书店售缺,请与本社发行部联系,联系及邮购电话:(010) 88254888, 88258888。

质量投诉请发邮件至 zllts@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

本书咨询联系方式:010-51260888-819, faq@phei.com.cn。

推荐序

深度学习的浪潮已经席卷了各行各业，从一开始被图像识别、自然语言处理这些相关专业领域的学者和技术人员熟悉到围棋领域由 Google 公司的 AlphaGo 一举成名，被普罗大众所熟悉也就用了三四年时间。在这之后，深度学习继续在各领域生根发芽并将会继续取得丰硕成果。

要学习深度学习，除了学习深度学习的相关理论知识，还要学习各大框架的使用方法。作为深度学习模型的载体，现在实现深度学习的框架主要有 Tensorflow 和 PyTorch，还有一些其他框架如 mxnet、CNTK 和 Chainer 等。而 PyTorch 自 2016 年 10 月首次对外发布以来，仅花了两年时间就可以和 Tensorflow 分庭抗礼，这与 PyTorch 的高性能和易用性是分不开的。截止 2019 年为止，深度学习的主要会议如 NIPS、ICCV、CVPR、ICML 等都有 50%~80% 的会议论文使用了 PyTorch，这也说明了 PyTorch 的简明易用、高性能等特点吸引了大量学术界的用户。

现在市面上讲解 PyTorch 的入门级书已经有了很多，这些书分别从安装、使用和简单示例角度入手讲解 PyTorch，其优点是通俗易懂，但还是缺少一些从底层模块角度讲解 PyTorch 内容的书来帮助读者进一步了解并用好 PyTorch。而本书最大的优点就是囊括了 PyTorch 的常用模块，并给模块配备了不同使用方法的代码和代码背后的底层模块如何实现的逻辑，以便引导用户入门 PyTorch 后继续探索 PyTorch 的各种用法，进而提升科研和工作效率。除此以外，本书还囊括了近几年学术界和工业界都很常用的诸多热门模型，如 SSD、FCN、GAN、LSTM 和 BERT 等的 PyTorch 实现，省去了读者去网上搜索相关实现的烦琐。本书最后专门介绍了 PyTorch 的一些高级使用方法，例如，如何编写算子和自动求导机制，这可以帮助对 PyTorch 有进一步开发需求的研究者和学习者开发需要的算子。

本书作者张校捷自博士毕业以后一直从事 PyTorch 相关的工作，相信他在

NVIDIA 公司的相关工作经验能帮助到广大读者更深入地了解 PyTorch, 从而学好深度学习。作为他的师兄, 鄙人也很荣幸为本书作序。

最后, 希望本书的出版能为国内的深度学习社区添砖加瓦, 能吸引更多热爱深度学习的人加入到这个领域里。

Intel IAGS 软件工程师 石元坤

前言

近年来，以深度学习为代表的人工智能技术正引领着计算机领域和工业领域的一场革命。如今，我们日常生活中的很多场景都和人工智能领域息息相关。比如，日常生活中经常使用的刷脸支付、智能语音助手，以及正在逐渐投入使用的自动驾驶系统，在这些日常应用中，人工智能都扮演着一个重要的角色。而人工智能在这些领域的应用逐渐将人们从简单枯燥的重复劳动中解放出来，让人们能够投入到更有创造力的活动中去。可以预见，在未来，人工智能将会朝着更加智能，而且能够在越来越多的场景中替代人类的方向发展，代替人类去完成一些枯燥乏味的劳动。

目前人工智能的实现依赖的是使用深度学习模型在海量的数据上进行训练。由于深度学习模型是高度模块化的，因此，可以通过组合不同的模块来构造不同的深度学习模型以完成相应的任务。为了利用深度学习模型的这个特点，更加快速地进行深度学习模型的构建和开发，人们开发了一系列深度学习框架，比较有名的深度学习框架包括 PyTorch、TensorFlow、Caffe 等。这些框架的特点是内置了一系列基础模块，并且能够使用一定的方式（比如 Python 语言或者其他配置文件）来控制基础模块的组合，构造复杂多样的深度学习模型。作为其中一个成熟的深度学习框架，PyTorch 在学术界和工业界得到了广泛的应用。特别是在学术界，由于 PyTorch 简单高效，越来越多的研究课题组正在选择 PyTorch 作为深度学习的研究工具。在工业领域，PyTorch 也逐渐加入了一系列新功能，方便深度学习模型的部署。作为一款优秀的深度学习框架，PyTorch 主要的特点包括能够无缝地和 Python 语言整合，方便深度学习模型的构建和调试，以及使用动态计算图模型，灵活地实现各种需要的功能等。可以说，PyTorch 的设计思想完美地契合了 Python 语言的设计思想，即简洁和高效。这也让 PyTorch 对初学者非常友好，容易入门和掌握。可以说，PyTorch 是易于上手，同时保留了灵活性，能够构造绝大多数深度学习模型的一款强大的深度学

习框架。

为了介绍这样一个功能强大的深度学习框架，本书从机器学习和深度学习的基础概念入手，首先让读者对深度学习的基础操作和主要任务有一定了解，在这个基础上，结合 PyTorch 中函数和类的定义介绍 PyTorch 是如何实现深度学习的这些基础操作的。接下来介绍了如何使用 PyTorch 中的这些基础操作来进行组合，构造复杂的深度学习模型。本书着重介绍了这些复杂的深度学习模型在两个重要领域（包括计算机视觉和自然语言处理）的应用，可以说，深度学习到目前为止在这两个领域都取得了巨大的成功。当然，深度学习的应用不止这两个领域。本书还介绍了深度学习在其他领域，如推荐系统、语音识别和语音合成，以及强化学习中的应用。同时，为了加深读者对 PyTorch 深度学习框架的理解，本书在介绍深度学习模型的同时，还着重介绍了 PyTorch 的源代码系统。为了让读者了解 PyTorch 的运行机制，本书还介绍了 PyTorch 的 C++ 后端是如何工作的，以及这个后端如何和前端的 Python 结合起来构造一个灵活的深度学习框架系统。从源代码阅读的角度来说，本书可以作为源代码的一个简单注释，帮助读者理解 PyTorch 各个构成部分的主要作用，以及这些构成部分是如何组合在一起协同工作的。

作为一个逐渐成熟的深度学习框架，PyTorch 经历了一系列的蜕变。从最初的 0.1 版本到目前最新的 1.3 版本，PyTorch 加入了一系列模块和新功能，大大增强了深度学习框架的灵活性并推展了其应用场景，同时也对代码进行了大量的修正，解决了很多代码中的错误。可以说，PyTorch 是一个飞速发展的深度学习框架。迄今为止，PyTorch 已经能够覆盖从学术研究到工业部署的一系列场景，在框架的功能和性能方面有了飞跃般的提升。相信在未来，随着基于深度学习模型的人工智能应用的逐渐推广，PyTorch 作为一个流行的深度学习框架将会得到更广泛的应用。

如无特殊说明，本书在公式中用斜体代表变量，黑斜体代表张量，正体代表常量。卷积神经网络的卷积核和中间特征的单位统一为像素。

因作者水平有限，书中错漏之处在所难免，恳请读者批评、指正。

作者

第 1 章 深度学习概念简介	1
1.1 深度学习的历史.....	1
1.1.1 深度学习的发展过程.....	1
1.1.2 深度学习在计算机视觉领域的发展.....	2
1.1.3 深度学习在自然语言处理和其他领域的发展.....	5
1.2 机器学习基本概念.....	7
1.2.1 机器学习的主要任务.....	7
1.2.2 机器模型的类型.....	8
1.2.3 损失函数以及模型的拟合.....	10
1.3 深度学习基本概念.....	13
1.3.1 向量、矩阵和张量及其运算.....	14
1.3.2 张量的存储.....	18
1.3.3 神经元的概念.....	18
1.4 输入数据的表示方式.....	19
1.4.1 图像数据的表示方式.....	19
1.4.2 文本数据的表示方式.....	21
1.4.3 音频数据的表示方式.....	22
1.5 线性变换和激活函数.....	25
1.5.1 全连接线性变换.....	25
1.5.2 卷积线性变换.....	27
1.5.3 激活函数.....	28
1.6 链式求导法则和反向传播.....	31
1.6.1 基于链式求导的梯度计算.....	31
1.6.2 激活函数的导数.....	33

1.6.3	数值梯度.....	35
1.7	损失函数和优化器.....	36
1.7.1	常用的损失函数.....	36
1.7.2	基于梯度的优化器.....	37
1.7.3	学习率衰减和权重衰减.....	41
1.8	本章总结.....	42
第2章	PyTorch 深度学习框架简介.....	43
2.1	深度学习框架简介.....	43
2.1.1	深度学习框架中的张量.....	43
2.1.2	深度学习框架中的计算图.....	44
2.1.3	深度学习框架中的自动求导和反向传播.....	45
2.2	PyTorch 框架历史和特性更迭.....	46
2.3	PyTorch 的安装过程.....	48
2.4	PyTorch 包的结构.....	51
2.4.1	PyTorch 的主要模块.....	51
2.4.2	PyTorch 的辅助工具模块.....	56
2.5	PyTorch 中张量的创建和维度的操作.....	58
2.5.1	张量的数据类型.....	58
2.5.2	张量的创建方式.....	60
2.5.3	张量的存储设备.....	64
2.5.4	和张量维度相关的方法.....	65
2.5.5	张量的索引和切片.....	68
2.6	PyTorch 中张量的运算.....	69
2.6.1	涉及单个张量的函数运算.....	69
2.6.2	涉及多个张量的函数运算.....	71
2.6.3	张量的极值和排序.....	72
2.6.4	矩阵的乘法和张量的缩并.....	74
2.6.5	张量的拼接和分割.....	76
2.6.6	张量维度的扩增和压缩.....	78
2.6.7	张量的广播.....	79
2.7	PyTorch 中的模块简介.....	80

2.7.1	模块类.....	80
2.7.2	基于模块类的简单线性回归类.....	81
2.7.3	线性回归类的实例化和方法调用.....	83
2.8	PyTorch 的计算图和自动求导机制.....	85
2.8.1	自动求导机制简介.....	85
2.8.2	自动求导机制实例.....	86
2.8.3	梯度函数的使用.....	87
2.8.4	计算图构建的启用和禁用.....	88
2.9	PyTorch 的损失函数和优化器.....	89
2.9.1	损失函数.....	89
2.9.2	优化器.....	91
2.10	PyTorch 中数据的输入和预处理.....	94
2.10.1	数据载入类.....	94
2.10.2	映射类型的数据集.....	95
2.10.3	torchvision 工具包的使用.....	96
2.10.4	可迭代类型的数据集.....	99
2.11	PyTorch 模型的保存和加载.....	100
2.11.1	模块和张量的序列化及反序列化.....	100
2.11.2	模块状态字典的保存和载入.....	102
2.12	PyTorch 数据的可视化.....	103
2.12.1	TensorBoard 的安装和使用.....	104
2.12.2	TensorBoard 的常用可视化数据类型.....	107
2.13	PyTorch 模型的并行化.....	109
2.13.1	数据并行化.....	110
2.13.2	分布式数据并行化.....	111
2.14	本章总结.....	115
第 3 章	PyTorch 计算机视觉模块.....	118
3.1	计算机视觉基本概念.....	118
3.1.1	计算机视觉任务简介.....	118
3.1.2	基础图像变换操作.....	119
3.1.3	图像特征提取.....	122

3.1.4	滤波器的概念	124
3.2	线性层	125
3.3	卷积层	127
3.4	归一化层	136
3.5	池化层	143
3.6	丢弃层	148
3.7	模块的组合	150
3.8	特征提取	152
3.9	模型初始化	156
3.10	常见模型结构	159
3.10.1	InceptionNet 的结构	159
3.10.2	ResNet 的结构	163
3.11	本章总结	166
第 4 章	PyTorch 机器视觉案例	168
4.1	常见计算机视觉任务和数据集	168
4.1.1	图像分类任务简介	168
4.1.2	目标检测任务简介	169
4.1.3	图像分割任务简介	170
4.1.4	图像生成任务简介	171
4.1.5	常用深度学习公开数据集	172
4.2	手写数字识别: LeNet	174
4.2.1	深度学习工程的结构	174
4.2.2	MNIST 数据集的准备	176
4.2.3	LeNet 网络的搭建	178
4.2.4	LeNet 网络的训练和测试	181
4.2.5	超参数的修改和 argparse 库的使用	184
4.3	图像分类: ResNet 和 InceptionNet	186
4.3.1	ImageNet 数据集的使用	186
4.3.2	ResNet 网络的搭建	188
4.3.3	InceptionNet 网络的搭建	192
4.4	目标检测: SSD	203

4.4.1	SSD 的骨架网络结构	203
4.4.2	SSD 的特征提取网络结构	204
4.4.3	锚点框和选框预测	209
4.4.4	输入数据的预处理	213
4.4.5	损失函数的计算	215
4.4.6	模型的预测和非极大抑制算法	217
4.5	图像分割: FCN 和 U-Net	218
4.5.1	FCN 网络结构	219
4.5.2	U-Net 网络结构	223
4.6	图像风格迁移	227
4.6.1	图像风格迁移算法介绍	228
4.6.2	输入图像的特征提取	230
4.6.3	输入图像的优化	233
4.7	生成模型: VAE 和 GAN	235
4.7.1	变分自编码器介绍	236
4.7.2	变分自编码器的实现	238
4.7.3	生成对抗网络介绍	241
4.7.4	生成对抗网络的实现	242
4.8	本章总结	247
第 5 章	PyTorch 自然语言处理模块	249
5.1	自然语言处理基本概念	249
5.1.1	机器翻译相关的自然语言处理研究	249
5.1.2	其他领域的自然语言处理研究	251
5.1.3	自然语言处理中特征提取的预处理	252
5.1.4	自然语言处理中词频特征的计算方法	254
5.1.5	自然语言处理中 TF-IDF 特征的计算方法	256
5.2	词嵌入层	259
5.3	循环神经网络层: GRU 和 LSTM	265
5.3.1	简单循环神经网络	265
5.3.2	长短时记忆网络 (LSTM)	267
5.3.3	门控循环单元 (GRU)	269

5.4	注意力机制	278
5.5	自注意力机制.....	282
5.5.1	循环神经网络的问题.....	282
5.5.2	自注意力机制的基础结构.....	283
5.5.3	使用自注意力机制构建 Seq2Seq 模型.....	286
5.5.4	PyTorch 中自注意力机制的模块.....	288
5.5.5	Pytorch 中的 Transformer 模块.....	290
5.6	本章总结	291
第 6 章	PyTorch 自然语言处理案例.....	293
6.1	word2vec 算法训练词向量.....	293
6.1.1	单词表的创建.....	293
6.1.2	word2vec 算法的实现.....	297
6.1.3	word2vec 算法的特性.....	299
6.2	基于循环神经网络的情感分析.....	300
6.3	基于循环神经网络的语言模型.....	303
6.3.1	语言模型简介.....	303
6.3.2	语言模型的代码.....	306
6.4	Seq2Seq 模型及其应用.....	309
6.4.1	Seq2Seq 模型的结构.....	309
6.4.2	Seq2Seq 模型编码器的代码.....	310
6.4.3	Seq2Seq 模型注意力机制的代码.....	313
6.4.4	Seq2Seq 模型解码器的代码.....	315
6.5	BERT 模型及其应用.....	319
6.5.1	BERT 模型的结构.....	319
6.5.2	BERT 模型的训练方法.....	323
6.5.3	BERT 模型的微调.....	325
6.6	本章总结	326
第 7 章	其他重要模型	328
7.1	基于宽深模型的推荐系统.....	328
7.1.1	推荐系统介绍.....	328

7.1.2	宽深模型介绍	329
7.2	DeepSpeech 模型和 CTC 损失函数	333
7.2.1	语音识别模型介绍	333
7.2.2	CTC 损失函数	335
7.3	使用 Tacotron 和 WaveNet 进行语音合成	340
7.3.1	Tacotron 模型中基于 Seq2Seq 的梅尔过滤器特征合成	341
7.3.2	Tacotron 模型的代码	342
7.3.3	WaveNet 模型介绍	350
7.3.4	因果卷积模块介绍	353
7.3.5	因果卷积模块的代码	354
7.3.6	WaveNet 模型的代码	356
7.4	基于 DQN 的强化学习算法	358
7.4.1	强化学习的基础概念	358
7.4.2	强化学习的环境	359
7.4.3	DQN 模型的原理	361
7.4.4	DQN 模型及其训练过程	362
7.5	使用半精度浮点数训练模型	366
7.5.1	半精度浮点数的介绍	367
7.5.2	半精度模型的训练	368
7.5.3	apex 扩展包的使用	369
7.6	本章总结	371
第 8 章	PyTorch 高级应用	372
8.1	PyTorch 自定义激活函数和梯度	372
8.2	在 PyTorch 中编写扩展	374
8.3	正向传播和反向传播的钩子	381
8.4	PyTorch 的静态计算图	385
8.5	静态计算图模型的保存和使用	390
8.6	本章总结	393
第 9 章	PyTorch 源代码解析	394
9.1	ATen 张量计算库简介	394

9.2	C++的 Python 接口	397
9.3	csrc 模块简介	401
9.4	autograd 和自动求导机制	405
9.5	C10 张量计算库简介	406
9.6	本章总结	408
	参考资料	409

读者服务



微信扫码回复：38641

- 获取博文视点学院 20 元付费内容抵扣券。
- 获取免费增值资源。
- 加入读者交流群，与更多读者互动。
- 获取精选书单推荐。

深度学习概念简介

作为近年来计算机科学的重要研究方向之一，人工智能这个概念得到了学术界和工业界的广泛追捧。迄今为止，人工智能的研究核心依然是基于大数据和高性能计算的深度神经网络，即我们所称的深度学习。由于深度学习需要用到大量的神经网络的构建和运算模块，基于这个需求，人们开发了很多深度学习框架，如 Caffe、MXNet、PyTorch 和 TensorFlow 等。这些框架的共同特点是极大地简化了深度学习的神经网络的构建过程。为了更好地理解这个框架的内在设计思路，以及如何使用这些框架，这里需要对深度学习的概念和术语进行简要介绍，希望对读者理解本书后续几章的内容有所帮助。

1.1 深度学习的历史

1.1.1 深度学习的发展过程

由于 PyTorch 和深度学习的联系紧密，下面回顾一下深度学习的历史。深度学习本质是一个前馈神经网络（Feed-Forward Neural Network），因此，它的历史可以追溯到人工神经网络（Artificial Neural Networks, ANN）研究的时期。人工神经网络主要是通过模拟动物的神经元的运行模式来对现实数据进行拟合和预测，即通过函数来模拟信息在神经元中的变换和传输。最早的人工神经网络是 20 世纪 60 年代 Frank Rosenblatt 发明的感知机（Perceptron）¹，通过对输入进行单层的线性变换来完成分类。然而这个模型并不能解决线性不可分的问题，对于很多现实问题并不适用，因此这个模型很快就被废弃了，从而造成了人工神经网络研究的低潮。到了 20 世纪 80 年代，人们的研究又重新转向人工神经网络，并且开发了多层感知机（Multi-layer

perceptron, MLP)²来解决感知机的问题。相比简单的感知机,多层感知机由于增加了人工神经网络的深度,对于相对复杂的函数拟合效果比较好,理论上,根据万能近似定理(Universal Approximation Theorem)³,对于多层感知机来说,只要参数和隐含层的数量足够,就能够拟合任意的连续函数,同时,人们也发明了反向传播算法(Backpropagation algorithm)⁴来高效地优化多层感知机的模型。使用反向传播算法虽然能够在神经网络层数比较少的时候有效地优化模型,但是不适用于神经网络层数较多的情况,这是由于在较深的网络中使用反向传播算法会导致梯度消失(Gradient Vanish)和梯度爆炸(Gradient Explode)。而且相比于一些比较简单的模型,比如支持向量机(Support Vector Machine, SVM)⁵,多层感知机的计算量比较大,模型的准确率也没有明显优势。因此,在20世纪90年代中期前后,人们对这种类型的人工神经网络逐渐失去了兴趣。神经网络方法研究的衰退一直到了2007年,人们发现了如何训练包含比较多隐含层的神经网络的方法,即深度信念网络(Deep Belief Network, DBN)⁶。深度信念网络的主要特征是叠合多层神经网络(可以使用限制玻尔兹曼机(Restricted Boltzmann Machine⁷, RBM)或者自编码器(Autoencoder⁸),并且逐层训练这些神经网络单元,最后进行微调训练,可以达到训练很深的神经网络的目的。同时,通过调整神经元的连接和激活,一些新的神经网络连接方式和激活函数也进入了人们的视线,著名的如卷积神经网络(Convolution Neural Network, CNN)、循环神经网络(Recurrent Neural Network, RNN)、残差网络(Residue Neural Network),以及新型激活函数如线性整流函数(Rectified Linear Unit, ReLU)。通过引入这些特殊结构的神经网络,能够使梯度在神经网络内部进行有效传播,不仅能够训练隐含层较多的神经网络,同时也可以使用比较多的参数,有效提高了模型的准确率。这里需要提到一点的是,隐含层的数量在神经网络拟合数据的准确性中起到了关键的作用。每个隐含层能够对前一层的神经网络的特征进行组合,从而构造更复杂的特征,并更好地拟合给定的数据集。因此,人们一般把现阶段的神经网络模型称为深度学习(Deep Learning),反映了这些神经网络具有很多隐含层的特点。

1.1.2 深度学习在计算机视觉领域的发展

我们可以从 ImageNet 大规模视觉理解竞赛(ImageNet Large Scale Visual Recognition Challenge, ILSVRC)使用的算法和准确率来一窥近年来深度学习的发展