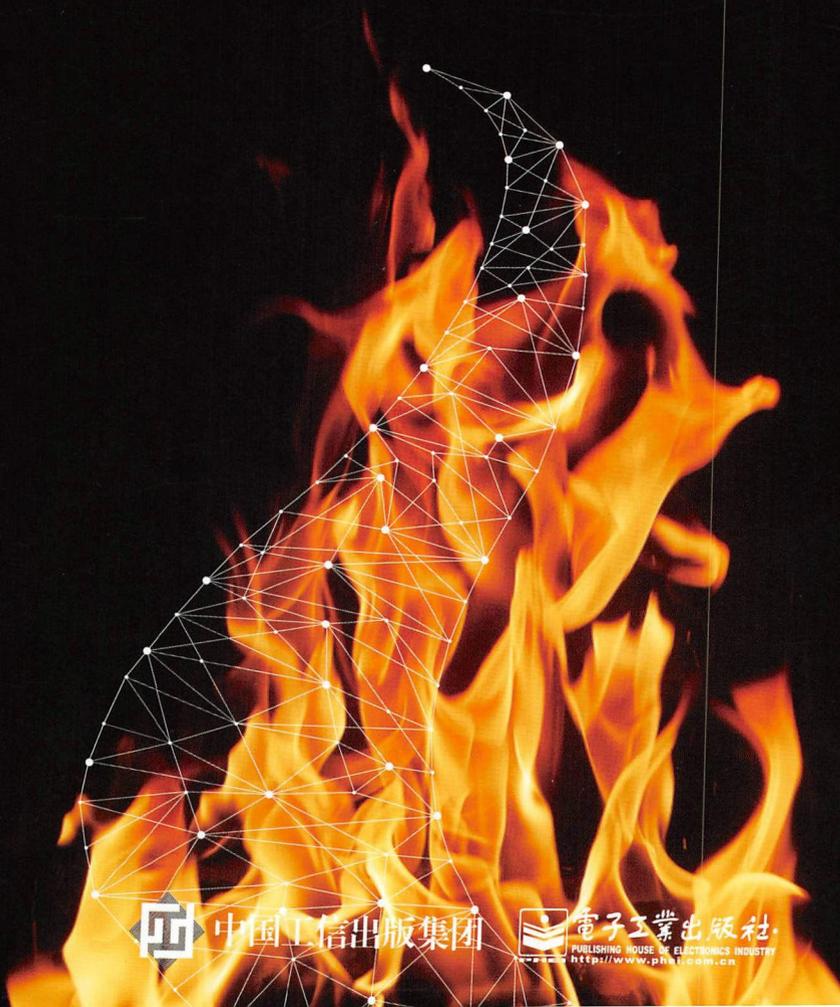


Broadview®
www.broadview.com.cn

深度学习框架 PyTorch 入门与实践

陈云◎编著



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



作者简介



陈云

Python 程序员、Linux 爱好者和 PyTorch 源码贡献者。主要研究方向包括计算机视觉和机器学习。曾获得“2017 知乎·看山杯机器学习挑战赛”一等奖，“2017 天池医疗 AI 大赛”第八名。热衷于推广 PyTorch，并有丰富的使用经验，活跃于 PyTorch 论坛和知乎相关板块。



轻松注册成为博文视点社区用户
(www.broadview.com.cn)，
扫码直达本书页面。

下载资源：本书所提供的示例代码及资源文件均可在【[下载资源](#)】处下载。

提交勘误：您对书中内容的修改意见可在【[提交勘误](#)】处提交，若被采纳，将获赠博文视点社区积分（在您购买电子书时，积分可用来抵扣相应金额）。

与读者交流：在页面下方【[读者评论](#)】处留下您的疑问或观点，与其他读者一同学习交流。

页面入口：

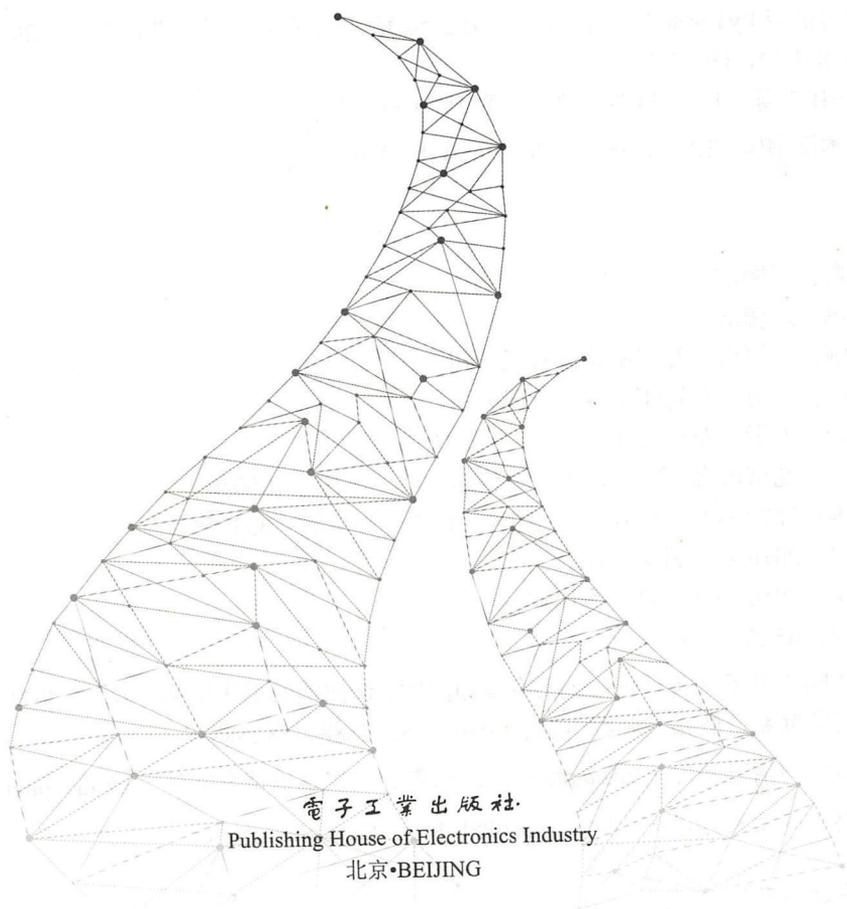
<http://www.broadview.com.cn/33077>

欢迎加入本书读者QQ群：397326578



深度学习框架 PyTorch 入门与实践

陈云◎编著



电子工业出版社
Publishing House of Electronics Industry
北京·BEIJING



内 容 简 介

本书从多维数组 Tensor 开始，循序渐进地带领读者了解 PyTorch 各方面的基础知识，并结合基础知识和前沿研究，带领读者从零开始完成几个经典有趣的深度学习小项目，包括 GAN 生成动漫头像、AI 滤镜、AI 写诗等。本书没有简单机械地介绍各个函数接口的使用，而是尝试分门别类、循序渐进地向读者介绍 PyTorch 的知识，希望读者对 PyTorch 有一个完整的认识。

本书内容由浅入深，无论是深度学习的初学者，还是第一次接触 PyTorch 的研究人员，都能在学习本书的过程中快速掌握 PyTorch。即使是有一定 PyTorch 使用经验的用户，也能够从本书中获得对 PyTorch 不一样的理解。

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

深度学习框架 PyTorch: 入门与实践 / 陈云编著. —北京: 电子工业出版社, 2018.1

ISBN 978-7-121-33077-3

I. ①深…II. ①陈…III. ①机器学习 - 研究 IV. ①TP181

中国版本图书馆 CIP 数据核字 (2017) 第 286730 号

策划编辑: 郑柳洁

责任编辑: 郑柳洁

印 刷: 三河市双峰印刷装订有限公司

装 订: 三河市双峰印刷装订有限公司

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: 18.75 字数: 353 千字

版 次: 2018 年 1 月第 1 版

印 次: 2018 年 4 月第 3 次印刷

定 价: 65.00 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 zlts@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。



前言

为什么写这本书

2016 年是属于 TensorFlow 的一年，凭借谷歌的大力推广，TensorFlow 占据了各大媒体的头条。2017 年年初，PyTorch 的横空出世吸引了研究人员极大的关注，PyTorch 简洁优雅的设计、统一易用的接口、追风逐电的速度和变化无方的灵活性给人留下深刻的印象。

作为一门 2017 年刚刚发布的深度学习框架，研究人员所能获取的学习资料有限，中文资料更是比较少。笔者长期关注 PyTorch 发展，经常在论坛上帮助 PyTorch 新手解决问题，在平时的科研中利用 PyTorch 进行各个方面的研究，有着丰富的使用经验。看到国内的用户对 PyTorch 十分感兴趣，迫切需要一本能够全面讲解 PyTorch 的书籍，于是本书就这么诞生了。

本书的结构

本书分为两部分：第 2~5 章主要介绍 PyTorch 的基础知识。

- 第 2 章介绍 PyTorch 的安装和配置学习环境。同时以最概要的方式介绍 PyTorch 的主要内容，让读者对 PyTorch 有一个大概的整体印象。
- 第 3 章介绍 PyTorch 中多维数组 Tensor 和动态图 autograd/Variable 的使用，并配以例子，让读者分别使用 Tensor 和 autograd 实现线性回归，比较二者的不同点。本章还对 Tensor 的底层设计，以及 autograd 的原理进行了分析，给读者以更全面具体的讲解。
- 第 4 章介绍 PyTorch 中神经网络模块 nn 的基础用法，同时讲解了神经网络中的“层”、“损失函数”、“优化器”等，最后带领读者用不到 50 行的代码搭建出曾夺得 ImageNet 冠军的 ResNet。
- 第 5 章介绍 PyTorch 中数据加载、GPU 加速和可视化等相关工具。



第 6~10 章主要介绍实战案例。

- 第 6 章是承上启下的一章，目标不是教会读者新函数、新知识，而是结合 Kaggle 中一个经典的比赛，实现一个深度学习中比较简单的图像二分类问题。在实现的过程中，带领读者复习前 5 章的知识，并提出代码规范以合理地组织程序和代码，使程序更可读、可维护。第 6 章还介绍在 PyTorch 中如何进行 debug。
- 第 7 章为读者讲解当前最火爆的生成对抗网络（GAN），带领读者从零开始实现一个动漫头像生成器，能够利用 GAN 生成风格多变的动漫头像。
- 第 8 章为读者讲解风格迁移的相关知识，并带领读者实现风格迁移网络，将自己的照片变成“高大上”的名画。
- 第 9 章为读者讲解一些自然语言处理的基础知识，并讲解 CharRNN 的原理。然后利用其收集几万首唐诗，训练出一个可以自动写诗歌的小程序。这个小程序可以控制生成诗歌的格式和意境，还能生成藏头诗。
- 第 10 章为读者介绍图像描述任务，并以最新的 AI Challenger 比赛的数据为例，带领读者实现一个可以进行简单图像描述的小程序。

第 1 章和第 11 章是本书的首章和末章，第 1 章介绍 PyTorch 的优势，以及和市面上其他几款框架的对比。第 11 章是对本书的总结，以及对 PyTorch 不足之处的思考，同时对读者未来的学习提出建议。

关于代码

本书的所有代码都开源在 GitHub^①上，其中：

- 第 2~5 章的代码以 Jupyter Notebook 形式提供，读者可以在自己的计算机上交互式地修改运行它。
- 第 6~10 章的代码以单独的程序给出，每个函数的作用与细节在代码中有大量的注释。

本书的代码，在最新版的 PyTorch 0.2 上运行，同时支持 Python 2 和 Python 3，其中：

- 前 5 章的代码同时在 Python 2.7 和 Python 3.5 上验证，并得到最终结果。
- 第 6~10 章的代码，主要在 Python 2.7 上运行并得到最终结果，同时在 Python 3.5 上测试未报错。

^①<https://github.com/chenyuntc/pytorch-book>



适读人群

学习本书需要读者具备以下基础知识：

- 了解 Python 的基础语法，掌握基础的 Python 使用方法。
- 有一定深度学习基础，了解反向传播、卷积神经网络等基础知识，但并不要求深入了解。
- 具备梯度、导数等高中数学基础知识。

以下知识不是必需的，但最好了解：

- numpy 的使用。
- 深度学习的基本流程或者其他深度学习框架的使用。

本书不适合哪些读者：

- 没有任何深度学习基础的用户。
- 没有 Python 基础的用户。
- 只能使用 Windows 的用户。

本书约定

在本书中，笔者是本书编著者的自称，作者指的是软件、论文等的作者，读者指阅读本书的你。

本书前 5 章的代码由 Jupyter Notebook 转换而来，其中：

- In 后面跟着的是输入的代码。
- Out 是指程序的运行结果，运行结果取决于 In 的最后一行。
- Print 后面跟着程序的打印输出内容，只有在 In 程序中调用了 print 函数/语句才会有 Print 输出。
- Jupyter 会自动输出 Image 对象和 matplotlib 可视化结果，所以书中以“程序输出”命名的图片都来自 Jupyter 的程序输出。这些图片的说明在代码注释中。

如何使用本书

本书第 2 章是 PyTorch 快速入门，第 3~5 章是对这些内容的详细深入介绍。第 6 章是一个简单而完整的深度学习案例。



前言

如果你是经验丰富的研究人员，之前对 PyTorch 十分熟悉，对本书的某些例子比较感兴趣，那么你可以跳过前 5 章，直接阅读第 6 章，了解这些例子的程序设计与文件组织安排，然后阅读相应的例子。

如果你是初学者，想以最快的速度掌握 PyTorch 并将 PyTorch 应用到实际项目中，那么你可以花费 2~3 小时阅读 2.2 节的相关内容。如果你需要深入了解某部分的内容，那么可以阅读相应章节。

如果你是初学者，想完整全面地掌握 PyTorch，那么建议你：

- 先阅读第 1~5 章，了解 PyTorch 的各个基础知识。
- 再阅读第 6 章，了解 PyTorch 实践中的技巧。
- 最后从第 7~10 章挑选出感兴趣的例子，动手实践。

最后，希望读者在阅读本书的时候，尽量结合本书的配套代码阅读、修改、运行之。

致谢

杜玉姣同学在我编写本书的时候，给了我许多建议，并协助审阅了部分章节，在此特向她表示谢意。在编写本书时，本书编辑郑柳洁女士给予了很大的帮助，在此特向她致谢。感谢我的家人一直以来对我的支持，感谢我的导师肖波副教授对我的指导。感谢我的同学、师弟师妹们，他们在使用 PyTorch 中遇到了很多问题，给了我许多反馈意见。

由于笔者水平所限，书中难免有错误和不当之处，欢迎读者批评指正。具体意见可以发表在 GitHub 上的 issue (<https://github.com/chenyuntc/pytorch-book/issues>) 中，或者通过邮箱 (i@knew.be) 联系笔者。



目录

1	PyTorch 简介	1
1.1	PyTorch 的诞生	1
1.2	常见的深度学习框架简介	2
1.2.1	Theano	3
1.2.2	TensorFlow	3
1.2.3	Keras	5
1.2.4	Caffe/Caffe2	5
1.2.5	MXNet	6
1.2.6	CNTK	7
1.2.7	其他框架	8
1.3	属于动态图的未来	8
1.4	为什么选择 PyTorch	10
1.5	星火燎原	12
1.6	fast.ai 放弃 Keras+TensorFlow 选择 PyTorch	13
2	快速入门	16
2.1	安装与配置	16
2.1.1	安装 PyTorch	16
2.1.2	学习环境配置	20
2.2	PyTorch 入门第一步	30
2.2.1	Tensor	30
2.2.2	Autograd: 自动微分	35
2.2.3	神经网络	38
2.2.4	小试牛刀: CIFAR-10 分类	43



3	Tensor 和 autograd	51
3.1	Tensor	51
3.1.1	基础操作	52
3.1.2	Tensor 和 Numpy	70
3.1.3	内部结构	73
3.1.4	其他有关 Tensor 的话题	76
3.1.5	小试牛刀：线性回归	78
3.2	autograd	81
3.2.1	Variable	82
3.2.2	计算图	86
3.2.3	扩展 autograd	94
3.2.4	小试牛刀：用 Variable 实现线性回归	99
4	神经网络工具箱 nn	103
4.1	nn.Module	103
4.2	常用的神经网络层	107
4.2.1	图像相关层	107
4.2.2	激活函数	110
4.2.3	循环神经网络层	114
4.2.4	损失函数	116
4.3	优化器	116
4.4	nn.functional	118
4.5	初始化策略	120
4.6	nn.Module 深入分析	122
4.7	nn 和 autograd 的关系	129
4.8	小试牛刀：用 50 行代码搭建 ResNet	130



5	PyTorch 中常用的工具	135
5.1	数据处理	135
5.2	计算机视觉工具包: torchvision	147
5.3	可视化工具	149
5.3.1	Tensorboard	150
5.3.2	visdom	152
5.4	使用 GPU 加速: cuda	158
5.5	持久化	161
6	PyTorch 实战指南	164
6.1	编程实战: 猫和狗二分类	164
6.1.1	比赛介绍	165
6.1.2	文件组织架构	165
6.1.3	关于__init__.py	167
6.1.4	数据加载	167
6.1.5	模型定义	170
6.1.6	工具函数	171
6.1.7	配置文件	174
6.1.8	main.py	176
6.1.9	使用	184
6.1.10	争议	185
6.2	PyTorch Debug 指南	187
6.2.1	ipdb 介绍	187
6.2.2	在 PyTorch 中 Debug	191
7	AI 插画师: 生成对抗网络	197
7.1	GAN 的原理简介	198
7.2	用 GAN 生成动漫头像	202
7.3	实验结果分析	211



8 AI 艺术家：神经网络风格迁移	215
8.1 风格迁移原理介绍	216
8.2 用 PyTorch 实现风格迁移	222
8.3 实验结果分析	233
9 AI 诗人：用 RNN 写诗	237
9.1 自然语言处理的基础知识	237
9.1.1 词向量	238
9.1.2 RNN	240
9.2 CharRNN	243
9.3 用 PyTorch 实现 CharRNN	246
9.4 实验结果分析	257
10 Image Caption：让神经网络看图讲故事	260
10.1 图像描述介绍	261
10.2 数据	262
10.2.1 数据介绍	262
10.2.2 图像数据处理	270
10.2.3 数据加载	272
10.3 模型与训练	275
10.4 实验结果分析	280
11 展望与未来	282
11.1 PyTorch 的局限与发展	282
11.2 使用建议	286



1 PyTorch 简介

1.1 PyTorch 的诞生

2017年1月，Facebook人工智能研究院（FAIR）团队在GitHub上开源了PyTorch（PyTorch的Logo如图1-1所示），并迅速占领GitHub热度榜榜首。

作为一个2017年才发布，具有先进设计理念的框架，PyTorch的历史可追溯到2002年就诞生于纽约大学的Torch。Torch使用了一种不是很大众的语言Lua作为接口。Lua简洁高效，但由于其过于小众，用的人不是很多，以至于很多人听说要掌握Torch必须新学一门语言就望而却步（其实Lua是一门比Python还简单的语言）。



图 1-1 PyTorch 的 Logo，英文 Torch 是火炬的意思，所以 Logo 中有火焰

考虑到 Python 在计算科学领域的领先地位，以及其生态完整性和接口易用性，几乎任何框架都不可避免地要提供 Python 接口。终于，在 2017 年，Torch 的幕后团队推出了 PyTorch。PyTorch 不是简单地封装 Lua Torch 提供 Python 接口，而是对 Tensor 之上的所有模块进行了重构，并新增了最先进的自动求导系统，成为当下最流行的动态图框架。

PyTorch 一经推出就立刻引起了广泛关注，并迅速在研究领域流行起来。图 1-2 所示为 Google 指数，PyTorch 自发布起关注度就在不断上升，截至 2017 年 10 月 18 日，



PyTorch 的热度已然超越了其他三个框架 (Caffe、MXNet 和 Theano), 并且其热度还在持续上升中。

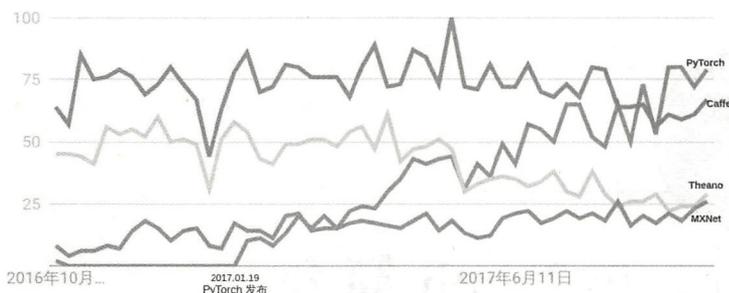


图 1-2 PyTorch 和 Caffe、Theano、MXNet 的 Google 指数对比 (类别为科学)

1.2 常见的深度学习框架简介

随着深度学习的发展,深度学习框架如雨后春笋般诞生于高校和公司中。尤其是近两年,Google、Facebook、Microsoft 等巨头都围绕深度学习重点投资了一系列新兴项目,他们也一直在支持一些开源的深度学习框架。

目前研究人员正在使用的深度学习框架不尽相同,有 TensorFlow、Caffe、Theano、Keras 等,常见的深度学习框架如图 1-3 所示。这些深度学习框架被应用于计算机视觉、语音识别、自然语言处理与生物信息学等领域,并获取了极好的效果。本节主要介绍当前深度学习领域影响力比较大的几个框架,限于笔者个人使用经验和了解程度,对各个框架的评价可能有不准确的地方。



图 1-3 常见的深度学习框架

1.2.1 Theano

Theano 最初诞生于蒙特利尔大学 LISA 实验室，于 2008 年开始开发，是第一个有较大影响力的 Python 深度学习框架。

Theano 是一个 Python 库，可用于定义、优化和计算数学表达式，特别是多维数组 (numpy.ndarray)。在解决包含大量数据的问题时，使用 Theano 编程可实现比手写 C 语言更快的速度，而通过 GPU 加速，Theano 甚至可以比基于 CPU 计算的 C 语言快上好几个数量级。Theano 结合了计算机代数系统 (Computer Algebra System, CAS) 和优化编译器，还可以为多种数学运算生成定制的 C 语言代码。对于包含重复计算的复杂数学表达式的任务而言，计算速度很重要，因此这种 CAS 和优化编译器的组合是很有用的。对需要将每一种不同的数学表达式都计算一遍的情况，Theano 可以最小化编译/解析的计算量，但仍然会给出如自动微分那样的符号特征。

Theano 诞生于研究机构，服务于研究人员，其设计具有较浓厚的学术气息，但在工程设计上有较大的缺陷。一直以来，Theano 困难调试、构建图慢等缺点为人所诟病。为了加速深度学习研究，人们在它的基础之上，开发了 Lasagne、Blocks、PyLearn2 和 Keras 等第三方框架，这些框架以 Theano 为基础，提供了更好的封装接口以方便用户使用。

2017 年 9 月 28 日，在 Theano 1.0 正式版即将发布前夕，LISA 实验室负责人，深度学习三巨头之一的 Yoshua Bengio 宣布 Theano 即将停止开发：“Theano is Dead”。尽管 Theano 即将退出历史舞台，但作为第一个 Python 深度学习框架，它很好地完成了自己的使命，为深度学习研究人员的早期拓荒提供了极大的帮助，同时也为之后深度学习框架的开发奠定了基本设计方向：以计算图为框架的核心，采用 GPU 加速计算。

2017 年 11 月，LISA 实验室在 GitHub 上开启了一个初学者入门项目，旨在帮助实验室新生快速掌握机器学习相关的实践基础，而该项目正是使用 PyTorch 作为教学框架。

点评：由于 Theano 已经停止开发，不建议作为研究工具继续学习。

1.2.2 TensorFlow

2015 年 11 月 10 日，Google 宣布推出全新的机器学习开源工具 TensorFlow。TensorFlow 最初是由 Google 机器智能研究部门的 Google Brain 团队开发，基于 Google 2011 年开发的深度学习基础架构 DistBelief 构建起来的。TensorFlow 主要用于进行机器学习和深度神经网络研究，但它是一个非常基础的系统，因此也可以应用于众多领域。由于

Google 在深度学习领域的巨大影响力和强大的推广能力, TensorFlow 一经推出就获得了极大的关注, 并迅速成为如今用户最多的深度学习框架。

TensorFlow 在很大程度上可以看作 Theano 的后继者, 不仅因为它们有很大一批共同的开发者, 而且它们还拥有相近的设计理念, 都是基于计算图实现自动微分系统。TensorFlow 使用数据流图进行数值计算, 图中的节点代表数学运算, 而图中的边则代表在这些节点之间传递的多维数组(张量)。

TensorFlow 编程接口支持 Python 和 C++。随着 1.0 版本的公布, Java、Go、R 和 Haskell API 的 alpha 版本也被支持。此外, TensorFlow 还可在 Google Cloud 和 AWS 中运行。TensorFlow 还支持 Windows 7、Windows 10 和 Windows Server 2016。由于 TensorFlow 使用 C++ Eigen 库, 所以库可在 ARM 架构上编译和优化。这也就意味着用户可以在各种服务器和移动设备上部署自己的训练模型, 无须执行单独的模型解码器或者加载 Python 解释器。

作为当前最流行的深度学习框架, TensorFlow 获得了极大的成功, 对它的批评也不绝于耳, 总结起来主要有以下四点。

- 过于复杂的系统设计, TensorFlow 在 GitHub 代码仓库的总代码量超过 100 万行。这么大的代码仓库, 对于项目维护者来说维护成为了一个难以完成的任务, 而对读者来说, 学习 TensorFlow 底层运行机制更是一个极其痛苦的过程, 并且大多数时候这种尝试以放弃告终。
- 频繁变动的接口。TensorFlow 的接口一直处于快速迭代之中, 并且没有很好地考虑向后兼容性, 这导致现在许多开源代码已经无法在新版的 TensorFlow 上运行, 同时也间接导致了許多基于 TensorFlow 的第三方框架出现 BUG。
- 接口设计过于晦涩难懂。在设计 TensorFlow 时, 创造了图、会话、命名空间、Place-Holder 等诸多抽象概念, 对普通用户来说难以理解。同一个功能, TensorFlow 提供了多种实现, 这些实现良莠不齐, 使用中还有细微的区别, 很容易将用户带入坑中。
- 文档混乱脱节。TensorFlow 作为一个复杂的系统, 文档和教程众多, 但缺乏明显的条理和层次, 虽然查找很方便, 但用户却很难找到一个真正循序渐进的入门教程。

由于直接使用 TensorFlow 的生产力过于低下, 包括 Google 官方等众多开发者尝试基于 TensorFlow 构建一个更易用的接口, 包括 Keras、Sonnet、TFLearn、TensorLayer、Slim、Fold、PrettyLayer 等数不胜数的第三方框架每隔几个月就会在新闻中出现一次, 但是又大多归于沉寂, 至今 TensorFlow 仍没有一个统一易用的接口。

凭借 Google 着强大的推广能力, TensorFlow 已经成为当今最炙手可热的深度学习框架, 但是由于自身的缺陷, TensorFlow 离最初的设计目标还很遥远。另外, 由于 Google

对 TensorFlow 略显严格的把控，目前各大公司都在开发自己的深度学习框架。

点评：不完美但最流行的深度学习框架，社区强大，适合生产环境。

1.2.3 Keras

Keras 是一个高层神经网络 API，由纯 Python 编写而成并使用 TensorFlow、Theano 及 CNTK 作为后端。Keras 为支持快速实验而生，能够把想法迅速转换为结果。Keras 应该是深度学习框架之中最容易上手的一个，它提供了一致而简洁的 API，能够极大地减少一般应用下用户的工作量，避免用户重复造轮子。

严格意义上讲，Keras 并不能称为一个深度学习框架，它更像一个深度学习接口，它构建于第三方框架之上。Keras 的缺点很明显：过度封装导致丧失灵活性。Keras 最初作为 Theano 的高级 API 而诞生，后来增加了 TensorFlow 和 CNTK 作为后端。为了屏蔽后端的差异性，提供一致的用户接口，Keras 做了层层封装，导致用户在新增操作或是获取底层的数据信息时过于困难。同时，过度封装也使得 Keras 的程序过于缓慢，许多 BUG 都隐藏于封装之中，在绝大多数场景下，Keras 是本节介绍的所有框架中最慢的一个。

学习 Keras 十分容易，但是很快就会遇到瓶颈，因为它缺少灵活性。另外，在使用 Keras 的大多数时间里，用户主要是在调用接口，很难真正学习到深度学习的内容。

点评：入门最简单，但是不够灵活，使用受限。

1.2.4 Caffe/Caffe2

Caffe 的全称是 Convolutional Architecture for Fast Feature Embedding，它是一个清晰、高效的深度学习框架，核心语言是 C++，它支持命令行、Python 和 MATLAB 接口，既可以在 CPU 上运行，也可以在 GPU 上运行。

Caffe 的优点是简洁快速，缺点是缺少灵活性。不同于 Keras 因为太多的封装导致灵活性丧失，Caffe 灵活性的缺失主要是因为它的设计。在 Caffe 中最主要的抽象对象是层，每实现一个新的层，必须要利用 C++ 实现它的前向传播和反向传播代码，而如果想要新层运行在 GPU 上，还需要同时利用 CUDA 实现这一层的前向传播和反向传播。这种限制使得不熟悉 C++ 和 CUDA 的用户扩展 Caffe 十分困难。

Caffe 凭借其易用性、简洁明了的源码、出众的性能和快速的原型设计获取了众多用户，曾经占据深度学习领域的半壁江山。但是在深度学习新时代到来之时，Caffe 已经表现出明显的力不从心，诸多问题逐渐显现（包括灵活性缺失、扩展难、依赖众多环