

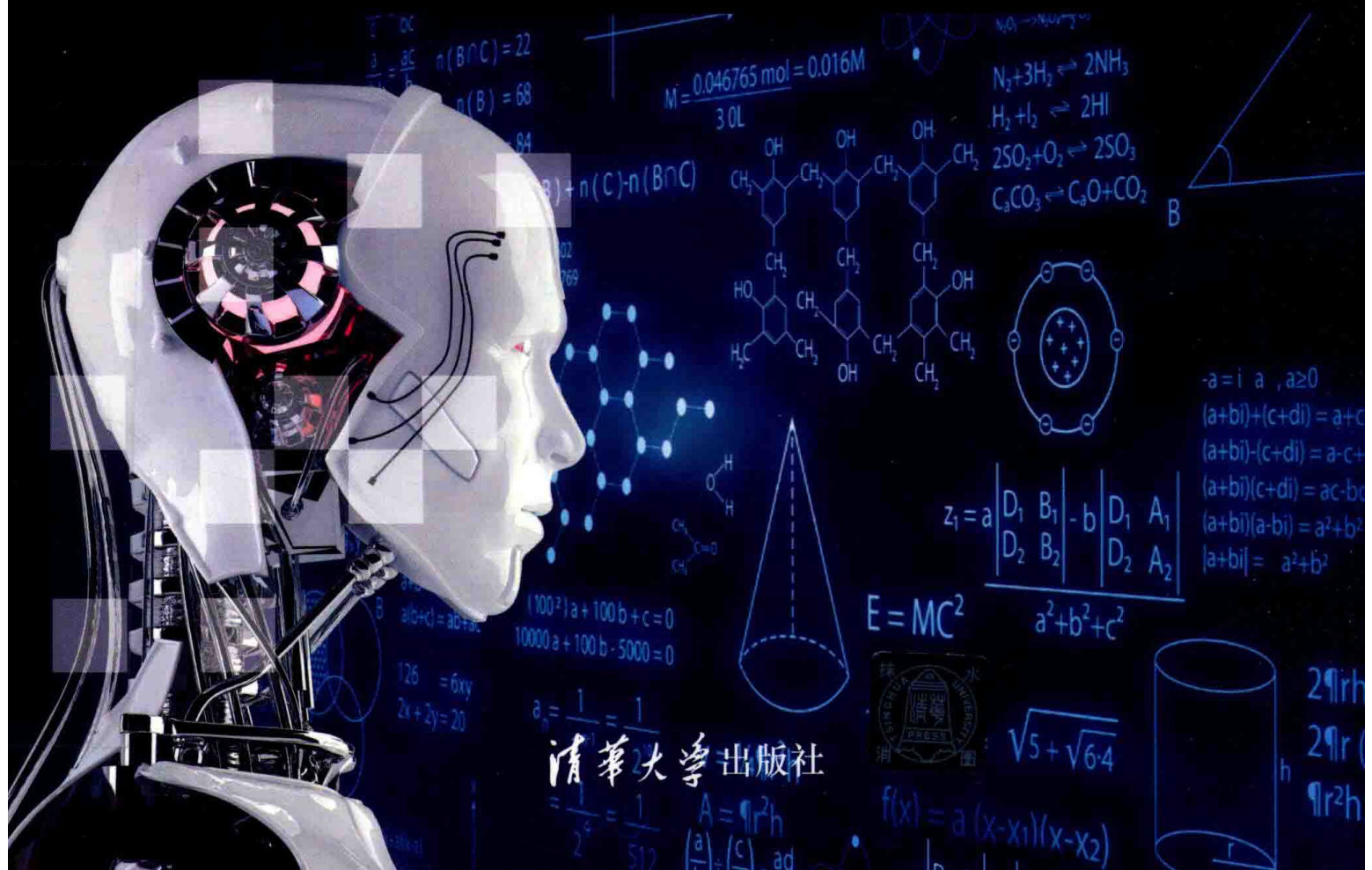
全面分析TensorFlow 2.0的新框架思想和模型设计
详细讲解TensorFlow 2.0以及Keras编程方法与技巧

Introduction to TensorFlow 2.0 Deep Learning

TensorFlow 2.0

深度学习从零开始学

王晓华 著



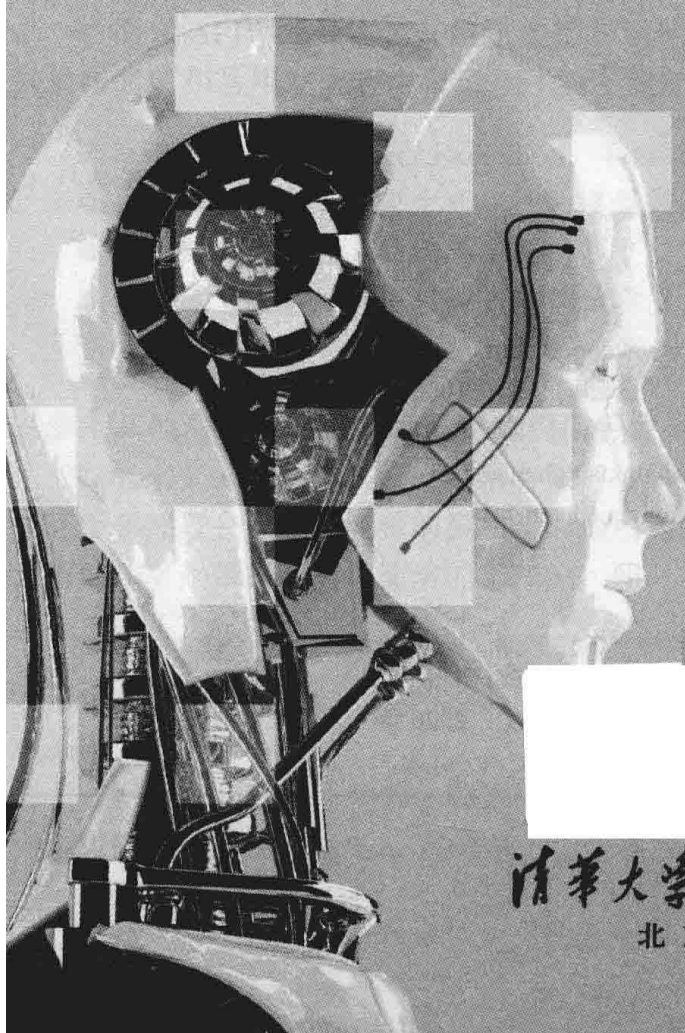
清华大学出版社

7

TensorFlow 2.0

深度学习从零开始学

王晓华 著



清华大学出版社
北京

内 容 简 介

随着人工智能的发展以及 TensorFlow 在人工智能方面的火热应用,越来越多的大学逐步开设深度学习和人工智能课程。本书既是一本为读者量身定制的 TensorFlow 2.0 入门教材,也是针对需要学习 TensorFlow 2.0 新内容的读者提供的基础与进阶知识的深入型教材。

本书分为 10 章,主要内容包括 TensorFlow 2.0 开发环境、TensorFlow 2.0 新特性、TensorFlow 与 Keras 的使用、TensorFlow 2.0 语法基础、卷积层详解与 MNIST 实战、Dataset 使用详解、TensorFlow Datasets 和 TensorBoard 详解、ResNet 及其实战、注意力机制、卷积神经网络实战。

本书内容详尽、示例丰富,是广大对 TensorFlow 2.0 感兴趣的读者必备的参考书,同时也非常适合大中专院校师生学习阅读,还可作为高等院校计算机及相关专业的教材使用。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

TensorFlow 2.0 深度学习从零开始学/王晓华著. — 北京:清华大学出版社, 2020.5
ISBN 978-7-302-55273-4

I. ①T… II. ①王… III. ①人工智能—算法 IV. ①TP18

中国版本图书馆 CIP 数据核字(2020)第 050660 号

责任编辑:夏毓彦

封面设计:王翔

责任校对:闫秀华

责任印制:沈露

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦 A 座 邮 编:100084

社总机:010-62770175 邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质量反馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

印 装 者:北京嘉实印刷有限公司

经 销:全国新华书店

开 本:190mm×260mm 印 张:15.25 字 数:415 千字

版 次:2020 年 6 月第 1 版 印 次:2020 年 6 月第 1 次印刷

定 价:69.00 元

产品编号:084293-01

前言

作为主流的深度学习框架，TensorFlow 引领了深度学习和人工智能领域的全面发展和成长壮大。它的出现使得深度学习的学习门槛被大大降低，不仅是数据专家，就连普通的程序设计人员甚至于相关专业的学生都可以用来开发新的 AI 程序而不需要深厚的编程功底。

然而这并不代表 TensorFlow 已经完美。由于代码设计的历史原因以及开源框架的通病，随着时间的积累 TensorFlow 在其框架内部积累了大量的过时和不推荐使用的 API，这极大地影响了 TensorFlow 的性能，因此在广泛听取使用者和代码项目组成员的意见后，TensorFlow 2.0 横空出世。

本书也是因为 TensorFlow 2.0 的出现而推出的一本包含基础与进阶内容的书。

整体而言，为了吸引用户，TensorFlow 2.0 从简单、强大、可扩展 3 个层面进行了重新设计。特别是在简单化方面，TensorFlow 2.0 提供更简化的 API、注重 Keras、结合了 Eager Execution。

“简化 API、减少冗余并改进文档和示例”是 TensorFlow 2.0 的主要任务，其更新重点放在简单和易用性上，主要进行了以下更新：

- 使用 Keras 和 Eager Execution，轻松建立简单的模型并执行；
- 在任何平台上实现生产环境的模型部署；
- 为研究提供强大的实验工具；
- 通过清除不推荐使用的 API 和减少重复来简化 API。

除此之外，在整体架构方面，TensorFlow 2.0 吸收和归并了新的设计模式和想法，以及参考其他深度学习框架所采用的积极后台模型调整了新的框架。

TensorFlow 2.0 架构统一了所有保留的组件和模块，并将其完整打包成一个综合平台，将训练和部署分成 2 个完整的模块，而通用的是对于模型本身的存储和参数的存储。简单化的新框架更是带来了更加简洁的工作流，即：

(1) 使用 `tf.data` 加载数据。使用输入管道读取训练数据，输入管道使用 `tf.data` 创建。利用 `tf.feature_column` 描述特征，如分段和特征交叉。此外还支持内存数据的便捷输入（如 NumPy）。

(2) 使用 `tf.keras` 构建、训练并验证模型。Keras 与 TensorFlow 部分紧密集成，因此用户可以随时访问 TensorFlow 的函数，如线性或逻辑回归、梯度上升树、随机森林等也可以直

接使用（使用 `tf.estimator` API 实现）。如果不想从头开始训练模型，用户也可以很快利用迁移学习来训练使用 TensorFlow Hub 模块的 Keras 或 Estimator 模型。

(3) 快速执行运行和调试过程，然后使用 `tf.function` 充分利用图形的优势。在默认情况下，TensorFlow 2.0 按快速执行方式运行，以便于顺利调试。此外，`tf.function` 注释可以方便地将 Python 程序转换为 TensorFlow 图形。此过程保留了 1.x TensorFlow 基于图形执行的所有优点：性能优化，远程执行以及方便序列化、导出和部署的能力，同时实现了在 Python 中表达程序的灵活性和易用性。

(4) 使用分布式策略进行分布式训练。对于大型机器学习训练任务，分布式策略 API 可以轻松地在不同硬件配置上分配和训练模型，无须更改模型的定义。由于 TensorFlow 支持各种硬件加速器，如 CPU、GPU 和 TPU，因此用户可以将训练负载分配到单节点/多加速器以及多节点/多加速器配置上（包括 TPU Pod）。这个 API 支持多种群集化配置，也提供了在本地或云环境中部署 Kubernetes 群集训练的模板。

(5) 导出到 Saved Model。TensorFlow 将对 Saved Model 进行标准化。作为 TensorFlow 服务的一部分，Saved Model 将成为 TensorFlow Lite、TensorFlow.js、TensorFlow Hub 等格式的可互换格式。

其中的 `data` 和 Keras 都是 TensorFlow 2.0 新引入并指定使用的 API。

`data` API 使得数据读取和输入模型训练变得非常容易，由程序读写不同格式数据时的“事无巨细”，变为仅仅使用一个 `for` 循环就可以轻松完成数据的读取和输入，让我们打开一个大型或者超大型数据集就像打开一个文本 `txt` 文件一样简单。

Keras API 使得上手 TensorFlow 2.0 更容易。更为重要的是，Keras 提供了几个模型构建 API (`Sequential`、`Functional` 以及 `Subclassing`)，因此用户可以选择正确的抽象化 (`Abstraction`) 级别，以及非常简便地编写符合项目需要的深度学习模型。

本书有何特色

1. 介绍全面，讲解详尽

本书全面地讲解 TensorFlow 2.0 的新框架设计思想和模型的编写，详细介绍 TensorFlow 2.0 的安装、使用以及 TensorFlow 2.0 官方所推荐的 Keras 编程方法与技巧等。

2. 作者经验丰富，代码编写细腻

本书的代码编写由低到高，针对各个环节都有详尽的说明，使得读者能够充分了解和掌握代码各个模块的编写方法和技巧，是一本非常好的 TensorFlow 2.0 学习教程。

作者是长期奋战在科研和工业界的一线算法设计和程序编写人员，实战经验丰富，对代码中可能会出现的各种问题和“坑”有丰富的处理经验，能够使得读者少走很多弯路。

3. 理论扎实，深入浅出

在代码设计的基础上，本书还深入浅出地介绍深度学习需要掌握的一些基本理论知识，通过大量的公式与图示结合的方式对理论进行介绍。

4. 提供完善的技术支持和售后服务

本书提供了专门的技术支持邮箱：booksaga@163.com。读者在阅读本书过程中有任何疑问都可以通过该邮箱获得帮助。

本书内容及知识体系

本书是基于 TensorFlow 2.0 的新架构模式和框架，完整介绍 TensorFlow 2.0 使用方法和一些进阶教程，主要内容如下：

第 1 章详细介绍 TensorFlow 2.0 的安装方法以及对应的运行环境的安装，并且通过一个简单的例子验证 TensorFlow 2.0 的安装效果。在本章中，还将介绍 TensorFlow 2.0 硬件的采购。切记，一块能够运行 TensorFlow 2.0 GPU 版本的显卡能让学习效率事半功倍。

从第 2 章开始是本书的重点部分，这一章从 Eager 的引入开始介绍 TensorFlow 2.0 的编程方法和步骤，并结合 Keras 进行 TensorFlow 2.0 模型设计的完整步骤，以及自定义层的方法。第 2 章的内容看起来很简单，但是是本书的基础内容和核心精华，读者一定要反复阅读，认真掌握所有内容和代码的编写。

第 3 章是有关 Keras 的介绍。Keras 是 TensorFlow 2.0 中新引入的一个高级 API。本章从简单的例子开始，循序渐进地介绍使用 Keras 进行程序设计的方法和步骤，同时还将介绍深度学习中一个非常重要的模块，即全连接神经网络。

第 4 章是理论部分，初步涉及一些深度学习的基本理论和算法。

使用卷积神经网络去识别物体是深度学习的一个经典内容，在第 5 章中详细介绍卷积神经网络的原理和各个模型的使用和自定义内容，以及如何借助卷积神经网络（CNN）算法构建一个简单的 CNN 模型进行 MNIST 数字识别。此章和第 3 章同为本书的重点内容，能够极大地协助读者对 TensorFlow 2.0 框架的使用和程序的编写。

第 6、7 章是 TensorFlow 2.0 中一些高级 API 的介绍。通过使用集成在 TensorFlow 中的数据获取类和创建专用 TFRecord 的方法，TensorFlow 2.0 在数据读取方面可以说是如虎添翼，极大地帮助读者解决数据的获取问题。模型训练的可视化是 TensorFlow 一项特有的功能，利用这个功能能够帮助读者更好地了解模型的训练过程和可能会遇到的问题。

第 8 章介绍 ResNet 的基本思想和内容。ResNet 是一个具有里程碑性质的框架，标志着粗犷的卷积神经网络设计向着精确化和模块化的方向转化。ResNet 本身的程序编写非常简单，但是其中蕴含的设计思想却是跨越性的。

第 9 章介绍具有“注意力”的多种新型网络模型，是未来的发展方向。在不同的维度和方面上加上“注意力”是需要读者加上所有注意力的地方。

第 10 章使用经典的卷积神经网络去解决文本分类的问题。实际上，除了传统的图像处理，使用卷积神经网络还能够对文本进行分类，一般采用的是循环神经网络。文本分类也可以引申到更多的序列化问题，这也是未来深度学习研究的方向。

示例代码、数据及开发环境下载

本书示例代码、数据及开发环境下载地址请扫描下方二维码获得。

如果下载有问题或需要技术支持，请联系 booksaga@163.com，邮件主题为“TensorFlow 2.0 深度学习从零开始学”。



适合阅读本书的读者

- TensorFlow 深度学习的初学者
- 深度学习应用开发的程序员
- 高等院校相关专业的师生
- 专业培训机构的师生

作者

2020年3月

目 录

第 1 章 TensorFlow 2.0 的安装	1
1.1 Python 基本安装和用法	1
1.1.1 Anaconda 的下载与安装	1
1.1.2 Python 编译器 PyCharm 的安装	4
1.1.3 使用 Python 计算 softmax 函数	7
1.2 TensorFlow 2.0 GPU 版本的安装	8
1.2.1 检测 Anaconda 中的 TensorFlow 版本	8
1.2.2 TensorFlow 2.0 GPU 版本基础显卡推荐和前置软件安装	9
1.3 Hello TensorFlow 2.0	12
1.4 本章小结	13
第 2 章 TensorFlow 2.0 令人期待的变化	14
2.1 新的架构、新的运行、新的开始	14
2.1.1 API 精简	15
2.1.2 Eager Execution	15
2.1.3 取消全局变量	15
2.1.4 使用函数而不是会话	15
2.1.5 弃用 collection	16
2.2 配角转成主角：从 TensorFlow Eager Execution 转正谈起	16
2.2.1 Eager 简介与调用	17
2.2.2 读取数据	18
2.3 使用 TensorFlow 2.0 模式进行线性回归的一个简单例子	20
2.3.1 模型的工具与数据的生成	20
2.3.2 模型的定义	20
2.3.3 损失函数的定义	21
2.3.4 梯度函数的更新计算	21
2.4 TensorFlow 2.0 进阶——AutoGraph 和 tf.function	23
2.5 本章小结	26
第 3 章 TensorFlow 和 Keras	27
3.1 模型！模型！模型！还是模型	27
3.2 使用 Keras API 实现鸢尾花分类的例子（顺序模式）	28
3.2.1 数据的准备	29
3.2.2 数据的处理	30
3.2.3 梯度更新函数的写法	31
3.2.4 使用 Keras 函数式编程实现鸢尾花分类的例子（重点）	32

3.2.5	使用保存的 Keras 模式对模型进行复用	35
3.2.6	使用 TensorFlow 2.0 标准化编译对 iris 模型进行拟合	35
3.3	多输入单一输出 TensorFlow 2.0 编译方法 (选学)	40
3.3.1	数据的获取与处理	40
3.3.2	模型的建立	41
3.3.3	数据的组合	41
3.4	多输入多输出 TensorFlow 2.0 编译方法 (选学)	44
3.5	全连接层详解	46
3.5.1	全连接层的定义与实现	46
3.5.2	使用 TensorFlow 2.0 自带的 API 实现全连接层	47
3.5.3	打印显示 TensorFlow 2.0 设计的模型结构和参数	51
3.6	本章小结	53
第 4 章	TensorFlow 2.0 语法基础	54
4.1	BP 神经网络简介	54
4.2	BP 神经网络的两个基础算法	58
4.2.1	最小二乘法 (LS 算法)	58
4.2.2	道士下山的故事——梯度下降算法	61
4.3	反馈神经网络反向传播算法	63
4.3.1	深度学习基础	63
4.3.2	链式求导法则	64
4.3.3	反馈神经网络原理与公式推导	66
4.3.4	反馈神经网络原理的激活函数	72
4.3.5	反馈神经网络原理的 Python 实现	73
4.4	本章小结	78
第 5 章	卷积层与 MNIST 实战	79
5.1	卷积运算	79
5.1.1	卷积运算的基本概念	80
5.1.2	TensorFlow 2.0 中卷积函数的实现	81
5.1.3	池化运算	83
5.1.4	softmax 激活函数	84
5.1.5	卷积神经网络原理	86
5.2	TensorFlow 2.0 编程实战: MNIST 手写体识别	89
5.2.1	MNIST 数据集	89
5.2.2	MNIST 数据集特征和标注	91
5.2.3	TensorFlow 2.0 编程实战: MNIST 数据集	93
5.2.4	使用自定义的卷积层实现 MNIST 识别	97
5.3	本章小结	101
第 6 章	TensorFlow 2.0 Dataset 使用详解	102
6.1	Dataset API 基本结构和内容	102

6.1.1	Dataset API 数据种类.....	103
6.1.2	Dataset API 基础使用.....	104
6.2	Dataset API 高级用法.....	105
6.2.1	Dataset API 数据转换方法.....	107
6.2.2	读取图片数据集的例子.....	110
6.3	使用 TFRecord API 创建和使用数据集.....	111
6.3.1	TFRecord 的基本概念.....	112
6.3.2	TFRecord 的创建.....	113
6.3.3	TFRecord 的读取.....	118
6.4	TFRecord 实战：带有处理模型的完整例子.....	124
6.4.1	创建数据集.....	125
6.4.2	创建解析函数.....	125
6.4.3	创建数据模型.....	126
6.4.4	创建读取函数.....	126
6.5	本章小结.....	128
第 7 章	TensorFlow Datasets 和 TensorBoard 详解.....	129
7.1	TensorFlow Datasets 简介.....	129
7.1.1	Datasets 数据集的安装.....	131
7.1.2	Datasets 数据集的使用.....	131
7.2	Datasets 数据集的使用——FashionMNIST.....	133
7.2.1	FashionMNIST 数据集下载与显示.....	134
7.2.2	模型的建立与训练.....	136
7.3	使用 Keras 对 FashionMNIST 数据集进行处理.....	138
7.3.1	获取数据集.....	138
7.3.2	数据集的调整.....	139
7.3.3	使用 Python 类函数建立模型.....	139
7.3.4	模型的查看和参数的打印.....	141
7.3.5	模型的训练和评估.....	142
7.4	使用 TensorBoard 可视化训练过程.....	144
7.4.1	TensorBoard 的文件夹设置.....	145
7.4.2	显式地调用 TensorBoard.....	146
7.4.3	使用 TensorBoard.....	148
7.5	本章小结.....	152
第 8 章	从冠军开始：ResNet.....	153
8.1	ResNet 基础原理与程序设计基础.....	153
8.1.1	ResNet 诞生的背景.....	154
8.1.2	模块工具的 TensorFlow 实现——不要重复发明轮子.....	157
8.1.3	TensorFlow 高级模块 layers 的用法.....	157
8.2	ResNet 实战：CIFAR-100 数据集分类.....	165
8.2.1	CIFAR-100 数据集.....	165

8.2.2	ResNet 残差模块的实现	168
8.2.3	ResNet 网络的实现	170
8.2.4	使用 ResNet 对 CIFAR-100 数据集进行分类.....	173
8.3	ResNet 的兄弟——ResNeXt.....	175
8.3.1	ResNeXt 诞生的背景.....	175
8.3.2	ResNeXt 残差模块的实现.....	177
8.3.3	ResNeXt 网络的实现.....	178
8.3.4	ResNeXt 和 ResNet 的比较.....	180
8.4	本章小结	180
第 9 章	注意力机制	181
9.1	何为“注意力”	181
9.2	注意力机制的两种常见形式	182
9.2.1	Hard Attention (硬性注意力)	183
9.2.2	Soft Attention (软性注意力)	183
9.3	注意力机制的两种实现形式	183
9.3.1	Spatial Attention (空间注意力)	184
9.3.2	Channel Attention (通道注意力)	185
9.4	注意力机制的两种经典模型	186
9.4.1	最后的冠军——SENet.....	186
9.4.2	结合 Spatial 和 Channel 的 CBAM 模型	189
9.4.3	注意力的前沿研究——基于细粒度的图像注意力机制	194
9.5	本章小结	196
第 10 章	卷积神经网络实战：识文断字也可以.....	197
10.1	文本数据处理	198
10.1.1	数据集介绍和数据清洗	198
10.1.2	停用词的使用	201
10.1.3	词向量训练模型 word2vec 的使用.....	203
10.1.4	文本主题的提取：基于 TF-IDF (选学)	207
10.1.5	文本主题的提取：基于 TextRank (选学)	211
10.2	针对文本的卷积神经网络模型——字符卷积	214
10.2.1	字符(非单词)文本的处理	214
10.2.2	卷积神经网络文本分类模型的实现——Conv1D (一维卷积)	222
10.3	针对文本的卷积神经网络模型——词卷积	224
10.3.1	单词的文本处理	224
10.3.2	卷积神经网络文本分类模型的实现——Conv2D (二维卷积)	227
10.4	使用卷积对文本分类的补充内容	230
10.4.1	汉字的文本处理	230
10.4.2	其他的细节	233
10.5	本章小结	233

第 1 章

TensorFlow 2.0 的安装

TensorFlow 被广泛应用于各类机器学习 (Machine Learning) 算法的编程实现, 前身是谷歌的神经网络算法库 DistBelief。初学者或专家借助 TensorFlow 可以在桌面、移动、网络 and 云端环境下轻松地创建机器学习模型。正因为如此, TensorFlow 这几年非常火爆, 几乎学习 AI (Artificial Intelligence, 人工智能) 方向的开发人员都要学习它。

TensorFlow 已经从 1.x 进入 2.x。版本的更迭让它的功能更加强大。本章摒弃 TensorFlow 的历史, 直接进入 TensorFlow 2.0 的安装和第一个例子。注意: Tensor 一词在本书中对应的译文为“张量”。

1.1 Python 基本安装和用法

Python 是深度学习的首选开发语言。对于安装来说, 很多第三方都提供了集成大量科学计算类库的 Python 标准安装包, 常用的是 Anaconda。

Anaconda 里面集成了很多关于 Python 科学计算的第三方库, 主要是安装方便。Python 是一个脚本语言, 如果不使用 Anaconda, 那么第三方库的安装会较为困难, 各个库之间的依赖性很难连接得很好。因此推荐使用集合了大量第三方类库的安装程序 Anaconda 来替代 Python 的安装。

1.1.1 Anaconda 的下载与安装

1. 第一步: 下载和安装

Anaconda 官方的下载地址是 <https://www.continuum.io/downloads/> (见图 1.1, 不推荐使用 Python 3.7 版)。



图 1.1 Anaconda 下载页面

这里提供的是集成了 Python 3.7 版本的 Anaconda 下载。建议选用 Python 3.6 版本的 Anaconda，如果选用 3.7 版本，那么在使用 TensorFlow 2.0 的时候可能会出现一些未知原因的错误，需要花时间去排除。

(1) 集成 Python 3.6 版本的 Anaconda 可以在清华大学 Anaconda 镜像网站下载，地址为 <https://mirrors.tuna.tsinghua.edu.cn/anaconda/archive/>，打开后如图 1.2 所示。

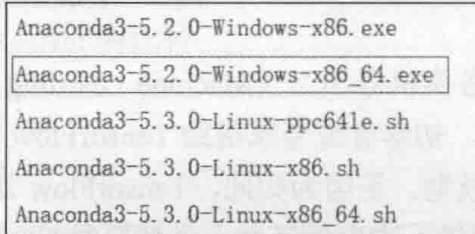


图 1.2 清华大学 Anaconda 镜像网站提供的副本

注意

选择以 Anaconda3 开头、以 64 结尾的安装文件，不要下载错了！

(2) 下载完成后获得 exe 文件，直接运行即可安装，与普通软件一样。安装完成以后，出现如图 1.3 所示的目录结构，说明安装正确。



图 1.3 Anaconda 安装目录

2. 第二步：打开控制台

依次单击“开始→所有程序→Anaconda→Anaconda Prompt”。这些步骤和打开 Windows 下的 CMD 控制台类似，输入命令就可以控制和配置 Python。在 Anaconda 中常用的是 conda 命令。该命令可以执行一些基本操作，如更新某个包：

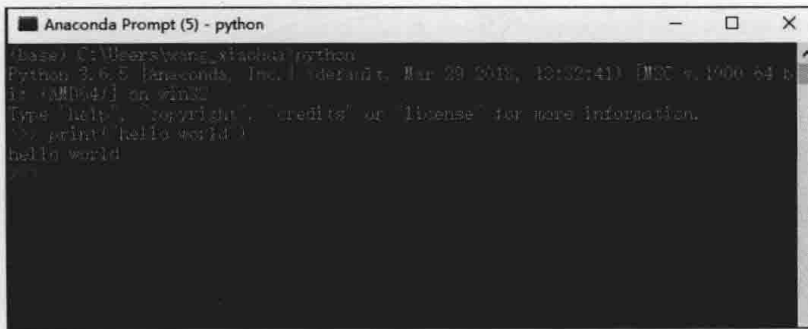
```
conda update name
```

3. 第三步：验证 Python

在控制台中输入“python”，若安装正确则会打印出版版本号，以及 Python 的控制符号>>>。在控制符号下输入代码：

```
print("hello world")
```

输出结果如图 1.4 所示。



```

Anaconda Prompt (5) - python
(base) C:\Users\ywang\Anaconda\python
Python 3.6.5 [Anaconda, Inc.] (default, Mar 29 2018, 13:32:41) [MSC v.1900 64 bit
(x86_64)] on win32
Type help(), copyright(), credits() or license() for more information.
>>> print('hello world')
hello world

```

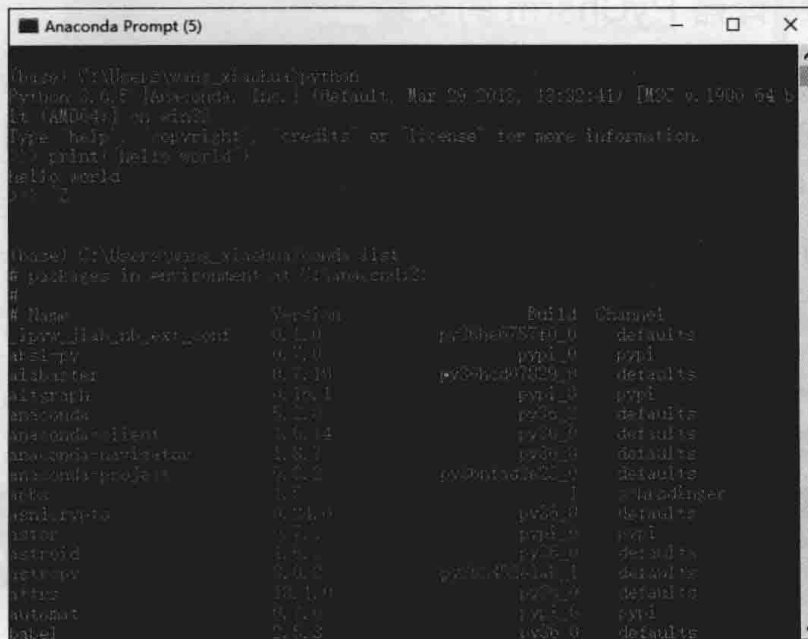
图 1.4 验证 Anaconda Python 安装成功

4. 使用 conda 命令

建议使用 Anaconda 的好处在于能够极大地帮助读者安装和使用大量第三方类库。查看已安装的第三方类库的命令是：

```
conda list
```

在 Anaconda Prompt 控制台中输入 `exit()` 或者重新打开 Anaconda Prompt 控制台后直接输入 `conda list` 命令，执行结果如图 1.5 所示。



```

Anaconda Prompt (5)
(base) C:\Users\ywang\Anaconda\python
Python 3.6.5 [Anaconda, Inc.] (default, Mar 29 2018, 13:32:41) [MSC v.1900 64 b
(x86_64)] on win32
Type help(), copyright(), credits() or license() for more information.
>>> print('hello world')
hello world
>>>

(base) C:\Users\ywang\Anaconda> conda list
# packages in environment at C:\anaconda2:
#
# Name                    Version           Build    Channel
libm2crypto               0.1.0             py36he67870_0  defaults
alibi-ipy                 0.5.0             py36_0     pypi
alibi-text                 0.7.10           py36hc397829_0  defaults
altgraph                  0.16.1           py36_0     pypi
anaconda                   5.2.0            py36_0     defaults
anaconda-client           1.6.14           py36_0     defaults
anaconda-navigator        1.8.7            py36_0     defaults
anaconda-profiler         0.6.2            py36nta53e35_0  defaults
arbit                      1.5              py36_0     conda-forge
conda-verify              0.2.10           py36_0     defaults
astor                      0.7.0            py36_0     pypi
astroid                    1.6.0            py36_0     defaults
astropy                   0.4.0            py36h4f6e1a1_1  defaults
attrs                      18.1.0           py36_0     defaults
autogat                    0.1.0            py36_0     pypi
babel                      2.5.3            py36_0     defaults

```

图 1.5 列出已安装的第三方类库

Anaconda 中使用 `conda` 进行操作的方法还有很多，其中最重要的是安装第三方类库，命令如下：

```
conda install name
```

这里的 `name` 是需要安装的第三方类库名，例如安装 NumPy 包，那么输入的命令就是：

```
conda install numpy
```

假如之前已经安装过 NumPy 包，那么该命令执行后会自动获取一些新的依赖库，并询问是否更新，如图 1.6 所示。

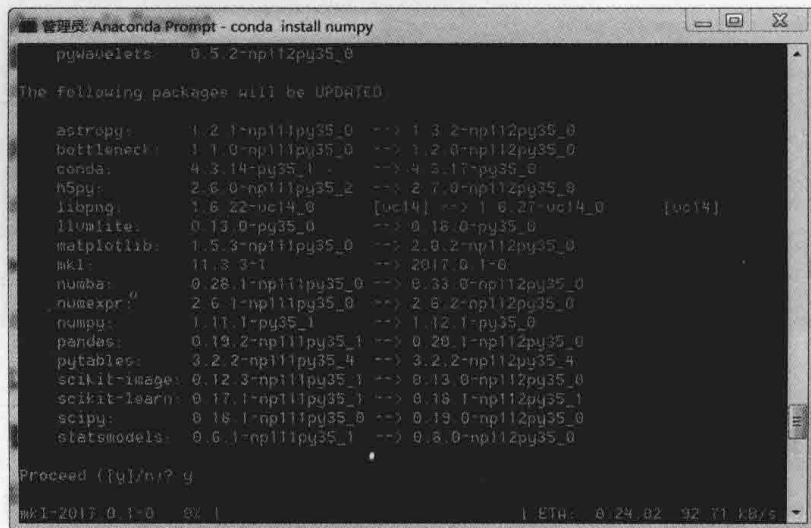


图 1.6 自动获取或更新依赖类库

使用 Anaconda 的好处就是默认安装好了大部分学习所需的第三方类库，从而大大减少了使用者在安装和使用某个特定类库时碰到依赖类库缺失的情况。

1.1.2 Python 编译器 PyCharm 的安装

和其他语言类似，Python 程序可以使用 Windows 自带的控制台进行编写。对于较为复杂的程序工程来说，这种方式容易混淆相互之间的层级和交互文件，因此在编写程序工程时建议使用专用的 Python 编译器 PyCharm。

1. 第一步：PyCharm 的下载和安装

PyCharm 的下载地址为 <http://www.jetbrains.com/pycharm/>。

(1) 进入 Download 页面后可以选择不同的版本（收费的专业版和免费的社区版），如图 1.7 所示。这里选择免费的社区版即可。

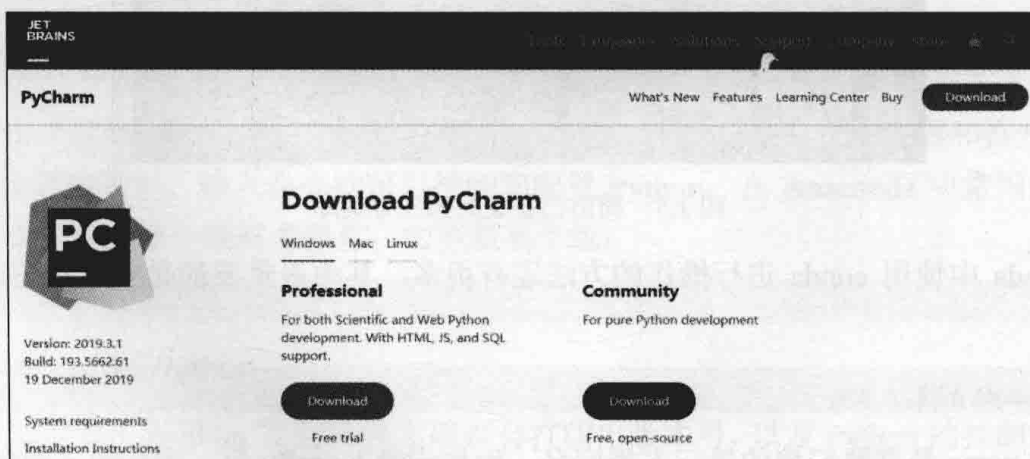


图 1.7 选择 PyCharm 的免费版

(2) 双击下载好的 PyCharm 安装文件，运行后进入安装界面，如图 1.8 所示。直接单击 Next 按钮，采用默认安装方式。

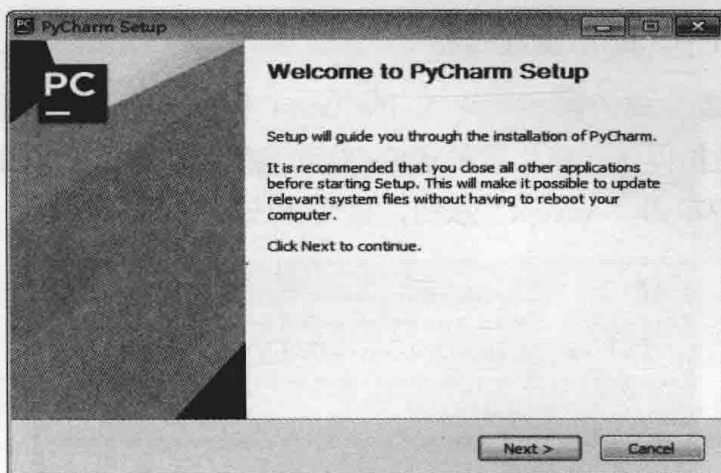


图 1.8 PyCharm 的安装界面

(3) 进入配置选择界面，如图 1.9 所示。需要注意的是，在安装 PyCharm 的过程中要对安装的版本进行选择，即选择 32 位的版本还是选择 64 位的版本，建议选择与 Python 位数相同的版本。

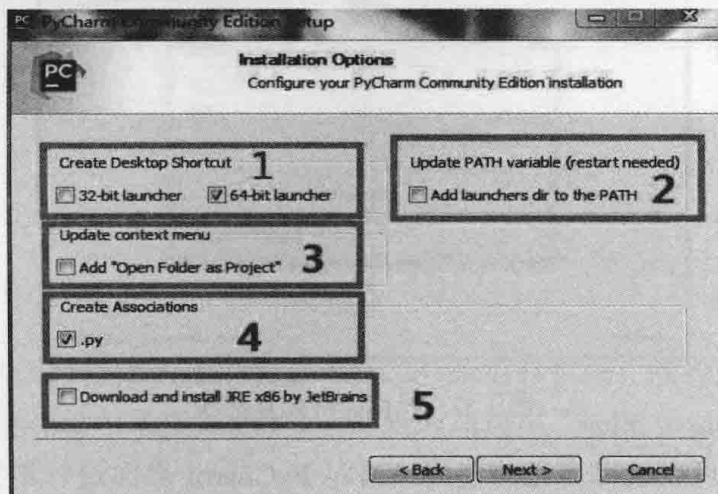


图 1.9 PyCharm 的配置选择（按个人真实情况选择）

(4) 最后出现安装完成的界面，如图 1.10 所示。单击“Finish”按钮完成安装。

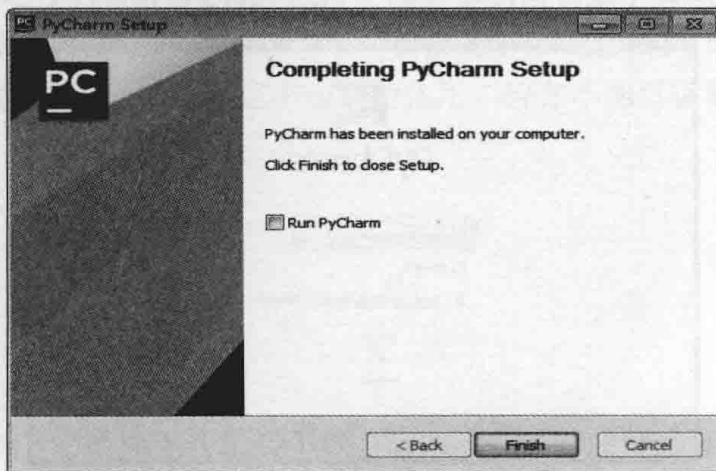



图 1.10 PyCharm 安装完成

2. 第二步：使用 PyCharm 创建程序

(1) 单击桌面上新生成的  图标进入 PyCharm 程序界面，首先是第一次启动的定位，如图 1.11 所示。这里是指程序存储位置的定位，建议选择第 2 个：由 PyCharm 自动指定即可。之后在弹出的对话框中单击“Accept”按钮，接受相应的协议。

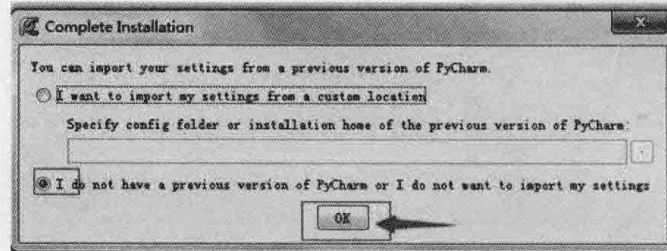


图 1.11 PyCharm 启动定位

(2) 接受协议后进入配置选项界面，如图 1.12 所示。

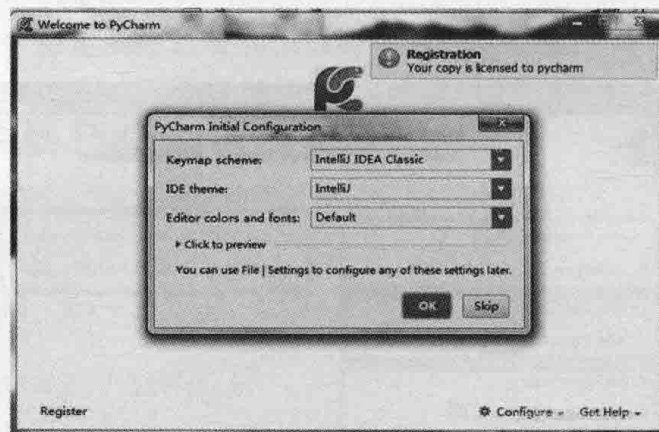


图 1.12 PyCharm 配置选项

(3) 在配置区域可以选择自己的使用风格对 PyCharm 界面进行配置。如果对其不熟悉，直接单击 OK 按钮，使用默认选项即可。

(4) 创建一个新的工程，如图 1.13 所示。建议新建一个 PyCharm 的工程文件，结果如图 1.14 所示。

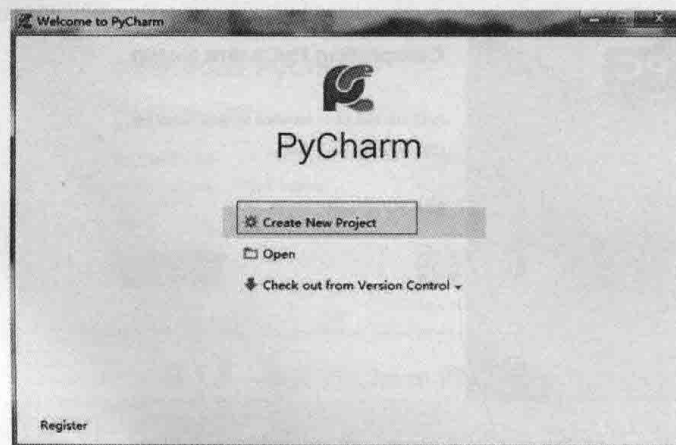


图 1.13 PyCharm 工程创建界面