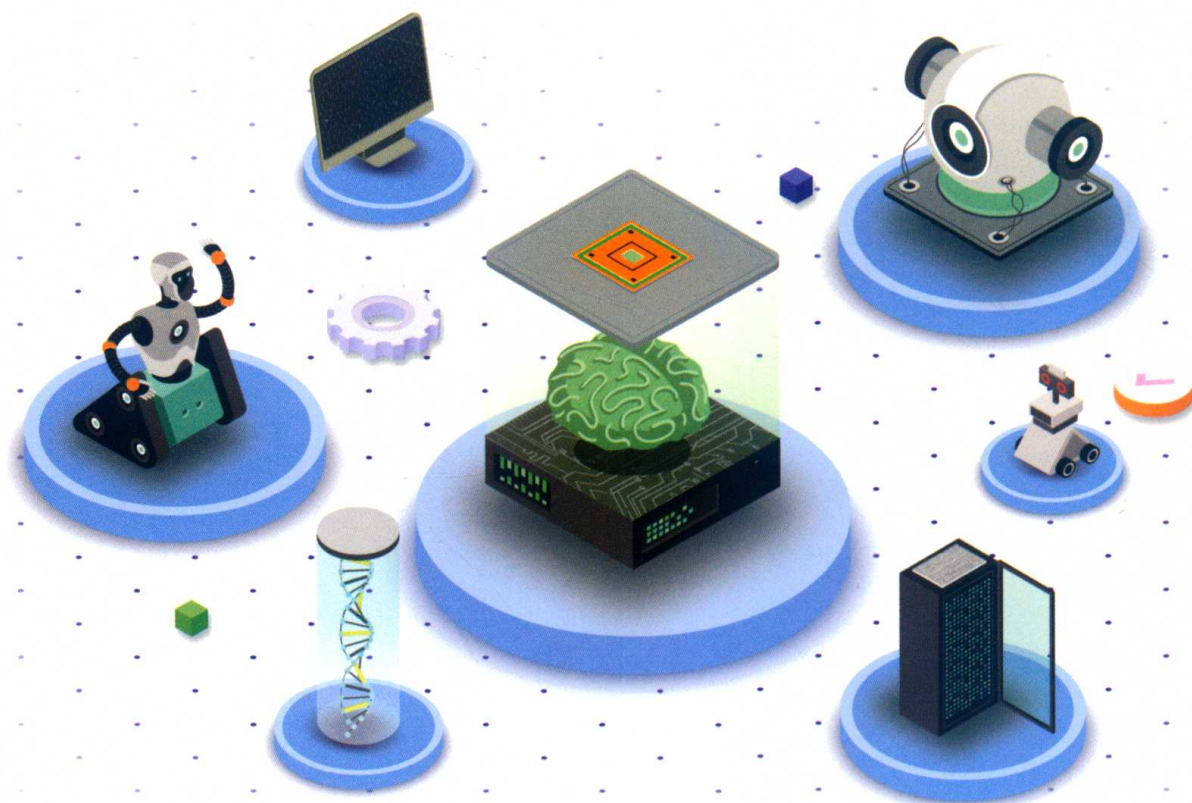


Deep Learning
Convolutional Neural Network
Technology and Practice

深度学习

卷积神经网络技术与实践

高敬鹏 著



机械工业出版社
China Machine Press

■ ■ ■ 智能系统与技术丛书

Deep Learning
Convolutional Neural Network
Technology and Practice

深度学习

卷积神经网络技术与实践

高敬鹏 著



机械工业出版社
China Machine Press

图书在版编目 (CIP) 数据

深度学习：卷积神经网络技术与实践 / 高敬鹏著. —北京：机械工业出版社，2020.6
(智能系统与技术丛书)

ISBN 978-7-111-65737-8

I. 深… II. 高… III. 机器学习 IV. TP181

中国版本图书馆 CIP 数据核字 (2020) 第 096652 号

深度学习：卷积神经网络技术与实践

出版发行：机械工业出版社（北京市西城区百万庄大街 22 号 邮政编码：100037）

责任编辑：陈佳媛

责任校对：李秋荣

印刷：北京文昌阁彩色印刷有限责任公司

版次：2020 年 7 月第 1 版第 1 次印刷

开本：186mm×240mm 1/16

印张：18

书号：ISBN 978-7-111-65737-8

定价：79.00 元

客服热情：(010) 88361066 88379833 68326294

投稿热线：(010) 88379604

华章网站：www.hzbook.com

读者信箱：hzit@hzbook.com

版权所有·侵权必究

封底无防伪标均为盗版

本书法律顾问：北京大成律师事务所 韩光 / 邹晓东

Preface 前 言

深度学习是人工智能领域中较为热门的一种机器学习技术。人工智能领域主要研究如何让机器完成通常利用人类智能才能完成的复杂工作，这是一个对人的意识、思维进行模拟和学习的过程。深度学习模仿人类神经网络的工作方式，广泛应用于图像识别、语音识别、自然语言处理等多个领域。常见的人工神经网络（Artificial Neural Network, ANN）有多层感知器（MultiLayer Perceptron, MLP）、卷积神经网络（Convolutional Neural Network, CNN）和循环神经网络（Recurrent Neural Network, RNN）等。

本书在 Windows 系统下，通过 Anaconda 搭建虚拟环境，以 TensorFlow 为框架，使用 Python 语言，详细阐述了深度学习、强化学习与深度强化学习的实现方法。本书共 11 章，主要内容包括深度学习简介、Python 基础、神经网络基础、卷积神经网络、经典卷积网络结构、迁移学习、循环神经网络、强化学习、深度强化学习、基于策略的算法更新与趋势等，通过具体案例，将 Python 语言、深度学习思想、强化学习思想和实际工程完美地结合起来。

为了使初学者提高对人工智能的兴趣，并在短时间内掌握深度学习与强化学习的要点，作者在编写过程中注重内容的选择，使本书具有以下特点。

- **由浅入深，循序渐进：**在内容编排上遵循由浅入深、由易到难的原则，将基础知识与大量实例相结合，边学边练。
- **实例丰富，涉及面广：**提供了丰富的 Python 程序设计实例，内容涉及深度学习与强化学习等。
- **兼顾原理，注重实用：**精简理论知识，在介绍深度学习与强化学习理论知识的同时，更注重实际应用。

本书由哈尔滨工程大学的高敬鹏撰写，其中第 3~8 章和第 9~11 章的程序调试分别由哈尔滨工程大学的王旭和王晨悦完成。此外，为本书撰写工作提供帮助的还有宋一兵、管殿柱等，在此对他们表示衷心的感谢。

感谢读者选择了本书，希望本书对读者的工作和学习有所帮助。由于作者水平和经验有限，书中疏漏之处在所难免，敬请读者指正。我的电子邮箱是：heu_tongxin@163.com。

高敬鹏

2020年1月于哈尔滨工程大学

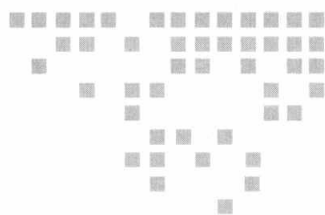
Contents 目 录

前言	
第 1 章 深度学习简介 1	
1.1 机器学习与深度学习..... 1	
1.2 TensorFlow 概述..... 2	
1.3 环境搭建..... 3	
1.3.1 在 Windows 系统下安装 Anaconda..... 3	2.4.3 for 循环..... 20
1.3.2 在 Anaconda 下安装 TensorFlow 与 Keras..... 6	2.4.4 break 和 continue..... 21
1.3.3 Spyder 编辑器..... 8	2.5 列表与元组..... 23
	2.5.1 创建..... 23
	2.5.2 查询..... 24
	2.5.3 修改..... 24
	2.5.4 删除..... 26
	2.6 字典..... 26
	2.6.1 字典的创建..... 27
	2.6.2 字典的常规操作..... 27
	2.6.3 字典的遍历..... 29
	2.7 函数..... 29
	2.7.1 函数的定义与调用..... 30
	2.7.2 参数传递..... 31
第 2 章 Python 基础 11	2.8 面向对象编程..... 33
2.1 数据类型..... 11	2.8.1 类与对象..... 33
2.1.1 数值类型..... 11	2.8.2 继承与多态..... 34
2.1.2 字符串类型..... 13	2.9 思考与练习..... 37
2.1.3 布尔类型..... 13	
2.2 变量与常量..... 14	
2.3 运算符..... 14	第 3 章 神经网络基础 38
2.3.1 运算符概述..... 14	3.1 单层神经网络..... 38
2.3.2 运算符优先级..... 15	3.2 多层神经网络..... 39
2.4 选择与循环..... 15	3.2.1 隐藏层..... 39
2.4.1 if 语句..... 15	3.2.2 输入层与输出层..... 41
2.4.2 while 循环..... 18	

3.3 激活函数	42	4.3.1 MNIST 数据集预处理	76
3.3.1 Sigmoid 函数	42	4.3.2 简单卷积神经网络搭建	77
3.3.2 Tanh 函数	43	4.4 CIFAR-10 数据集	84
3.3.3 Relu 函数	44	4.4.1 下载 CIFAR-10 数据集	85
3.3.4 Softmax 函数	45	4.4.2 CIFAR-10 数据集预处理	87
3.4 神经网络工作过程	45	4.5 简单卷积神经网络实现 CIFAR-10	
3.5 损失函数	47	分类	88
3.5.1 均方差函数	47	4.6 思考与练习	93
3.5.2 交叉熵函数	47		
3.6 优化算法	48	第 5 章 经典卷积网络结构	94
3.7 反向传播	49	5.1 LeNet 概述	94
3.8 泛化能力	52	5.2 LeNet 实现 MNIST 分类	95
3.9 多层感知器	53	5.2.1 MNIST 数据预处理	95
3.10 MNIST 数据集	54	5.2.2 基于 Keras 搭建 LeNet 网络	
3.10.1 下载 MNIST 数据集	54	结构	95
3.10.2 数据预处理	56	5.2.3 对 LeNet 网络模型进行编译、	
3.11 Keras 实现感知器的手写体识别	58	训练、评估与预测	98
3.11.1 单层感知器手写体识别	58	5.3 AlexNet 概述	101
3.11.2 多层感知器手写体识别	61	5.4 AlexNet 实现 MNIST 分类	103
3.12 思考与练习	67	5.4.1 基于 Keras 搭建 AlexNet 网络	
		结构	103
第 4 章 卷积神经网络	68	5.4.2 对 AlexNet 网络模型进行	
4.1 卷积神经网络结构及原理	68	编译、训练、评估与预测	107
4.1.1 卷积神经网络特点	69	5.5 VGG16 概述	110
4.1.2 卷积层	70	5.6 VGG16 实现 MNIST 分类	111
4.1.3 池化层	72	5.6.1 基于 Keras 搭建 VGG16 网络	
4.1.4 全连接层	73	结构	112
4.2 卷积神经网络工作过程	74	5.6.2 对 VGG16 网络模型进行评估	
4.3 简单卷积神经网络实现 MNIST		与预测	115
分类	76	5.7 思考与练习	117

第 6 章 经典卷积网络结构进阶	118	8.4 简单 RNN 实现 Reuters 分类	180
6.1 GoogLeNet 概述	118	8.5 LSTM 实现 Reuters 分类	185
6.2 GoogLeNet 实现 MNIST 分类	119	8.6 思考与练习	190
6.2.1 基于 Keras 搭建 GoogLeNet 网络结构	119	第 9 章 强化学习	191
6.2.2 对 GoogLeNet 进行训练、 评估与预测	125	9.1 初识强化学习	191
6.3 ResNet 概述	129	9.1.1 什么是强化学习	192
6.4 ResNet50 实现 MNIST 分类	131	9.1.2 强化学习能解决什么 类型的问题	193
6.4.1 基于 Keras 搭建 ResNet50 网络 结构	131	9.1.3 强化学习如何解决问题	194
6.4.2 对 ResNet50 网络模型进行 训练、评估与预测	138	9.2 强化学习理论基础	194
6.5 思考与练习	142	9.2.1 基本组成元素	194
第 7 章 迁移学习	143	9.2.2 基本模型	196
7.1 基于卷积网络实现迁移学习	143	9.2.3 价值函数	198
7.2 InceptionV3 实现迁移学习	144	9.3 求解强化学习——有模型	199
7.3 Xception 实现迁移学习	150	9.3.1 动态规划与贝尔曼方程	199
7.4 MobileNet 实现迁移学习	155	9.3.2 策略迭代	200
7.5 简单卷积网络实现迁移学习	164	9.3.3 值迭代	202
7.6 思考与练习	172	9.3.4 值迭代算法实现格子世界	202
第 8 章 循环神经网络	173	9.4 求解强化学习——无模型	208
8.1 循环神经网络概述	173	9.4.1 蒙特卡罗算法	208
8.2 长短期记忆网络	174	9.4.2 时间差分法	209
8.2.1 LSTM 前向传播	175	9.4.3 Q-learning 算法	210
8.2.2 LSTM 反向传播	176	9.4.4 Q-learning 实现格子世界	211
8.3 Reuters 数据集	176	9.5 思考与练习	213
8.3.1 Reuters 数据集概述	176	第 10 章 深度强化学习	214
8.3.2 文本信息预处理	177	10.1 深度强化学习框架	214
		10.2 TensorFlow 编程	216
		10.2.1 TensorFlow 的计算模型—— 计算图	216

10.2.2 TensorFlow 的数据模型—— 张量	219	11.1.1 策略目标函数	251
10.2.3 TensorFlow 的运行模型—— 会话	220	11.1.2 策略梯度定理	252
10.2.4 TensorFlow 变量	222	11.1.3 策略梯度算法实现	254
10.2.5 TensorFlow 共享变量	225	11.2 演员 – 评论家算法	258
10.3 Gym 的安装及使用	226	11.2.1 演员 – 评论家算法原理	258
10.3.1 Gym 的安装	226	11.2.2 演员 – 评论家算法实现	259
10.3.2 Gym 的使用	227	11.3 优势演员 – 评论家算法	263
10.4 基于值的算法更新	229	11.3.1 优势演员 – 评论家算法 原理	264
10.4.1 Q-learning 实现	229	11.3.2 优势演员 – 评论家算法 实现	265
10.4.2 DQN 算法原理	233	11.4 异步优势演员 – 评论家算法	269
10.4.3 DQN 算法实现	236	11.4.1 异步优势演员 – 评论家 算法原理	270
10.4.4 DDQN 算法原理	241	11.4.2 异步优势演员 – 评论家算法 实现	272
10.4.5 DDQN 算法实现	243	11.5 深度强化学习的发展趋势	278
10.5 思考与练习	248	11.6 思考与练习	279
第 11 章 基于策略的算法更新与 趋势	250		
11.1 策略梯度法	250		



深度学习简介

1.1 机器学习与深度学习

1. 机器学习

早在 20 世纪 50 年代，人们首次提出了人工智能（Artificial Intelligence, AI）。人工智能领域主要研究如何让机器完成通常由人类智能才能完成的复杂工作，这是一个对人的意识和思维进行模拟和学习的过程。

机器学习是人工智能的一个分支，它使用大量的数据通过算法进行训练进而得到相应的模型。当出现新的数据时，可通过训练完毕的模型对新数据进行预测。机器学习又可分为 3 类：有监督学习（Supervised Learning）、无监督学习（Unsupervised Learning）和强化学习（Reinforcement Learning）。

- 有监督学习：对于所有的训练样本，每个样本的标签都是人为标注的，即所有样本标签都是已知的。通过这些已知标签的样本去训练模型，将输入映射为相应的输出，使得模型拥有对训练样本识别分类的能力。有监督学习可分为统计分析与回归分析，其中统计分析又可分为二分类问题和多分类问题。
- 无监督学习：对于所有的训练样本，每个样本的标签都是未知的。对于需要训练的模型来说，只能根据样本间的相似性，将相似性高的样本分成一类，而无法进一步预测到底是哪一类别。
- 强化学习：强化学习强调如何基于环境而行动，以取得最大化的预期利益。强化学

习能够让智能机器人在未知的环境中进行自我决策，并且这个决策过程不是间断性的，而是可以长期做出连续性的决策。也就是说只要涉及智能决策的问题，并且在符合强化学习的规则的情况下都可以应用。智能决策也就是在环境中连续不断地做出决策。在实际生活中，强化学习多应用于游戏博弈，最受瞩目的就是在围棋比赛中的 AlphaGO。在比赛前期它通过不断自我博弈、与他人博弈的强化学习，最终在与人类的对决中胜出。

2. 深度学习

深度学习模仿人类神经网络的工作方式，广泛应用于图像识别、语音识别、自然语言处理等多个领域。深度学习作为机器学习的一个分支，同样可以广泛应用于机器学习，如有监督学习、无监督学习和强化学习。目前，深度学习主要通过搭建神经网络架构的方式来实现，常用的神经网络有深度神经网络（Deep Neural Network, DNN）、卷积神经网络（Convolutional Neural Network, CNN）和循环神经网络（Recurrent Neural Network, RNN）。神经网络搭建完成后，通过大量的样本对该神经网络进行训练，在训练的过程中不断优化网络层中的超参数，训练结束后便可得到神经网络模型，用该网络模型便可对新的样本进行预测。

1.2 TensorFlow 概述

目前深度学习主要通过搭建神经网络架构的方式来实现。在深度学习中，数据的运算都是以张量（矩阵）形式来完成的，进而通过张量的运算来模拟神经网络。由 Google 公司开发的 TensorFlow 可以让张量运算达到最高性能。TensorFlow 是 Google 的开源机器学习库，主要用于机器学习、深度学习等方面的研究。TensorFlow 的通用性也可以使其应用于其他领域。

TensorFlow 拥有非常强大与灵活的功能，包括：

- ❑ TensorFlow 在 CPU、GPU 和 TPU 中都可执行。
- ❑ 在不修改代码的前提下，TensorFlow 可以在不同平台上执行，如 Windows、Linux、Android、iOS 和 Raspberry Pi 等。
- ❑ TensorFlow 具备分布式计算能力，可同时在数百台计算机上训练神经网络模型，大大缩短模型训练时间。
- ❑ 对于 TensorFlow 的执行语言来说，Python 的支持是最好的，也可以选择 C++ 等语言。关于 Python 的详细内容将在第 2 章中介绍。
- ❑ TensorFlow 在实现深度学习时，可以自行设计各种深度网络，较为灵活，但随之而来的是需要花更多的时间编写更多的程序。还可以选择以 TensorFlow 开发的高级

API, 较为常用的高级 API 有 Keras。本书将主要以 Keras 实现深度学习, 还可以选择 TF-Learn、TF-Slim 等 API。

关于 TensorFlow 编程的详细内容将在第 10 章中介绍。

1.3 环境搭建

Anaconda 是 Python 的一个开源发行版本, 其包含了 conda、Python 等 180 多个科学包及其依赖项。本节将介绍如何在 Windows 系统下安装 Anaconda, 如何在 Anaconda 虚拟环境下搭建 TensorFlow 与 Keras, 以及一些常用编辑器的安装方法。

1.3.1 在 Windows 系统下安装 Anaconda

Anaconda 可以从其官网 (<https://www.anaconda.com/>) 下载并安装。由于 TensorFlow 需要 64 位的 Python 作为支持, 所以在官网中需要选择 Windows 系统下的 64 位的 Python 3.7 版本的安装包进行下载。

安装包下载完成后, 在相应文件夹中双击 .exe 文件, 出现如图 1-1 所示 Anaconda 安装界面。

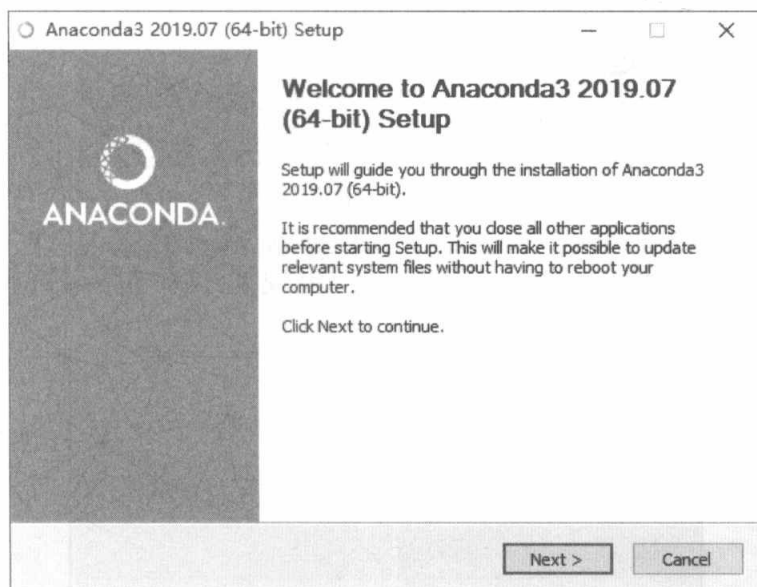


图 1-1 Anaconda 安装界面

单击 Next, 出现如图 1-2 所示许可协议界面,

单击 I Agree 按钮, 出现如图 1-3 所示选择安装类型界面。

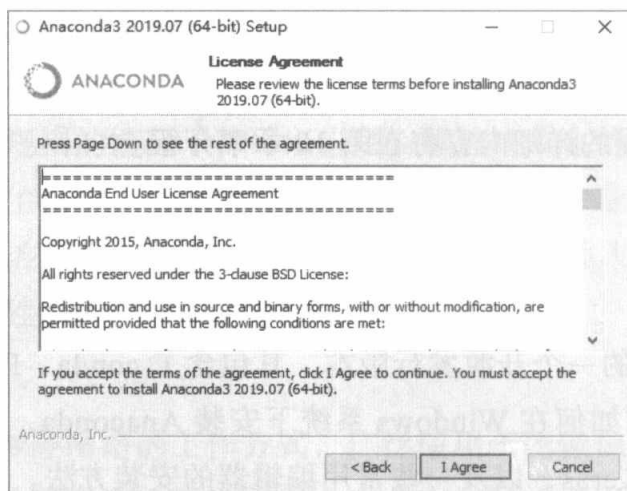


图 1-2 许可协议界面

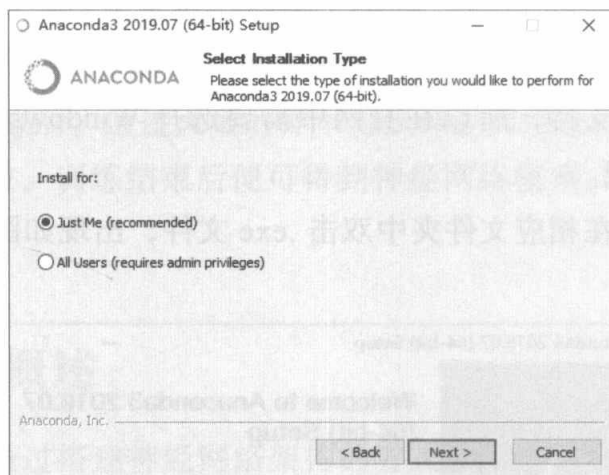


图 1-3 选择安装类型界面

选择两个单选项之一，然后单击 Next 按钮，出现如图 1-4 所示选择安装地址界面。

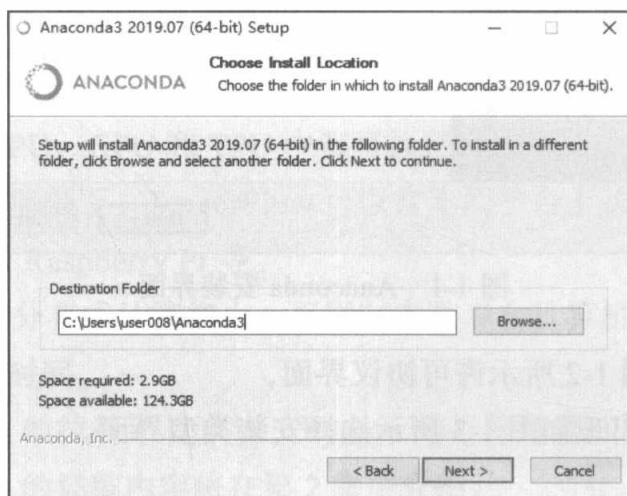


图 1-4 选择安装地址界面

安装地址默认为 C 盘的用户目录，也可以自行选择。单击 Next 按钮出现如图 1-5 所示高级安装选项界面。

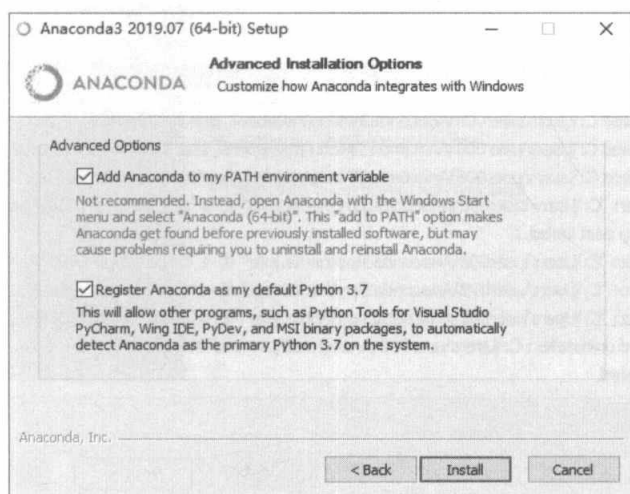


图 1-5 高级安装选项界面

勾选“Add Anaconda to my PATH environment variable (将 Anaconda 添加到我的路径环境变量)”，这一选项默认将用户变量直接添加上了，后续不用再添加。勾选“Register Anaconda as my default Python 3.7 (将 Anaconda 注册为默认的 Python 3.7)”。最后单击 Install 按钮进行安装，出现如图 1-6 所示安装界面。

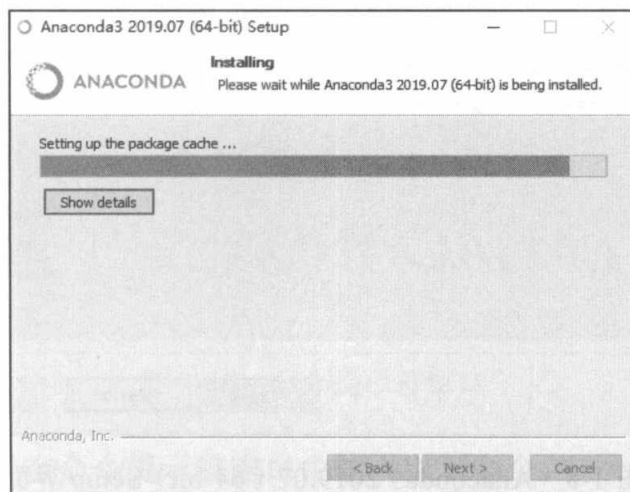


图 1-6 安装界面

安装完成后的界面如图 1-7 所示。

安装完成后，单击 Next 按钮，出现如图 1-8 所示 Anaconda3 2019.07 (64-bit) Setup 界面。

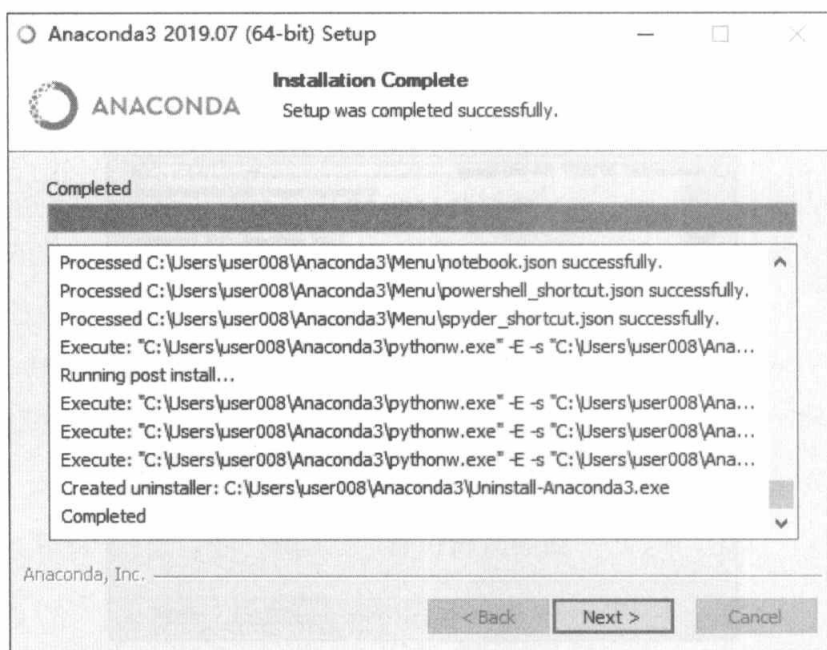


图 1-7 安装完成界面

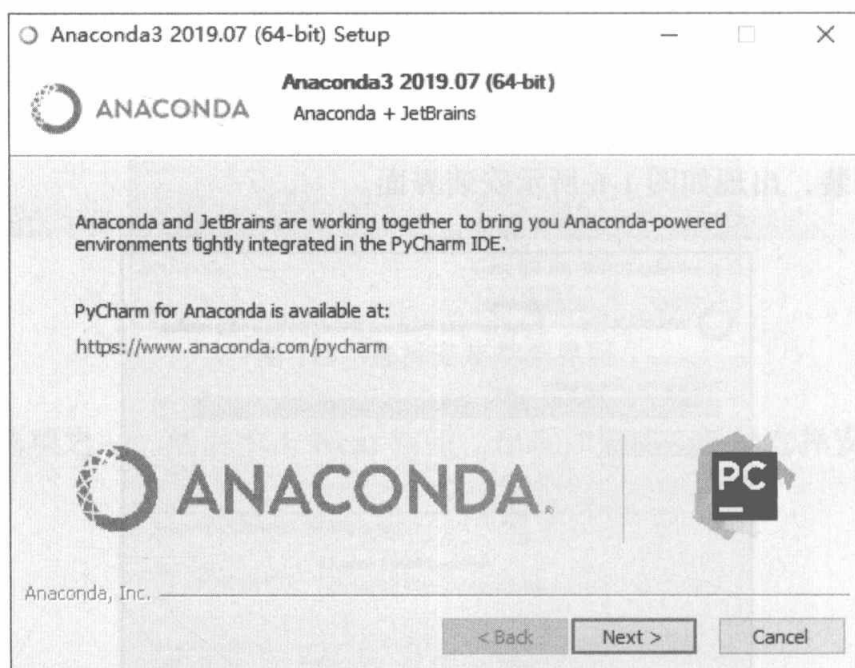


图 1-8 Anaconda3 2019.07 (64-bit) Setup 界面

单击 Next 按钮，出现如图 1-9 所示安装结束界面。单击 Finish 按钮完成安装。

1.3.2 在 Anaconda 下安装 TensorFlow 与 Keras

Anaconda 安装完成后，可进一步安装 TensorFlow 与 Keras。运行 Anaconda Prompt，出现命令提示符窗口，如图 1-10 所示。

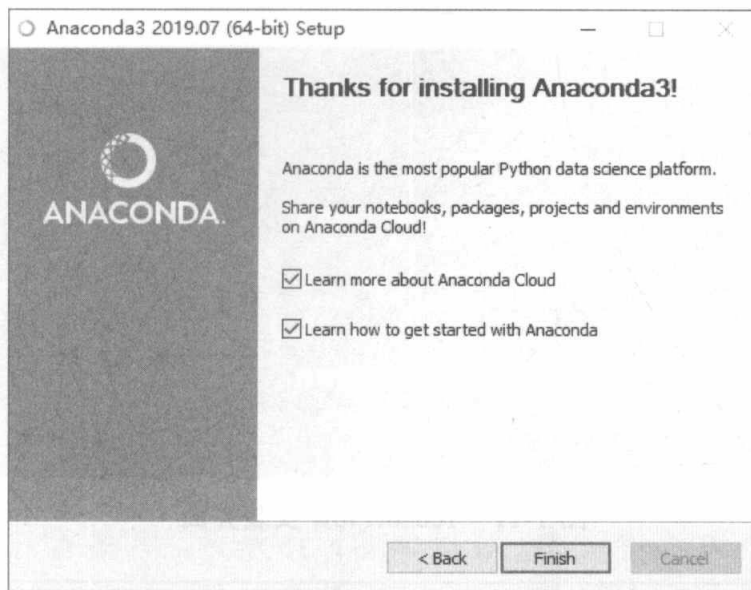


图 1-9 安装结束界面

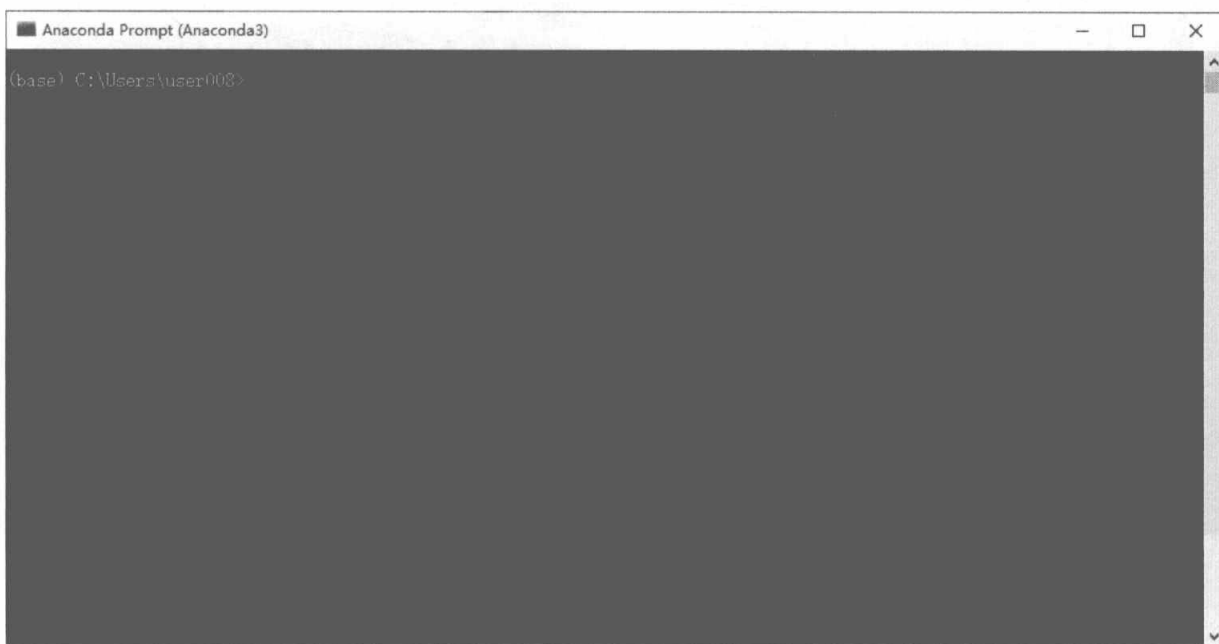


图 1-10 命令提示符窗口

安装 TensorFlow 需要在命令提示符窗口中输入以下命令：

```
pip install tensorflow
```

TensorFlow 安装界面如图 1-11 所示。等待安装结束即可。

安装 Keras 需要在命令提示符窗口中输入以下命令：

```
pip install keras
```

Keras 安装界面如图 1-12 所示。

