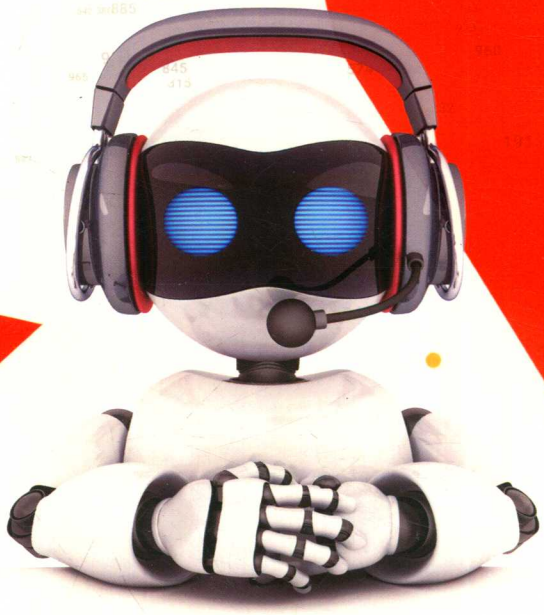


TensorFlow

智能算法与应用

◎ 胡　鹤　编著

中国工信出版集团
電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

TensorFlow 智能算法与应用

胡 鹤 编著

電子工業出版社
Publishing House of Electronics Industry
北京 • BEIJING

内 容 简 介

TensorFlow 是目前最受关注的机器学习框架，其模块化设计非常适合大数据环境下智能算法的开发与应用。本书介绍了使用 TensorFlow 进行智能算法的实践，包括经典的机器学习算法和深度学习算法实现。本书力求做到理论与实践平衡统一，在相关理论上深入浅出，辅以多种 TensorFlow 实现技术对理论进行具体实践，有助于读者快速理解与掌握智能算法的精髓和 TensorFlow 技术的要点。

本书共 4 篇。入门篇介绍学习环境搭建和 TensorFlow 框架的基本使用；基础篇介绍传统智能算法及其 TensorFlow 的实现；进阶篇介绍深度神经网络方法和 CNN、RNN、LSTM、GRU 等基础的深度学习算法；应用篇介绍 GAN 学习算法和 TensorFlow Hub 迁移学习。

本书适合智能算法初学者入门学习，同时也适合作为计算机及相关专业学生的教材和上机指导书。

图书在版编目（CIP）数据

TensorFlow 智能算法与应用/胡鹤编著. —北京：电子工业出版社，2019.7

ISBN 978-7-121-36899-8

Ⅰ. ①T… Ⅱ. ①胡… Ⅲ. ①人工智能－算法－研究 Ⅳ. ①TP18

中国版本图书馆 CIP 数据核字（2019）第 120394 号

责任编辑：张 迪（zhangdi@phei.com.cn）
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：787×1 092 1/16 印张：15 字数：384 千字
版 次：2019 年 7 月第 1 版
印 次：2019 年 7 月第 1 次印刷
定 价：59.00 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）88254469，zhangdi@phei.com.cn。

前 言

2017 年 12 月 15 日，美国宇航局宣布首次发现在 2000 多光年之外的一个名为“开普勒 90”的恒星周围有 8 颗行星组成的行星系统，构成了另一个“太阳系”。值得注意的是，这个重大的天文发现并不是由人类直接发现的，而是借助了谷歌公司的 TensorFlow 系统实现的智能算法，对开普勒望远镜获得的海量恒星亮度数据进行分析得到的。该智能算法成功地学会了如何从海量天文数据中搜寻疑似的系外行星信号，对于系外行星信号的判断正确率达到了 96%以上。该算法最终成功帮助人类定位出“开普勒 90”太阳系。TensorFlow 已经在不知不觉中改变了我们的世界。澳大利亚的科学家用 TensorFlow 开发的图像识别模型，在数万张的海洋航拍照片中，可以快速并且准确地找到珍惜的海牛。也有科学家利用 TensorFlow 把语音处理技术用到鸟类保护上，他们在丛林里安装了很多话筒，采集鸟类的声音，智能模型就可以很准确地估算出鸟类在一片森林中的数量，从而可以更加精准地对鸟类实行保护。

国内很多公司都在应用 TensorFlow 开发智能应用，京东内部搭建了 TensorFlow 训练平台，用于开发图像、自然语言相关的模型，并且把它们用到客服广告等领域。小米也在尝试类似的技术路线，支持他们生态线上各种特殊的应用。网易的有道笔记、有道翻译官也使用了 TensorFlow 视觉和语言的模型。中国电信在其营业厅 APP 应用中开发了充值卡扫描项目，使用 TensorFlow 搭建了 CNN + LSTM + CTC 的识别模型，使得用户打开摄像头对准充值卡密码轻松一扫即可完成充值。随着国家《新一代人工智能发展规划》的推动，在可预期的将来，还会有越来越多的公司和科研单位投入到 TensorFlow 的应用中来。

本书采用实例驱动的方式介绍 TensorFlow 框架下的智能算法开发。介绍重要的知识点（如线性回归模型、逻辑回归模型、CNN、RNN、LSTM、GAN 等内容）时，紧接着就有对应代码来验证和解释算法的构造与运行过程。对这些经典的智能算法，本书提供了多种 TensorFlow 的框架技术实现案例，包括 TensorFlow 原生模型、TensorFlow Estimator 模型和 TensorFlow Keras 模型等。通过对比不同的实现技术，可以更好地理解 TensorFlow 开发智能算法的技术路线。“工欲善其事，必先利其器”，学习人工智能最主要的任务就是理解智能算法的基础原理，掌握其实现技术并能够应用到自己的项目中去。本书的目的是力求通过官方权威资料，理论与实战项目相结合，使读者在练习中熟练掌握利用 TensorFlow 快速开发智能算法，并能够将算法转化为实际应用和项目。本书的定位就是为想在人工智能应用领域学习和工作的人士提供助力，本书适合智能算法初学者入门学

习，同时也适合作为计算机及相关专业学生的教材和上机指导书。

为了让广大读者更好地理解和使用书中的示例代码，作者为大家提供了一个完全公开的 GitHub 代码库来维护本书的示例程序。该代码库的网址为 https://github.com/luckh2/tensorflow-algo。衷心地希望各位读者能够从本书中获益，这也是对我最大的支持和鼓励。对于书中出现的任何错误或者不准确的地方，欢迎大家批评指正，并发送邮件至 luckh2@163.com。最后我想感谢家人的陪伴和支持，还要感谢所有为本书付出心血的电子工业出版社的编辑们，感谢电子工业出版社张迪老师的鼓励和帮助。感谢在写作过程中给予我大力支持的所有人，没有你们的支持也就没有这本书的诞生。

作　者

2019 年 6 月

目 录

入 门 篇

第 1 章 学习环境搭建 …… 3

1.1 Docker 工具箱 …… 3
1.2 运行 Docker 镜像 …… 6
1.3 Jupyter 笔记本 …… 10
1.3.1 Jupyter 界面 …… 10
1.3.2 Jupyter 单元格 …… 12
1.3.3 Jupyter 模式 …… 14
1.3.4 Jupyter 常用指令 …… 14
1.4 NumPy 库 …… 15
1.4.1 ndarray 数据基础 …… 16
1.4.2 ndarray 广播运算 …… 20
1.4.3 ndarray 函数运算 …… 22
1.4.4 ndarray 索引切分 …… 24
1.5 Pandas …… 25
1.5.1 Pandas 基础对象 …… 26
1.5.2 Pandas 选择数据 …… 29
1.5.3 Pandas 处理实例 …… 31
1.6 Scikit-Learn …… 34
1.6.1 sklearn.datasets …… 34
1.6.2 Pandas 处理 …… 35
1.6.3 sklearn 回归 …… 36

第 2 章 TensorFlow 入门 …… 38

2.1 Hello TensorFlow …… 39
2.2 TensorFlow 数据结构 …… 39
2.3 TensorFlow 计算-数据流图 …… 40
2.3.1 常量节点（Constant） …… 42

2.3.2 占位符节点（Placeholder） …… 42
2.3.3 变量节点（Variable） …… 43
2.3.4 操作节点（Operation） …… 45
2.4 TensorFlow 会话与基本操作 …… 45
2.5 TensorFlow 可视化 …… 47
第 3 章 TensorFlow 进阶 …… 49
3.1 TensorFlow 数据处理 …… 50
3.1.1 索引计算 …… 50
3.1.2 矩阵计算 …… 51
3.1.3 形状计算 …… 53
3.1.4 规约计算 …… 54
3.1.5 分割计算 …… 55
3.1.6 张量的形状 …… 57
3.1.7 张量的运算 …… 58
3.1.8 骰子游戏 …… 61
3.2 TensorFlow 共享变量 …… 62
3.2.1 name_scope 名字域 …… 62
3.2.2 variablescope 变量域 …… 63
3.3 TensorFlow 模型配置 …… 64

基 础 篇

第 4 章 线性回归算法 …… 69
4.1 BOSTON 数据集 …… 70
4.2 TensorFlow 模型 …… 72
4.2.1 准备数据 …… 72
4.2.2 定义模型 …… 72
4.2.3 训练模型 …… 73
4.2.4 评估模型 …… 73
4.2.5 可视化模型 …… 73
4.3 Estimator 模型 …… 75
4.3.1 Dataset API …… 75
4.3.2 估算器介绍 …… 76
4.3.3 准备数据 …… 77
4.3.4 定义模型 …… 78
4.3.5 训练模型 …… 78
4.3.6 评估模型 …… 78

4.3.7 可视化模型 …… 79
4.4 Keras 模型 …… 81
4.4.1 定义模型 …… 81
4.4.2 训练模型 …… 81
4.4.3 评估模型 …… 82
4.4.4 可视化模型 …… 82
第 5 章 逻辑回归算法 …… 84
5.1 线性回归到逻辑回归 …… 84
5.2 最小二乘到交叉熵 …… 86
5.3 MNIST 数据集 …… 88
5.4 TensorFlow 模型 …… 88
5.4.1 准备数据 …… 89
5.4.2 定义模型 …… 89
5.4.3 训练模型 …… 90
5.4.4 评估模型 …… 91
5.4.5 可视化模型 …… 91
5.5 Estimator 模型 …… 92
5.5.1 准备数据 …… 92
5.5.2 定义模型 …… 93
5.5.3 训练模型 …… 93
5.5.4 评估模型 …… 93
5.5.5 可视化模型 …… 94
5.6 Keras 模型 …… 95
5.6.1 准备数据 …… 95
5.6.2 定义模型 …… 96
5.6.3 训练模型 …… 96
5.6.4 评估模型 …… 96
5.6.5 可视化模型 …… 97
第 6 章 算法的正则化 …… 99
6.1 过拟合 …… 99
6.2 正则化 …… 99
6.3 编程实战 …… 103

进 阶 篇

第 7 章 神经网络与深度学习算法 …… 113
7.1 神经网络 …… 113

7.1.1 激活函数……114
7.1.2 编程实战……119
7.2 神经网络训练……123
7.2.1 训练困难分析……124
7.2.2 编程实战……124
7.3 多类别神经网络……133
7.3.1 逻辑回归与深度网络……133
7.3.2 权重可视化……135
7.4 神经网络嵌入……136
7.4.1 一维数轴排列……137
7.4.2 二维数轴排列……137
7.4.3 传统类别表示……138
7.4.4 嵌入表示……140
第 8 章 卷积神经网络（CNN）……141
8.1 卷积神经网络简介……141
8.2 CNN 与 DNN……142
8.3 卷积操作……142
8.4 卷积实战……145
8.5 池化操作……149
8.6 池化实战……149
8.7 Relu 非线性激活……150
8.8 TensorFlow 卷积神经网络实战……151
8.9 Estimalor 卷积神经网络实战……155
8.10 Keras 卷积神经网络实战……159
第 9 章 循环神经网络（RNN）……162
9.1 循环神经网络简介……162
9.2 DNN、CNN 与 RNN……162
9.3 手工循环神经网络……164
9.4 static_rnn 循环神经网络……165
9.5 dynamic_rnn 循环神经网络……167
9.6 TensorFlow 循环神经网络实战……169
9.7 Estimator 循环神经网络实战……173
9.8 Keras 循环神经网络实战……176
9.9 LSTM 模型……178
9.10 GRU 模型……180

第 10 章　自动编码器（AutoEncoder）……182

10.1　自动编码器简介……182
10.2　自动编码器与 PCA……183
10.3　稀疏自动编码器……185
10.4　栈式自动编码器（SAE）……187
10.4.1　关联权重……190
10.4.2　分阶段训练……192
10.4.3　无监督预训练……194
10.5　降噪自动编码器（DAE）……198
10.6　变分自动编码器（VAE）……200
10.6.1　变分自动编码器原理……200
10.6.2　变分自动编码器生成数字……203

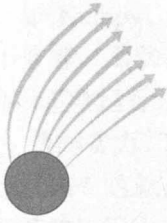
应　用　篇

第 11 章　生成式对抗网络……207

11.1　生成式对抗网络简介……207
11.2　GAN 工作原理……207
11.3　GAN 改进模型……209
11.4　GAN 模型实战……212
11.5　GAN 训练技巧……221
11.6　GAN 未来展望……222

第 12 章　使用 TensorFlow Hub 进行迁移学习……223

12.1　图像迁移学习……223
12.2　文本迁移学习……224
12.3　完整的文本分类器……225
12.4　迁移学习分析……228



入门篇

第 1 章
学习环境搭建

学习 TensorFlow，首先要配置好 TensorFlow 的运行环境，这也是很多入门者感到困难的地方。要保证 TensorFlow 能够正常运行，必须确保软件版本、各种依赖库和组件的安装都正确。举例来说，安装 Python 时，有 2.x 和 3.x 不同的版本，有面向 32 位和 64 位不同的操作系统环境，还必须有各种依赖库，如 Numpy、Matplotlib 等，往往还要配置环境变量。如果某些版本的模块与当前环境不兼容，那就会出现各种各样的错误。开发者常常碰到的一个场景是：在自己的机器上可以正常运行，换一台机器上就运行不了。环境配置如此麻烦，换一台机器，就要重来一次，费时费力。能不能从根本上解决问题？安装软件的时候，能不能把正确的环境打包复制过来？这就是 Docker 起作用的地方。

Docker（搬运工）是一个遵从 Apache 协议开源的应用容器引擎，它把要运行的应用，以及依赖软件打包到一个轻量级、可移植的自给自足的容器中，然后可以发布到任何流行的 Linux、Windows 和 MacOS 机器上。Docker 的推出具有划时代的意义，它彻底释放了计算虚拟化的威力，极大地提高了应用的部署效率，降低了云计算资源的供应成本。使用 Docker，我们可以把配置好的 TensorFlow 资源下载到自己的机器上直接运行，避免了烦琐的安装和配置，极大地提高了效率，节约了时间。Docker 有 3 个基本要素，即 Docker Containers（容器），负责应用程序的运行，包括操作系统、用户添加的文件和元数据；Docker Images（镜像），是一个只读模板，用来运行 Docker 容器；DockerFile，文件指令集，用来说明如何自动创建 Docker 镜像。镜像文件是 Docker 应用的基础，在 Docker Hub 上存储了大量的公共镜像，其中就包括了我们要使用的 TensorFlow 镜像。总的来说，Docker 通过容器提供服务，仅使用较小的操作系统代价。可以在任何物理和虚拟机甚至云端部署 Docker 容器。由于 Docker 容器非常轻便，因此它们非常易于扩展。Docker 让开发人员轻松开发应用程序，将其发送到容器中，然后可以在任何地方部署。

1.1 Docker 工具箱

我们以 Windows 系统为例，介绍安装 Docker 相关工具的过程，MacOS 系统的安装可以参照 https://docs.docker.com/toolbox/toolbox_install_mac/网址中的内容；Linux 系统（包括 Ubuntu、Debian、CentOS 和 Fedora）可以安装 Dokcer CE（Community Edition），参见 https://docs.docker.com/install/linux/docker-ce/ubuntu/网址中的内容。

目前支持 Windows 系统的 Docker 工具包括 Docker Toolbox 和 Docker CE。经过测试，我们推荐使用 Docker Toolbox，Docker Toolbox 支持 64 位的 Windows 7、8、10 等系统。首先检查系统是否打开虚拟化支持，用户可以右键单击任务栏，选择“任务管理器”，切换到“性能”页进行查看，如图 1-1 所示。

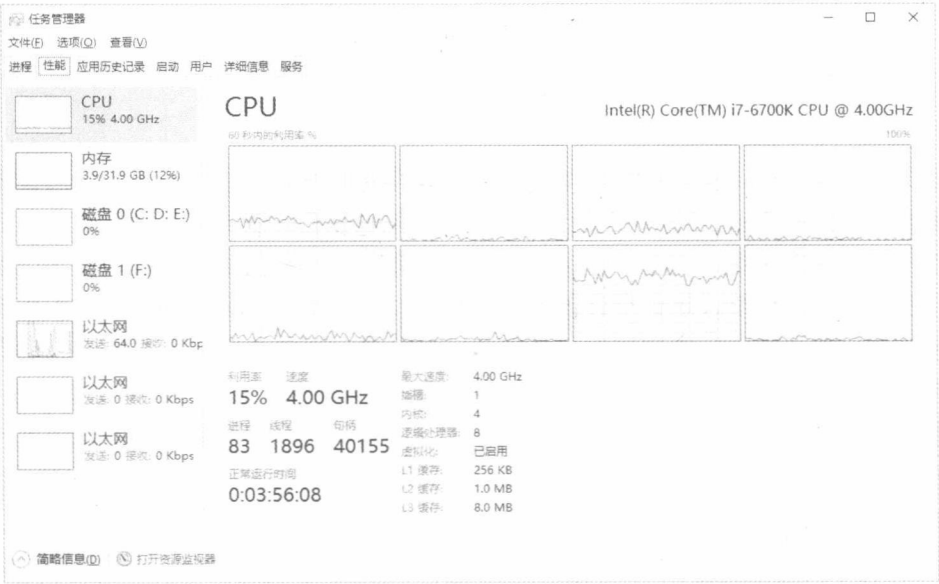


图 1-1 “性能”页

如果虚拟化功能未启用，需要重新启动计算机进入 BIOS 将虚拟化功能打开。通过网址 https://docs.docker.com/toolbox/toolbox_install_windows/进入相应页面，在页面中单击相应下载按钮进行下载。因为下载文件较大，推荐使用下载工具进行下载。下载后运行 DockerToolbox.exe 进行安装，如图 1-2 所示。

图 1-2 Docker Toolbox 安装界面

安装过程中接受所有默认选项即可。安装完成后，双击“Docker Quickstart Terminal”，启动 Docker 客户端，如图 1-3 所示。

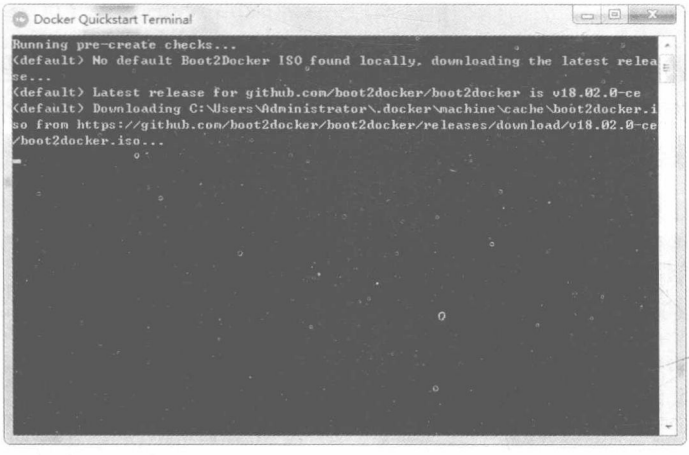


图 1-3　首次运行 Docker 客户端

在第一次运行 Docker 时，会在线下载最新版本的 Boot2Docker.iso 文件，文件会被下载到 C:.docker 目标文件夹中，如果当前用户不是 Administrator，就用当前用户名替换路径中的 Administrator，如用户名是 zhangsan，文件夹就是 C:.docker。下载耗时较长，有时会超时失败，建议用下载工具下载相应文件，下载后把它复制到目标文件夹中即可。

以下列出 Docker 最常用的一些命令。

（1）docker version：查看 Docker 的版本信息。

（2）docker info：列出全局信息。

（3）docker run hello-world：启动容器，运行最简示例镜像，测试 Docker 能否正常运行。

（4）docker run -it image_name:tag_name /bin/bash：启动容器，-t 分配一个伪终端并绑定到容器的标准输入上；-i 则让容器的标准输入保持打开；-d 则进入后台运行。

（5）docker pull image_name：从仓库获取所需要的镜像。

（6）docker images：列出本地镜像。

（7）docker ps –a：列出当前容器。

（8）docker start container_id：启动容器。

（9）docker stop container_id：停止容器。

（10）docker inspect container_id：查看容器详情。

（11）docker attach container_id：进入运行中的容器。

（12）docker rm：删除容器。

（13）docker rmi：删除镜像。

（14）docker save -o tensorflow.tar tensorflow/tensorflow:latest-py3：导出镜像为打包文件。

（15）docker load -i tensorflow.tar：从打包文件导入镜像。

1.2 运行 Docker 镜像

运行 Docker 后，初始页面会显示一条载货小鲸鱼的文本图像，这是 Docker 的标志。同时还会显示出为当前机器配置的 IP 地址，一般是 192.168.99.100。后面在运行 TensorFlow 镜像时需要使用这个地址，如图 1-4 所示。

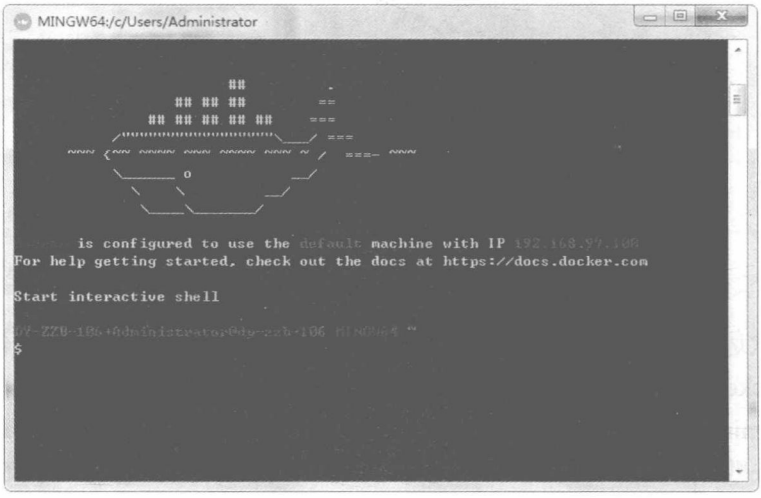


图 1-4 Docker 客户端初始页面

在 Docker 中运行各种应用都需要使用对应的镜像文件，在 hub.docker.com 网站上可以找到数以万计的镜像文件，我们要使用的 TensorFlow 镜像也可以从该网站下载。官方的 TensorFlow 镜像位于 Docker Hub 中的 tensorflow/tensorflow 中。镜像版本使用表 1-1 中的标签标记。

表 1-1 TensorFlow 镜像的标签

标签	描述
latest	最新发布的 TensorFlow CPU 二进制映像（默认）
nightly	每晚构建 TensorFlow 镜像（最新，不保证稳定）
version	指定 TensorFlow 二进制映像的版本，如 1.5
标签变体	
tag-devel	指定的标签版本和源代码
tag-gpu	指定的标签版本支持 GPU
tag-py3	指定的标签版本支持 Python 3
tag-gpu-py3	指定的标签版本支持 GPU 和 Python 3
tag-devel-py3	指定的标签版本支持 Python 3 和源代码
tag-devel-gpu	指定的标签版本支持 GPU 和源代码
tag-devel-gpu-py3	指定的标签版本支持 GPU 和 Python 3，以及源代码

我们可以使用 pull 命令从 Docker Hub 网站下载对应 Python 3 版本的最新 TensorFlow 镜像：

```
docker pull tensorflow/tensorflow:latest-py3
```

这条命令将下载 Python 3 版本的最新 TensorFlow（当下是 1.5 版）镜像，tensorflow/tensorflow 是镜像的名字、斜线前的 tensorflow 表示 Docker Hub 用户名，斜线后的 tensorflow 表示资源库名称，冒号后面的 latest-py3 是 Docker 镜像的标签（Tag），用以区分镜像的不同的子版本。以 TensorFlow 为例，其镜像支持数十种标签，包括 latest、nightly、latest-gpu、latest-devel-gpu 等。其中的 latest 是默认标签，也就是说如果未指定标签，就会使用 latest。我们这里没有使用 latest 标签，因为 tensorflow/tensorflow:latest 镜像对应 Python 2.7 版本，我们要使用的是 Python 3.5 版，对应的是 latest-py3 标签。

pull 命令成功运行后，TensorFlow 的镜像文件就下载到本地了，我们可以通过 images 命令来查看本地的所有镜像：

```
docker images
```

在运行 TensorFlow 镜像之前，我们需要准备好要测试的代码资源。首先在当前用户目录下新建一个子目录 git，如果用户名是 zhangsan，文件夹就是 C:，将本书配套的代码或者其他练习代码复制到该文件夹中。运行镜像时，运行环境和本地环境是隔离的，为了让运行起来的镜像（容器）能够访问本地资源，需要进行资源映射，如 TensorFlow 镜像中封装了 Jupyter 工具，默认使用 8888 端口提供服务，就要使用-p 命令将本地的 8888 端口映射到容器的 8888 端口。为了让容器能够访问本地 git 目录中的文件，我们使用-v 命令将本地的 git 目录映射到容器里相应的目录。TensorFlow 镜像中的默认路径是/notebooks，可以把本地的 git 目录映射到/notebooks 下面。要运行 TesnorFlow，可以使用 run 命令，同时指定端口映射和路径映射：

```
docker run -it -p 8888:8888 -v ~/git/:/notebooks/git/tensorflow/tensorflow:
latest-py3
```

在这条 run 命令里，-it 表示要启动一个交互式（Interactive）的终端（Terminal）；-p 表示端口（Port）映射；-v 表示路径（Volume）映射。映射的格式是“本地资源：容器资源”，如~/git/:/notebooks/git/表示把本地当前用户目录（~）下的 git 子目录映射到容器里根目录（/）下的 notebooks/git 子目录。为什么容器目录里要加上 notebooks 目录呢？如果你打开 TensorFlow 容器看一下就明白了，在我们运行的 TensorFlow 容器里，事先已经建立了/notebooks 目录，Jupyter 就是从这个目录启动的，所以/notebooks 目录就是 Jupyter 的根目录，我们把本地的 git 目录映射到/notebooks/git，所以打开 Jupyter 页面后就可以看到 git 目录，单击进入就可以测试代码了。

执行完 docker run 命令后，系统会提示我们打开 Jupyter 的服务页面，如图 1-5 所示。