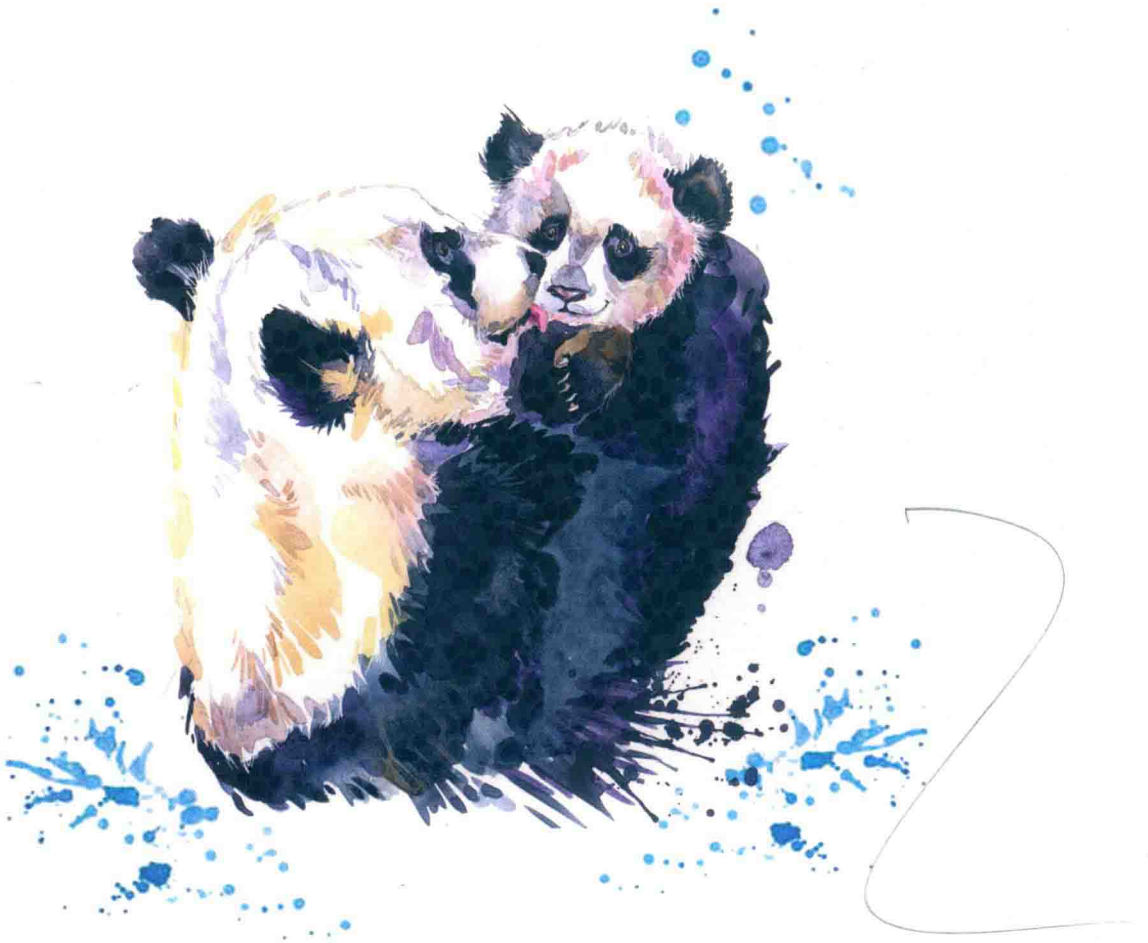
CMP BOOKS
机工IT

- 88个典型案例+4个综合案例带您迅速掌握深度学习
- 附赠全部案例程序的源代码



TensorFlow

深度学习从入门到进阶

张德丰 编著

机械工业出版社
CHINA MACHINE PRESS

TensorFlow 深度学习从入门到进阶

张德丰　编著

机械工业出版社

本书以 TensorFlow 为主线进行讲解，书中每章节都以理论引出，以 TensorFlow 应用巩固结束，理论与实践相结合，让读者快速掌握 TensorFlow 机器学习。本书共 11 章，主要包括 TensorFlow 与深度网络、TensorFlow 编程基础、TensorFlow 编程进阶、线性回归、逻辑回归、聚类分析、神经网络算法、卷积神经网络、循环神经网络、其他网络、机器学习综合实战等内容。

本书适合 TensorFlow 初学者阅读，也适合研究 TensorFlow 的广大科研人员、学者、工程技术人员学习参考。

图书在版编目（CIP）数据

TensorFlow 深度学习从入门到进阶 / 张德丰编著. —北京：机械工业出版社，2020.4

ISBN 978-7-111-65263-2

Ⅰ. ①T… Ⅱ. ①张… Ⅲ. ①人工智能—算法 Ⅳ. ①TP18

中国版本图书馆 CIP 数据核字（2020）第 057033 号

机械工业出版社（北京市百万庄大街 22 号 邮政编码 100037）

策划编辑：张淑谦 责任编辑：张淑谦

责任校对：张艳霞 责任印制：郜 敏

北京中兴印刷有限公司印刷

2020 年 5 月 • 第 1 版第 1 次印刷

184mm×260mm • 24 印张 • 596 千字

0001—2000 册

标准书号：ISBN 978-7-111-65263-2

定价：109.00 元

电话服务

客服电话：010-88361066

010-88379833

010-68326294

网络服务

机 工 官 网：www.cmpbook.com

机 工 官 博：weibo.com/cmp1952

金 书 网：www.golden-book.com

机工教育服务网：www.cmpedu.com

前　言

近年来，机器学习已经从科学和理论专家的技术资产转变为 IT 领域大多数大型企业日常运营中的常见主题。深度学习等相关技术开始用于应对数据量的爆炸问题，使得访问前所未有的大量信息成为可能。此外，硬件领域的限制促使研发人员开发大量的并行设备，这让用于训练同一模型的数据能够成倍增长。

硬件和数据可用性方面的进步使研究人员能够重新审视先驱者在基于视觉的神经网络架构（卷积神经网络等）方面开展的工作，将它们用于许多新的问题。这都归功于具备普遍可用性的数据以及现在计算机拥有的强悍的计算能力。

为了解决这些新的问题，机器学习的从业者创建了许多优秀的机器学习包，它们每个都拥有一个特定的目标来定义、训练和执行机器学习模型。2015 年 11 月 9 日，谷歌公司进入了机器学习领域，决定开源自己的机器学习框架 TensorFlow，谷歌内部许多项目都以此为基础。

本书将机器学习背后的基本理论与应用实践联系起来，通过这种方式让读者聚焦于如何正确地提出问题、解决问题。书中讲解了如何使用 TensorFlow 强大的机器学习库和一系列统计模型来解决一系列的机器问题。不管你是数据科学领域的初学者，还是想进一步拓展对数据科学领域认知的进阶者，本书都是一个重要且不可错过的选择，它能帮助你了解如何使用 TensorFlow 解决数据爆炸问题。

之所以学习利用 TensorFlow 解决机器问题，是因为 TensorFlow 完全绑定兼容 Python，即 Python 具有的特点，TensorFlow 也具备，所以利用 TensorFlow 对大数据进行提取、分析、降维完全没有压力。

本书共分四大部分：

第一部分，介绍 TensorFlow 及深度学习的基础知识（第 1～3 章）。

这部分主要介绍深度学习的定义、优势、应用、TensorFlow 的特点、环境搭建、张量、图、会话、变量、队列与线程等内容。通过本部分内容的学习，读者将对深度学习、TensorFlow 的特点、功能及其编程基础、进阶编程有全面的认识，可轻松掌握 TensorFlow 并认识深度学习。

第二部分，介绍 TensorFlow 在线性回归、逻辑回归、聚类分析等方面的机器应用（第 4～6 章）。

本部分内容主要介绍如何利用 TensorFlow 软件解决实现线性回归问题、戴明回归算法、岭回归与 Lasso 回归算法、弹性网络回归算法、逻辑函数的逆函数、Softmax 回归、支持向量机、K-均值聚类法等问题。本部分每章节都采用理论、公式、应用实例相结合的方式，让读者领略利用 TenosrFlow 解决机器问题的方便、快捷。

第三部分，介绍神经网络等相关问题（第 7～10 章）。

本部分内容主要包括反向网络、激励函数、卷积网络、循环网络、自编码网络、对抗网络等内容，通过这部分内容的学习，读者将学会从各个方面利用 TensorFlow 软件深入透彻解决神经网络等机器问题，进一步领略 TensorFlow 的强大功能，感受到 TensorFlow 可以成为现今流行软件的原因。

第四部分，介绍机器学习的综合实例（第 11 章）。

本部分主要是在前面介绍的机器学习相关知识的基础上，综合应用机器学习知识求解实际问题。其中有几个实例都用同一组数据集，利用不同的方法进行求解，比较各种方法的求解结果，通过对比学习，让读者更直观地感受各方法的优缺点，以使读者在以后的应用中根据需要选择合适的方法。

本书适合 TensorFlow 初学者阅读，也适合研究 TensorFlow 的广大科研人员、学者、工程技术人员学习参考。

由于作者水平有限，错误和疏漏之处在所难免。在此，诚恳地期望得到各领域专家和广大读者的批评指正。

张德丰

目　录

第1章 TensorFlow与深度学习

TensorFlow 是一个采用数据流图（Data Flow Graphs）进行数值计算的开源软件库。TensorFlow 最初由谷歌大脑小组（隶属于谷歌机器智能研究机构）的研究员和工程师开发，是基于 DistBelief 研发的第二代人工智能学习系统，用于机器学习和深度神经网络方面的研究，但这个系统的通用性使其也可广泛用于其他计算领域。2015 年 11 月 9 日，谷歌发布人工智能系统 TensorFlow 并宣布开源。

其命名来源于本身的原理，Tensor（张量）意味着 N 维数组，Flow（流）意味着基于数据流图的计算。TensorFlow 运行过程就是张量从图的一端流动到另一端的计算过程。张量从图中流过的直观图像是其取名为“TensorFlow”的原因。

1.1 深度学习的由来

在 2011 年，谷歌开展了大规模的面向科学和产品开发的深度学习应用研究，其中就包括 TensorFlow 的前身 DistBelief。DistBelief 主要用于构建各尺度下的神经网络分布式学习和交互系统，又被称为“第一代机器学习系统”，其在 Alphabet 旗下其他公司的产品开发中得到改进和广泛应用。2015 年，在 DistBelief 的基础上，谷歌完成了对“第二代机器学习系统”TensorFlow 的开发并对代码进行了开源。相比于 DistBelief，TensorFlow 的性能得到了改进，构架更灵活、可移植性更强。此后，TensorFlow 得到了快速发展。

1.2 语言与系统的支持

随着 TensorFlow 技术的发展与完善，其不仅支持多种客户端语言下的安装和运行，还可以绑定并支持版本兼容运行的 C 和 Python 语言。

1. Python

在 Python 语言框架下，TensorFlow 有 3 个不同的版本，分别为 CPU 版本（TensorFlow）、包含 GPU 加速的版本（TensorFlow-gpu）、每日编译版本（tf-nightly、tf-nightly-gpu）。其中，安装 Python 版本的 TensorFlow 可以使用模块管理工具 pip/pip3 或 anaconda 在终端直接运行。

此外，Python 版 TensorFlow 也可以使用 Docker 安装。

2. C

TensorFlow 中兼容了 C 语言下的 API，可用于构建其他语言的 API。支持 macOS 10.12.6

Sierra 或更高版本，macOS 版本不包含 GPU 加速。其安装过程如下：

- 下载 TensorFlow 预编译的 C 语言文件到本地系统路径下并解压缩。
- 使用 ldconfig 编译链接。
- 用户还可在其他路径解压文件并手动编译链接。
- 编译 C 接口时要确保本地的 C 编译器能够访问 TensorFlow 库。

3. 配置 GPU

TensorFlow 支持在 Linux 和 Windows 系统下使用统一计算架构。配置 GPU 时要求系统有相对应的支持版本。

在 Linux 下配置 GPU 时，将 CUDA Toolkit 和 CUPTI 的路径加入$LD_LIBRARY_PATH 的环境变量中即可。对于 CUDA 为 3.0 或其他版本的 NVIDIA 程序，需要从源文件中来编译 TensorFlow。对 Windows 下的 GPU 配置，需要将 CUDA、CUPTI 和 cuDNN 的安装路径添加到%PATH%的环境变量中。

Linux 系统下使用 docker 安装的 Python 版 TensorFlow 也可配置 GPU 加速且无需 CUDA Toolkit。

1.3 TensorFlow 的特点

TensorFlow 能在这么短的时间内得到如此广泛的应用，主要得益于它自身的特点。

首先，TensorFlow 作为一个支持深度学习的计算框架，能够支持 Linux、Windows 等各种移动平台。

其次，TensorFlow 本身提供了非常丰富的机器学习相关的 API，是目前所有机器学习框架中提供 API 最齐全的。在 TensorFlow 中可实现基本的向量矩阵计算、各种优化算法、卷积神经网络、循环神经网络等，TensorFlow 还提供了可视化的辅助工具和及时更新的最新算法库。

更重要的是，谷歌大力支持 TensorFlow，同时开源世界众多贡献者为其添砖加瓦，使其得到飞速发展。

TensorFlow 的特点主要表现在：

- 灵活性高

在 TensorFlow 中，只要把一个计算过程表示成数据流图即可实现计算。可以用计算图建立计算网络进行相关操作。用户可以基于 TensorFlow 编写自己的库，还可以编写底层的 C++代码，并能自定义地将编写的功能添加到 TensorFlow 中。

- 可移植性强

TensorFlow 有很强的可移植性，可以在台式计算机中的一个或多个 CPU（或 GPU）、服务器、移动设备等上运行。

- 自动求微分

TensorFlow 有强大的自动求微分能力。TensorFlow 用户只需要定义预测模型的结构，将结构和目标函数结合在一起、添加数据，即可利用 TensorFlow 自动计算相关的微分导数。

● 支持多语言

TensorFlow 支持多种语言，如 Python、C++、Java、Go 语言，它可以直接采用 Python 来构建和执行计算图，还可以采用交互式的 IPython 语言执行 TensorFlow 程序。

● 开源项目丰富

TensorFlow 在 GitHub 上的主项目下包含了许多应用领域的最新研究算法的代码实现，如自然语言某些处理领域达到人类专家水平的 syntaxnet 项目等。TensorFlow 的使用者可以方便地借鉴这些已有的高质量项目快速构建自己的机器学习应用。

● 算法库丰富

TensorFlow 的算法库是开源机器学习框架中最齐全的，而且还可以不断地添加新的算法库。这些算法库基本上满足了使用者大部分的需要。

● 性能最优化

由于给予了线程、队列、异步操作等最佳的支持，TensorFlow 可以将用户手边硬件的计算潜能全部发挥出来。用户可以自由地将 TensorFlow 图中的计算元素分配到不同设备上，TensorFlow 可以帮我们管理好这些不同副本。

● 科研和产品相结合

谷歌的科学家用 TensorFlow 尝试新的算法，产品团队则用 TensorFlow 来训练和使用计算模型，并直接提供给在线用户。TensorFlow 可以让应用型研究者将想法迅速运用到产品中，还可以让学术型研究者更直接地彼此分享代码，从而提高科研产出率。

1.4 核心组件

TensorFlow 的核心组件（Core Runtime）如图 1-1 所示，主要包括分发中心（Distributed Master）、执行器（Dataflow Executor）、内核应用（Kernel Implementation）以及最底端的网络层（Networking Layer）和设备层（Device Layer）。

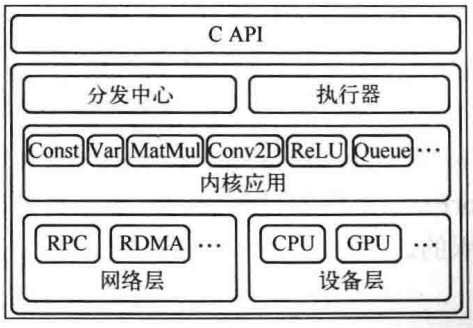


图 1-1 核心组件结构图

分发中心从输入的数据流图中剪取子图，将其划分为操作片段并启动执行器。分发中心处理数据流图时会进行预设定的操作优化，包括公共子表达式消去、常量折叠等。

执行器负责图操作在进程和设备中的运行、收发其他执行器的结果。执行器在调度本地设备时会选择进行并行计算和 GPU 加速。

内核应用负责单一的图操作，包括数学计算、数组操作、控制流和状态管理操作。内核应用使用 Eigen 执行张量的并行计算；cuDNN 库等执行 GPU 加速，此外用户可以在内核应用中注册额外的内核以提升基础操作。

1.5 TensorFlow 的主要依赖包

本节将介绍 TensorFlow 依赖的两个最主要的工具包——Protocol Buffer 和 Bazel。虽然 TensorFlow 依赖的工具包不仅限于这两个，但 Protocol Buffer 和 Bazel 是相对比较重要的，用户在使用 TensorFlow 的过程中很有可能会接触到。

1.5.1 Protocol Buffer 包

Protocol Buffer 包是谷歌公司开发的一种数据描述语言，可用于数据存储、通信协议等方面。它不依赖于语言和平台且可扩展性很强。Protocol Buffer 的结构化数据是什么呢？假设要记录一些用户信息，包括每个用户的名字、ID 和 E-mail 地址等信息，那么其结构形式为：

```
name:李明
id:112233
email:liming@126.com
```

上面的用户信息就是一个结构化数据。此处介绍的结构化数据指的是拥有多种属性的数据，如上述用户信息中包含名字、ID 和 E-mail 地址 3 种不同属性，那么它就是一个结构化数据。要将这些结构化的用户信息持久化或者进行网络传输，就需要先将它们序列化。何为序列化？就是将结构化的数据变成数据流的格式（变为一个字符串）。将结构化的数据序列化，并从序列化之后的数据流中还原出原来的结构化数据，统称为处理结构化数据，这就是 Protocol Buffer 包解决的主要问题。

除 Protocol Buffer 之外，XML 和 JSON 是两种比较常用的结构化数据处理工具。

Protocol Buffer 格式的数据和 XML 或 JSON 格式的数据有比较大的区别：首先，Protocol Buffer 序列化之后得到的数据是二进制流；其次，XML 或 JSON 格式的数据信息都包含在了序列化之后的数据中，不需要任何其他信息就能还原序列化之后的数据。

Protocol Buffer 是 TensorFlow 系统中使用到的重要工具，TensorFlow 中的数据基本都是通过 Protocol Buffer 来组织的。

1.5.2 Bazel 包

Bazel 是从谷歌开源的自动化构成工具，谷歌内部大部分的应用都是通过它来编译的。Bazel 在速度、可伸缩性、灵活性以及对不同程序语言和平台的支持上都要更加出色。本节将简单介绍 Bazel 是怎样工作的。

workspace（工作空间）是 Bazel 的一个基本概念。一个 workspace 可以简单地理解为一个文件夹，在这个文件夹中包含了编译一个软件所需要的源代码以及输出编译结果的链

接地址。一个 workspace 所对应的文件夹是这个项目的根目录，在这个根目录中需要有一个 workspace 文件，此文件定义了对外部资源的依赖关系。

在一个 workspace 内，Bazel 通过 BUILD 文件来找到需要编译的目标。BUILD 文件采用一种类似于 Python 的语法来指定每一个编译目标的输入、输出以及编译方式。Bazel 的编译方式是事先定义好的。Bazel 对 Python 支持的编译方式有 3 种：py_binary、py_library 和 by_test。其中 py_binary 将 Python 程序编译为可执行文件，py_test 将 Python 编译为测试程序，py_library 将 Python 程序编译为库函数，供其他 py_binary 或 py_test 调用。

下面给出一个简单的样例来说明 Bazel 是如何工作的。如下面的代码所示，在样例项目空间中有 4 个文件：workspace、BUILD、hello_world.py 和 hello.lib.py。

```
-rw-rw-rr-root root 208 BUILD
-rw-rw-rr-root root 48 hello_lib.py
-rw-rw-rr-root root 47 hello_world.py
-rw-rw-rr-root root 0 workspace
```

workspace 给出此项目的外部依赖关系。为了简单起见，这里使用一个空文件，表明这个项目没有对外部的依赖。hello_lib.py 完成打印“Hello World”的简单功能，它的代码为：

```
def print_hello_world():
  print("Hello World")
```

hello_world.py 通过调用 hello_lib.py 中定义的函数来完成输出，它的代码为：

```
import hello_lib
hello_lib.print_hello_world()
```

在 BUILD 文件中定义的两个编译目标：

```
py_library{
    name="hello_lib",
    srcs=[
      "hello_lib.py",
     ]
}

py_binary{
    name="hello_world ",
    srcs=[
       "hello_world.py",
    ],
    deps=[
      ":hello_lib",
    ],
}
```

从这个样例中可看出，BUILD 文件是由一系列编译目标组成的。定义编译目标的先后

顺序不会影响编译的结果。在每一个编译目标的第一行要指定编译方式，在这个样例中就是 py_library 或者 py_binary。在样例中 hello_world.py 需要调用 hello_lib.py 中的函数，所以 hello_world 的编译目标中将 hello_lib 作为依赖关系。

1.6 搭建环境

本节主要介绍在 Windows 的平台上如何安装 TensorFlow，以及简单的运行测试。

1.6.1 安装环境

因为深度学习计算过程中大量的操作是向量和矩阵的计算，而 GPU 在向量和矩阵计算方面比 CPU 有一个数量级的速度提升，所以机器学习在 GPU 上运算效率更高。

通过以下方式来查看 Windows 系统上的 GPU 信息。

在“运行”对话框中输入 dxdiag，如图 1-2 所示，然后单击“确定”按钮，此时会打开“DirectX 诊断工具”对话框。单击其中的“显示”选项卡，可以查看机器的显卡信息，如图 1-3 所示。

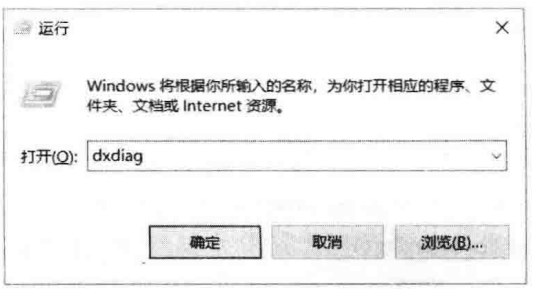


图 1-2 输入 dxdiag 命令

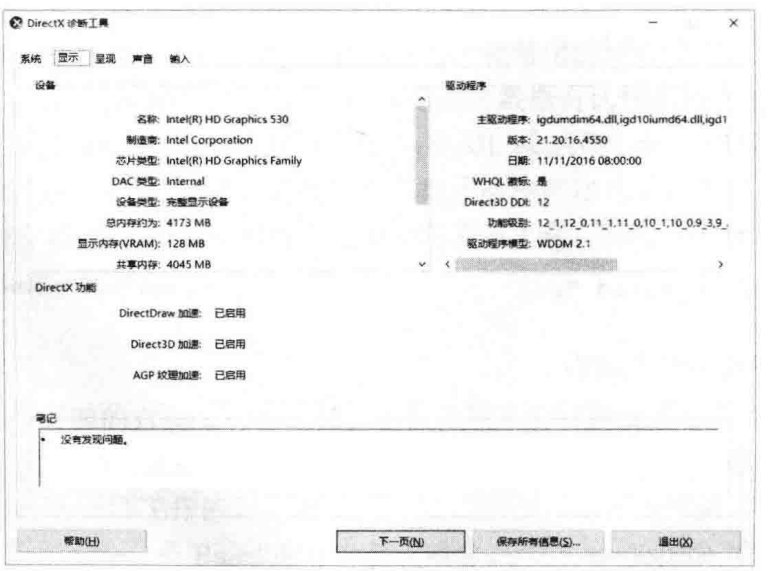


图 1-3 查看 Windows 的显卡信息

由图 1-3 可以看到，这个机器上的显卡芯片类型是 Intel(R) HD Graphics Family。

1.6.2 安装 TensorFlow

TensorFlow 的 Python 语言 API 支持 Python 2.7 和 Python 3.3 以上的版本。本书使用的是 TensorFlow 3.6.5 版本。

1. 安装 pip

pip 是用来安装和管理 Python 包的管理工具。首先，去 Python 官网下载 pip 最新版本（https://pypi.python.org/pypi/pip#downloads），下载完成后，在 Windows 系统上安装 pip 的命令为：

```
python setup.py install
```

接着在 Windows 中设置环境变量，方法为在 Windows 环境变量的 PATH 变量后添加“\Python 安装目录\Scripts”。

目前，TensorFlow 在 Windows 上只支持 64 位的 Python 3.6.5 版本。

2. 通过 pip 安装 TensorFlow

TensorFlow 已经把最新版本的安装程序上传到了 Pypi，所以可以通过最简单的方式来安装 TensorFlow（要求 pip 版本在 8.1 版本或者更高）。

安装 CPU 版本的 TensorFlow 的命令如下：

```
#Python3.6.5
sudo pip3 install tensorflow
```

安装支持 GPU 版本的 TensorFlow 的命令如下：

```
#Python3.6.5
sudo pip3 install tensorflow-gpu
```

在 Windows 系统上安装 CPU 版本（0.12 版本）的命令如下：

```
C:\> pip install --upgrade
   https://storage.googleapis.com/tensorflow/windows/cpu/tensorflow-
0.12.0-cp35
```

TensorFlow 在 Windows 上依赖 MSVCP140.DLL，这里需要提前安装 Visual C++ 2015 redistributable（x64 位），其下载地址为 https://www.microsoft.com/en-us/download/details.aspx?id=53587，下载文件为 vc_redist.x64.exe。

1.6.3 安装测试

如果顺利的话，到这里已经成功安装了 TensorFlow，那么简单测试一下安装是否成功。

```
>>> import tensorflow as tf
>>>print(tf.__version__)
1.7.0
```

上面这段代码若正常运行，会打印出 TensorFlow 的版本号，这里是“1.7.0”。

但也可能会存在一些问题：

如果在 import tensorflow as tf 之后，打印出来 Cuda 的 so 或者 CuDNN 的 so 没有找到，一般是因为 Cuda 或者 CuDNN 的路径没有添加到环境变量里。

下面再进行一个简单的计算，看看 TensorFlow 是否运行正常。输入如下代码：

```
>>> import tensorflow as tf
>>>hello = tf.constant('Hello, TensorFlow!')
>>>sess = tf.Session()
2020-03-9 12:18:51.120893: I T:\src\github\tensorflow\tensorflow\core\
platform\cpu_feature_guard.cc:140] Your CPU supports instructions that this
TensorFlow binary was not compiled to use: AVX2
>>>print(sess.run(hello))
b'Hello, TensorFlow!'
>>> a = tf.constant(2)
>>> b = tf.constant(3)
>>> c=sess.run(a + b)
>>>print("2+3= %d"%c)
2+3=5
```

如果这段代码可以正常输出“Hello, TensorFlow!”和“2+3= 5”，那么说明 TensorFlow 已经成功安装了。

1.7 Geany

Geany 是一个小巧的使用 GTK+2 开发的跨平台开源集成开发环境，以 GPL 许可证分发源代码，是免费的自由软件，当前版本为 1.31。它支持基本的语法高亮、代码自动完成、调用提示、插件扩展。支持文件类型包括 C、CPP、Java、Python、PHP、 HTML、DocBook、Perl、LateX 和 Bash 脚本。该软件小巧、启动迅速，主要缺点是界面简陋、运行速度慢、功能简单。

要下载 Windows Greany 安装程序，可访问 httt://geany.org/，单击 Download 下的 Releases，找到安装程序 geany-1.25_setup.exe 或类似的文件。下载安装程序后，运行它并接受所有的默认设置。

启动 Geany，选择“文件|另存为”，将当前的空文件保存为 hello_world.py，再在编辑窗口中输入代码：

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

效果如图 1-4 所示。

现在选择菜单“生成|设置生成”命令，将看到文字 Compile 和 Execute，它们旁边都有一个命令。默认情况下，这两个命令都是 Python（全部小写），但 Geany 不知道这个命令位于系统的什么地方。需要添加启动终端会话时使用的路径。在编译命令和执行中，添加命令 Python 所在的驱动器和文件夹。编译命令应类似于图 1-5 所示。