

Keras Deep Learning Cookbook

Keras深度学习实战

[印] 拉蒂普·杜瓦 (Rajdeep Dua) 著
曼普里特·辛格·古特 (Manpreet Singh Ghotra)
罗娜 祁佳康 译



机械工业出版社
China Machine Press

Keras Deep Learning Cookbook

Keras深度学习实战

拉蒂普·杜瓦 (Rajdeep Dua)

[印] 曼普里特·辛格·古特 (Manpreet Singh Ghotra) 著

罗娜 祁佳康 译



机械工业出版社
China Machine Press

图书在版编目 (CIP) 数据

Keras 深度学习实战 / (印) 拉蒂普·杜瓦 (Rajdeep Dua) 等著; 罗娜, 祁佳康译. —北京: 机械工业出版社, 2019.5

(智能系统与技术丛书)

书名原文: Keras Deep Learning Cookbook

ISBN 978-7-111-62627-5

I.K… II. ①拉… ②罗… ③祁… III. 软件设计 IV. TP311.5

中国版本图书馆 CIP 数据核字 (2019) 第 084161 号

本书版权登记号: 图字 01-2018-8351

Rajdeep Dua, Manpreet Singh Ghotra: Keras Deep Learning Cookbook (ISBN: 978-1-78862-175-5).

Copyright © 2018 Packt Publishing. First published in the English language under the title “Keras Deep Learning Cookbook”.

All rights reserved.

Chinese simplified language edition published by China Machine Press.

Copyright © 2019 by China Machine Press.

本书中文简体字版由 Packt Publishing 授权机械工业出版社独家出版。未经出版者书面许可, 不得以任何方式复制或抄袭本书内容。

Keras 深度学习实战

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: 刘 锋

责任校对: 殷 虹

印 刷: 北京文昌阁彩色印刷有限责任公司

版 次: 2019 年 6 月第 1 版第 1 次印刷

开 本: 186mm × 240mm 1/16

印 张: 12.5

书 号: ISBN 978-7-111-62627-5

定 价: 69.00 元

凡购本书, 如有缺页、倒页、脱页, 由本社发行部调换

客服热线: (010) 88379426 88361066

投稿热线: (010) 88379604

购书热线: (010) 68326294

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

THE TRANSLATOR'S WORDS

译者序

作为机器学习中的一个重要研究领域，人工神经网络的发展历史一波三折。2006 年以来，随着深度学习的兴起和成功应用，人工神经网络迎来了新的生机。与传统的人工神经网络相比，深度学习的最大特点是网络层数有了大幅度的增加，配合其他相关技术，解决了以图像、声音等作为数据的传统人工神经网络难以解决的问题。作为快速实现深度学习的平台，TensorFlow 一定程度上简化了神经网络的构建程序，Keras 进一步对 TensorFlow 进行了封装，从而能更加快速地把用户的思想转化为代码。

本书从实用的角度出发，全方面介绍了如何使用 Keras 解决深度学习中的各类问题。本书假设读者无任何关于深度学习编程的基础知识，首先介绍了 Keras 这一高度模块化、极简式的深度学习框架的安装、配置和编译等平台搭建知识，而后详细介绍了深度学习所要求的数据预处理等基本内容，在此基础上介绍了卷积神经网络、生成式对抗网络、递归神经网络这三种深度学习方法并给出了相关实例代码，最后本书介绍了自然语言处理、强化学习两方面的内容。

本书是一本实践性很强的深度学习工具书，既适合希望快速学习和使用 Keras 深度学习框架的工程师、学者和从业者，又特别适合立志从事深度学习和 AI 相关的行业并且希望用 Keras 开发实际项目的工程技术人员。

本书翻译工作得到国家自然科学基金项目（项目编号：61403140）的资助，在此表示衷心感谢。

感谢华章公司的刘锋编辑不辞辛苦地与译者沟通相关细节内容，同时感谢他在翻译本书过程中给予的诸多帮助。

限于本人水平，难免会对本书中部分内容的理解或中文语言表达存在不当之处，敬请读者批评指正，以便能够不断改进。

罗娜 祁佳康
2019 年于上海

ABOUT THE REVIEWER

审校者简介

Sujit Pal 工作于 Reed-Elsevier PLC Group 内的 Elsevier 实验室，研究涉及信息检索、分布式处理、本体开发、自然语言处理、机器学习，并用 Python、Scala 和 Java 进行开发。通过结合在这些领域的专长，帮助公司实现产品功能构建或改进。他相信终身学习，并长期在 sujitpal.blogspot.com 上记录他的一些见解。

前 言

Keras 采用 Python 编写，能够快速准确地训练卷积神经网络和递归神经网络，已经成为当下流行的深度学习库。

本书讲述了如何在 Keras 库的帮助下，高效地解决在训练深度学习模型时遇到的各种问题。内容包括如何安装和设置 Keras，如何在 TensorFlow、Apache MXNet 和 CNTK 后端开发中使用 Keras 实现深度学习。

从加载数据到拟合和评估模型获得最佳性能，你将逐步解决在深度学习建模时可能遇到的所有问题。在本书的帮助下，你将实现卷积神经网络、递归神经网络、对抗网络等。除此之外，你还将学习如何训练这些模型以实现真实的图像处理和语言处理任务。

本书的最后，你将完成一个实例以进一步了解如何利用 Python 和 Keras 的强大功能实现有效的深度学习。

本书读者对象

本书适合数据科学家或机器学习专家，可以帮助他们解决在训练深度学习模型时遇到的常见问题。阅读本书前，需要对 Python 有基本的了解，并了解机器学习和神经网络的内容。

本书涵盖的内容

第 1 章介绍了 Keras 的安装和设置过程以及如何配置 Keras。

第 2 章介绍了使用 CIFAR-10、CIFAR-100 或 MNIST 等数据集，以及用于图像分类的其他数据集和模型。

第3章介绍了使用 Keras 的各种预处理和优化技术，优化技术包括 TFOptimizer、AdaDelta 等。

第4章详细描述了不同的 Keras 层，包括递归层和卷积层等。

第5章通过宫颈癌分类和数字识别数据集的实例，详细解释如何使用卷积神经网络算法。

第6章包括基本的生成式对抗网络（GAN）和边界搜索 GAN。

第7章涵盖了递归神经网络的基础，以便实现基于历史数据集的 Keras。

第8章包括使用 Keras 进行单词分析和情感分析的 NLP 基础知识。

第9章展示了如何在 Amazon 评论数据集中使用 Keras 模型进行文本概述。

第10章侧重于使用 Keras 设计和开发强化学习模型。

阅读本书须知

读者应该掌握 Keras 和深度学习的基本知识。

排版约定

本书中包含许多排版约定。

文本中的代码元素、数据库表名、文件夹名、文件名、文件扩展名、路径名、用户输入和 Twitter 句柄都采用代码字体表示。举个例子：“最后，我们将所有评论保存到 pickle 文件中。”

代码段如下所示：

```
stories = list()
for i, text in enumerate(clean_texts):
    stories.append({'story': text, 'highlights': clean_summaries[i]})
```


当我们希望引起你对代码块的特定部分的注意时，相关的行或代码会以粗体显示：


```
from keras.datasets import cifar10
```

命令行的输入、输出表示如下：

```
sudo apt-get install graphviz
```

粗体：表示新术语、重要单词或词组。

 警告或重要说明。

 提示或小技巧。

示例代码及彩图下载

本书的示例代码及所有截图和样图，可以从 <http://www.packtpub.com> 通过个人账号下载，也可以访问华章图书官网 <http://www.hzbook.com>，通过注册并登录个人账号下载。

目 录

译者序	
审校者简介	
前言	
第 1 章 Keras 安装	1
1.1 引言	1
1.2 在 Ubuntu 16.04 上安装 Keras	1
1.2.1 准备工作	2
1.2.2 怎么做	2
1.3 在 Docker 镜像中使用 Jupyter Notebook 安装 Keras	7
1.3.1 准备工作	7
1.3.2 怎么做	7
1.4 在已激活 GPU 的 Ubuntu 16.04 上安装 Keras	9
1.4.1 准备工作	9
1.4.2 怎么做	10
第 2 章 Keras 数据集和模型	13
2.1 引言	13
2.2 CIFAR-10 数据集	13
2.3 CIFAR-100 数据集	15
2.4 MNIST 数据集	17
2.5 从 CSV 文件加载数据	18
2.6 Keras 模型入门	19
2.6.1 模型的剖析	19
2.6.2 模型类型	19
2.7 序贯模型	20
2.8 共享层模型	27
2.8.1 共享输入层简介	27
2.8.2 怎么做	27
2.9 Keras 函数 API	29
2.9.1 怎么做	29
2.9.2 示例的输出	31
2.10 Keras 函数 API——链接层	31
2.11 使用 Keras 函数 API 进行图像分类	32
第 3 章 数据预处理、优化和可视化	36
3.1 图像数据特征标准化	36
3.1.1 准备工作	36
3.1.2 怎么做	37
3.2 序列填充	39

3.2.1	准备工作	39	5.2.1	准备工作	74
3.2.2	怎么做	39	5.2.2	怎么做	74
3.3	模型可视化	41	5.3	数字识别	84
3.3.1	准备工作	41	5.3.1	准备工作	84
3.3.2	怎么做	41	5.3.2	怎么做	85
3.4	优化	43			
3.5	示例通用代码	43	第 6 章 生成式对抗网络		89
3.6	随机梯度下降优化法	44	6.1	引言	89
3.6.1	准备工作	44	6.2	基本的生成式对抗网络	90
3.6.2	怎么做	44	6.2.1	准备工作	91
3.7	Adam 优化算法	47	6.2.2	怎么做	91
3.7.1	准备工作	47	6.3	边界搜索生成式对抗网络	98
3.7.2	怎么做	47	6.3.1	准备工作	99
3.8	AdaDelta 优化算法	50	6.3.2	怎么做	100
3.8.1	准备工作	51	6.4	深度卷积生成式对抗网络	106
3.8.2	怎么做	51	6.4.1	准备工作	107
3.9	使用 RMSProp 进行优化	54	6.4.2	怎么做	108
3.9.1	准备工作	54			
3.9.2	怎么做	54	第 7 章 递归神经网络		116
			7.1	引言	116
第 4 章 使用不同的 Keras 层实现			7.2	用于时间序列数据的简单 RNN	117
分类		58	7.2.1	准备工作	118
4.1	引言	58	7.2.2	怎么做	119
4.2	乳腺癌分类	58	7.3	时间序列数据的 LSTM 网络	128
4.3	垃圾信息检测分类	66	7.3.1	LSTM 网络	128
			7.3.2	LSTM 记忆示例	129
第 5 章 卷积神经网络的实现		73	7.3.3	准备工作	129
5.1	引言	73	7.3.4	怎么做	129
5.2	宫颈癌分类	73			

7.4 使用 LSTM 进行时间序列预测	133	8.3.3 完整代码清单	162
7.4.1 准备工作	134	第 9 章 基于 Keras 模型的文本摘要	164
7.4.2 怎么做	135	9.1 引言	164
7.5 基于 LSTM 的等长输出序列到序列学习	143	9.2 评论的文本摘要	164
7.5.1 准备工作	143	9.2.1 怎么做	165
7.5.2 怎么做	144	9.2.2 参考资料	172
第 8 章 使用 Keras 模型进行自然语言处理	150	第 10 章 强化学习	173
8.1 引言	150	10.1 引言	173
8.2 词嵌入	150	10.2 使用 Keras 进行《CartPole》游戏	174
8.2.1 准备工作	151	10.3 使用竞争 DQN 算法进行《CartPole》游戏	181
8.2.2 怎么做	151	10.3.1 准备工作	183
8.3 情感分析	157	10.3.2 怎么做	187
8.3.1 准备工作	157		
8.3.2 怎么做	159		

Keras 安装

本章包括以下内容：

- ❑ 在 Ubuntu 16.04 上安装 Keras
- ❑ 在 Docker 镜像中使用 Jupyter Notebook 安装 Keras
- ❑ 在已激活 GPU 的 Ubuntu 16.04 上安装 Keras

1.1 引言

在本章中，我们将讨论如何在 Ubuntu 和 CentOS 上安装 Keras。本书使用的是 64 位的 Ubuntu 16.04（即 Canonical 公司于 2017 年 10 月 26 日发布的 Ubuntu 16.04 LTS 和 amd64 xenial 镜像）。

1.2 在 Ubuntu 16.04 上安装 Keras

在安装 Keras 之前，我们必须安装 Theano 和 TensorFlow 软件包及其依赖项。除此之外，请确认你的操作系统已经安装了 Python。接下来是 Python 的安装过程介绍。

i Conda 是一个运行在多个 OS（Windows、macOS 和 Linux）上的开源软件包管理系统和环境管理系统。用于在 Python 环境下安装多个版本的软件包及其依赖项，并能在本地计算机上对各个版本进行创建、保存、加载和环境切换。

1.2.1 准备工作


首先，你需要确保在本地或云端具有完整的 Ubuntu 16.04 操作系统。

1.2.2 怎么做

接下来将介绍在安装 Keras 之前必须安装的各个组件。

安装 miniconda

首先，为了更方便地安装所需软件包，你需要先进行 miniconda 的安装。miniconda 是 conda 软件包管理器的精简版本，可以用它进行 Python 虚拟环境的创建。

 建议读者安装 Python 2.7 或 Python 3.4, Python = 2.7 * 或 (≥ 3.4 和 <3.6), 并安装 Python 开发包 (Linux 发行的 `python-dev` 或 `python-devel`)。本书将基于 Python 2.7 进行讲解。

1. 为安装 miniconda, 首先从 continuum 版本库下载所需 sh 安装文件:


```
wget
https://repo.continuum.io/miniconda/Miniconda2-latest-Linux-x86_64.
sh
chmod 755 Miniconda2-latest-Linux-x86_64.sh
./Miniconda2-latest-Linux-x86_64.sh
```

2. 成功安装 conda 后, 我们就可以用它来安装 Theano、TensorFlow 和 Keras 的依赖项。

安装 numpy 和 scipy

numpy 和 scipy 是进行 Theano 安装的前提条件, 建议安装以下版本:

- NumPy: 不低于 1.9.1 且不高于 1.12。
- SciPy: 不低于 0.14 且低于 0.17.1, 用于处理稀疏矩阵和 Theano 中支持的一些特殊函数时强烈推荐, 否则 SciPy 0.8 以上版本即可以满足需求。
- 建议安装 BLAS (具有 Level 3 功能): 可通过 conda 与 `mkl-service` 包免费获得 MKL 数据库, 在该库内找到 BLAS。

 基本线性代数子程序库 (BLAS) 提供规范的线性代数运算程序, 程序采用 C 或 Fortran 编写。包括例如向量加法、标量乘法、点积、线性组合和矩阵乘法, 其中 Level 3 对应矩阵与矩阵的乘法运算。

1. 执行以下命令安装 `numpy` 和 `scipy` (确保 `conda` 在你的 `PATH` 中):

```
conda install numpy
conda install scipy
```

`scipy` 安装时的输出如下所示, 注意 `libgfortran` 安装也是 `scipy` 安装中的一部分:

```
Fetching package metadata .....
Solving package specifications: .
Package plan for installation in environment
/home/ubuntu/miniconda2:
```

2. 以下软件包也同时被安装:

```
libgfortran-ng: 7.2.0-h9f7466a_2
scipy: 1.0.0-py27hf5f0f52_0
Proceed ([y]/n)?
libgfortran-ng 100%
|#####|
Time: 0:00:00 36.60 MB/s
scipy-1.0.0-py 100%
|#####|
Time: 0:00:00 66.62 MB/s
```

安装 mkl

1. `mkl` 是用于 Intel 及兼容处理器的数学库。它是 `numpy` 的一部分, 但我们需要在安装 `Theano` 和 `TensorFlow` 之前先安装它:

```
conda install mkl
```

安装时的输出如下所示, 本例中, `miniconda2` 已经安装了最新版的 `mkl`:

```
Fetching package metadata .....
Solving package specifications: .
# All requested packages already installed.
# packages in environment at /home/ubuntu/miniconda2:
#
mkl 2018.0.1 h19d6760_4
```

2. 当完成上述安装后, 就可以开始安装 `TensorFlow` 了。

安装 TensorFlow

1. 通过执行以下命令, 利用 `conda` 开始安装 `tensorflow`:

```
conda install -c conda-forge tensorflow
```

如下所示，执行该命令可获取元数据并安装一系列的程序包：

```
Fetching package metadata .....
Solving package specifications: .
Package plan for installation in environment
/home/ubuntu/miniconda2:
```

2. 继续安装以下程序包：

```
bleach: 1.5.0-py27_0 conda-forge
funcsigs: 1.0.2-py_2 conda-forge
futures: 3.2.0-py27_0 conda-forge
html5lib: 0.9999999-py27_0 conda-forge
markdown: 2.6.9-py27_0 conda-forge
mock: 2.0.0-py27_0 conda-forge
pbr: 3.1.1-py27_0 conda-forge
protobuf: 3.5.0-py27_0 conda-forge
tensorboard: 0.4.0rc3-py27_0 conda-forge
tensorflow: 1.4.0-py27_0 conda-forge
webencodings: 0.5-py27_0 conda-forge
werkzeug: 0.12.2-py_1 conda-forge
```

3. 高版本的程序包将取代低版本的包：

```
conda: 4.3.30-py27h6ae6dc7_0 --> 4.3.29-py27_0 conda-forge
conda-env: 2.6.0-h36134e3_1 --> 2.6.0-0 conda-forge
Proceed ([y]/n)? y
conda-env-2.6. 100%
|#####|
Time: 0:00:00 1.67 MB/s
...
mock-2.0.0-py2 100%
|#####|
Time: 0:00:00 26.00 MB/s
conda-4.3.29-p 100%
|#####|
Time: 0:00:00 27.46 MB/s
```

4. 使用以下命令创建名为 `hello_tf.py` 的新文件，来测试是否成功安装了

TensorFlow：

```
vi hello_tf.py
```

5. 将以下代码添加到此文件并保存：

```
import tensorflow as tf
hello = tf.constant('Greetings, TensorFlow!')
sess = tf.Session()
print(sess.run(hello))
```

6. 在命令行执行已创建的文件:

```
python hello_tf.py
```

若得到如下输出则证明你已经成功安装了 TensorFlow 库:

```
Greetings, TensorFlow!
```

安装 Keras

i conda-forge 是 GitHub 平台上带有 conda 的工具。

1. 使用 conda-forge 上的 conda 安装 Keras。
2. 在终端上执行以下命令:

```
conda install -c conda-forge keras
```

以下输出表明 Keras 已成功安装:

```
Fetching package metadata .....
Solving package specifications: .
Package plan for installation in environment
/home/ubuntu/miniconda2:
```

同时, 以下软件包会被自动安装:

```
h5py: 2.7.1-py27_2 conda-forge
hdf5: 1.10.1-1 conda-forge
keras: 2.0.9-py27_0 conda-forge
libgfortran: 3.0.0-1
pyyaml: 3.12-py27_1 conda-forge
Proceed ([y]/n)? y
libgfortran-3. 100%
|#####|
Time: 0:00:00 35.16 MB/s
hdf5-1.10.1-1. 100%
|#####|
Time: 0:00:00 34.26 MB/s
pyyaml-3.12-py 100%
|#####|
Time: 0:00:00 60.08 MB/s
h5py-2.7.1-py2 100%
|#####|
Time: 0:00:00 58.54 MB/s
keras-2.0.9-py 100%
|#####|
Time: 0:00:00 45.92 MB/s
```


3. 使用以下代码检验 Keras 的安装：

```
$ python
Python 2.7.14 |Anaconda, Inc.| (default, Oct 16 2017, 17:29:19)
```

4. 执行以下命令以验证 Keras 是否成功安装：

```
> from keras.models import Sequential
Using TensorFlow backend.
>>>
```

可以看到，此时 Keras 基于 TensorFlow 的后端。

在 Theano 后端上运行 Keras

1. 通过修改默认配置（`keras.json` 文件）将 Keras 的后端由 TensorFlow 更改为 Theano：

```
vi .keras/keras.json
```

默认文件的内容如下：

```
{ "image_data_format": "channels_last",
  "epsilon": 1e-07,
  "floatx": "float32",
  "backend": "tensorflow"
}
```

2. 修改后的文件内容如下，其中“`backend`”已更改为“`theano`”：

```
{ "image_data_format": "channels_last",
  "epsilon": 1e-07,
  "floatx": "float32",
  "backend": "theano"
}
```

3. 运行 Python 控制台，并基于 Theano 后端从 `keras.model` 导入 `Sequential`：

```
$ python
Python 2.7.14 |Anaconda, Inc.| (default, Oct 16 2017, 17:29:19)
[GCC 7.2.0] on linux2
Type "help", "copyright", "credits" or "license" for more
information.
>>> from keras.models import Sequential
```

可以看到，此时 Keras 的后端已改为 Theano。

至此，我们安装了 `miniconda`、`Theano` 和 `TensorFlow` 的所有依赖项，以及 `TensorFlow`