

Natural Language Processing with TensorFlow

TensorFlow 自然语言处理

[澳] 图珊·加内格达拉 (Thushan Ganegedara) 著

马恩驰 陆健 译



■ ■ ■ 智能系统与技术丛书

Natural Language Processing with TensorFlow

TensorFlow 自然语言处理

[澳] 图珊·加内格达拉 (Thushan Ganegedara) 著

马恩驰 陆健 译

RFID



机械工业出版社
China Machine Press

图书在版编目 (CIP) 数据

TensorFlow 自然语言处理 / (澳) 图珊·加内格达拉 (Thushan Ganegedara) 著; 马恩驰, 陆健译. —北京: 机械工业出版社, 2019.6

(智能系统与技术丛书)

书名原文: Natural Language Processing with TensorFlow

ISBN 978-7-111-62914-6

I. T… II. ①图… ②马… ③陆… III. 人工智能—算法 IV. TP18

中国版本图书馆 CIP 数据核字 (2019) 第 106307 号

本书版权登记号: 图字 01-2018-6319

Thushan Ganegedara: Natural Language Processing with TensorFlow (ISBN: 978-1-78847-831-1).
Copyright © 2018 Packt Publishing. First published in the English language under the title “Natural Language Processing with TensorFlow”.

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。

TensorFlow 自然语言处理

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: 杨宴蕾

责任校对: 殷虹

印刷: 北京诚信伟业印刷有限公司

版次: 2019 年 6 月第 1 版第 1 次印刷

开本: 186mm × 240mm 1/16

印张: 20

书号: ISBN 978-7-111-62914-6

定价: 99.00 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

译者序

近几年来，自然语言处理（NLP）技术已经取得了阶段性进展，在电商、金融、翻译、智能硬件、医疗、旅游等行业广泛应用，应用场景涵盖语音交互、文本分类、情感分析、意图分析、图像识别等。在深度学习技术的驱动下，自然语言处理技术应用又上了一个台阶。这其中比较有代表性的是语音交互技术，在深度学习的驱动下，语音识别、声纹识别、语音合成等技术已经大规模应用于工业领域。以天猫精灵为代表的智能音箱也已经走入家庭生活中。根据 Statista 统计数据显示，在 2018 年，全球自然语言处理的市场规模达 5.83 亿美元，到 2024 年预计将达到约 21 亿美元，自然语言处理技术的应用前景广阔。

TensorFlow 作为广泛使用的深度学习框架，在自然语言处理领域被广泛使用。比较成熟的应用场景有搜索引擎、个性化推荐、商业化广告、语音识别等。本书主要介绍 NLP 在 TensorFlow 中的实现，内容涉及词嵌入的各种方法、CNN/RNN/LSTM 的 TensorFlow 实现及应用、LSTM 在文本生成及图像标题生成方面的应用以及从统计机器翻译到神经网络翻译的转变，最后探讨自然语言处理的未来。通过结合 TensorFlow 的代码实现，可以让读者更容易理解自然语言处理技术的原理。

回顾 2018 年，自然语言处理技术的一个聚焦点是谷歌的 BERT（Bidirectional Encoder Representations from Transformers）。谷歌 AI 团队新发布的 BERT 模型在机器阅读理解顶级水平测试 SQuAD1.1 中表现出惊人的成绩：两项衡量指标上全面超越人类，并且还在 11 种不同 NLP 测试中创出最佳成绩。毋庸置疑，BERT 模型开启了 NLP 的新时代。BERT 是基于 Transformer 的双向编码器表征。与其他语言表征模型不同，BERT 旨在通过联合调节所有层中的上下文来预训练深度双向表征。BERT 的另一个亮点是预训练，在预训练过程中作者随机掩蔽 15% 的标记，随机掩蔽的时候 10% 的单词会被替代成其他单词，10% 的单词不替换，剩下 80% 才被替换为 [MASK]。在预训练语料选取上，作者强调语料的选取很关键，要选用文档级别的语料而不是句子级别的，这样可以具备抽象表达连续长序列特征的能力。

如果说 BERT 是预训练语言模型的代表，那么无监督机器翻译、常识推理、元学习、理解表示、使用大文档的 QA 问答系统和推理等领域在 2018 年一样备受关注。

尽管 NLP 在过去一段时间取得阶段性进展，但仍需要继续突破。比如 BERT 仍然强依赖于训练数据，阅读理解还是在给定问句的情况下从文章中抽取答案，推理进展相对缓慢。当前自然语言处理主要基于 CNN 和 RNN 及各种衍生，问答模型通常会阶段收敛，学习不到语言真正深层的表达。综上所述，当前自然语言处理的水平仍在推理和深层表征上与人类有较大的差距，还有很长一段路要走。

本书是我和陆健利用业余时间合作翻译完成的。第 1 ~ 7 章由陆健翻译，内容涉及 TensorFlow 介绍、词嵌入的各种方法、CNN/RNN/LSTM 的 TensorFlow 实现及应用。第 8 ~ 11 章及附录由我翻译，内容涉及 LSTM 在文本生成及图像标题生成方面的应用、从统计机器翻译到神经网络翻译的转变以及自然语言处理的未来。由于译者水平有限，翻译过程难免会有瑕疵，如有相关问题请发邮件至 maec1208@gmail.com。

感谢华章公司编辑在翻译过程中给予的协助，感谢我的爱人和孩子在本书翻译过程中给予的理解与支持。

马恩驰

人工智能实验室 @ 杭州

2019 年 3 月

前 言

在我们所处的数字信息时代，数据量呈指数级增长，在我们阅读本书时，它也正以前所未有的速度增长。此类数据大多数是与语言相关的数据（文本或语言），例如电子邮件、社交媒体帖子、电话和网络文章。自然语言处理（NLP）有效地利用这些数据来帮助人们开展业务或日常工作。NLP 已经彻底改变了我们使用数据改善业务和生活的方式，并且这种改变将在未来一直持续。

NLP 最普遍的应用案例之一是虚拟助手（VA），例如 Apple 的 Siri、Google 的 Assistant 和 Amazon 的 Alexa。当你向 VA 询问“瑞士最便宜的酒店价格”时，就会触发一系列复杂的 NLP 任务。首先，VA 需要了解（解析）你的请求（例如，它需要知道你要检索酒店价格，而不是寻找附近的公园）。VA 需要做出的另一个决定是“哪家酒店便宜？”接下来，VA 需要对瑞士的城市进行排名（可能基于你过去的旅行历史）。然后，VA 可能会访问 Booking.com 和 Agoda.com 等网站，获取瑞士的酒店价格，并通过分析每家酒店的价格和评论对其进行排名。因此，你在几秒钟内看到的是执行了一系列非常复杂的 NLP 任务的结果。

那么，是什么使得这些 NLP 任务在处理我们的日常任务时如此聪明和准确？其底层方法是深度学习算法。深度学习算法本质上是复杂的神经网络，它可以将原始数据映射到所需的输出，而无须针对特定任务执行任何特征工程。这意味着只需提供客户的酒店评论，算法就可以直接回答“客户对这家酒店的评价如何？”这样的问题。此外，深度学习已经在一系列 NLP 任务（例如，语音识别和机器翻译）中达到甚至超过了人类的表现。

通过阅读本书，你可以学习如何使用深度学习去解决许多有趣的 NLP 问题。如果你想成为一个改变世界的人，那么研究 NLP 是至关重要的。这些任务包括学习单词的语义，生成全新的故事，以及通过研究双语句对进行语言翻译。所有技术章节都附有练习，这些练习会指导读者逐步实现这些系统。对于本书中的所有练习，我们都使用基于 Python 的 TensorFlow 库，TensorFlow 是一种流行的分布式计算库，可以非常方便地实现深度神经网络。

本书读者

本书适用于那些有志于利用语言数据改造世界的初学者。本书将为你提供解决 NLP 任务的坚实基础。在本书中，我们将涵盖 NLP 的各个方面，更多地关注实际应用而不是理论基础。等到学习这些方法的更高级理论时，拥有解决各种 NLP 任务的良好实践知识将帮助你实现更平稳的过渡。此外，扎实的实践知识可以帮助你最大限度地将算法从一个特定领域迁移到更多领域。

本书内容

第 1 章是对 NLP 的简单介绍。该章将首先讨论我们需要 NLP 的原因。接下来，将讨论 NLP 中一些常见的子任务。之后，将讨论 NLP 的两个主要阶段，即传统阶段和深度学习阶段。通过研究如何使用传统算法解决语言建模任务，我们将了解传统阶段 NLP 的特点。然后，将讨论深度学习阶段，在这一阶段中深度学习算法被大量用于 NLP。我们还将讨论深度学习算法的主要系列。最后，将讨论一种最基本的深度学习算法：全连接神经网络。该章结束时提供一份路线图，简要介绍后面的内容。

第 2 章介绍 Python TensorFlow 库，这是我们实现解决方案的主要平台。首先在 TensorFlow 中编写一段代码，执行一个简单的计算，并讨论从运行代码到得到结果这一过程中到底发生了什么。我们将详细介绍 TensorFlow 的基础组件。把 Tensorflow 比作丰富的餐厅，了解如何完成订单，以便进一步加强对 TensorFlow 的理解。稍后，将讨论 TensorFlow 的更多技术细节，例如数据结构和操作（主要与神经网络相关）。最后，我们将实现一个全连接的神经网络来识别手写数字。这将帮助我们了解如何使用 TensorFlow 来实现端到端解决方案。

第 3 章首先讨论如何用 TensorFlow 解决 NLP 任务。在该章中，我们将讨论如何用神经网络学习单词向量或单词表示。单词向量也称为词嵌入。单词向量是单词的数字表示，相似单词有相似值，不同单词有不同值。首先，将讨论实现这一目标的几种传统方法，包括使用称为 WordNet 的大型人工构建知识库。然后，将讨论基于现代神经网络的方法，称为 Word2vec，它在没有任何人为干预的情况下学习单词向量。我们将通过一个实例来了解 Word2vec 的机制。接着，将讨论用于实现此目的两种算法变体：skip-gram 和连续词袋 (CBOW) 模型。我们将讨论算法的细节，以及如何在 TensorFlow 中实现它们。

第 4 章介绍与单词向量相关的更高级方法。首先，会比较 skip-gram 和 CBOW，讨论其

中哪一种有明显优势。接下来，将讨论可用于提高 Word2vec 算法性能的几项改进。然后，将讨论一种更新、更强大的词嵌入学习算法：GloVe（全局向量）算法。最后，将在文档分类任务中实际观察单词向量。在该练习中，我们将看到单词向量十分强大，足以表示文档所属的主题（例如，娱乐和运动）。

第 5 章讨论卷积神经网络（CNN），它是擅长处理诸如图像或句子这样的空间数据的神经网络家族。首先，讨论如何处理数据以及处理数据时涉及哪种操作，以便对 CNN 有较深的理解。接下来，深入研究 CNN 计算中涉及的每个操作，以了解 CNN 背后的数学原理。最后，介绍两个练习。第一个练习使用 CNN 对手写数字图像进行分类，我们将看到 CNN 能够在此任务上很快达到较高的准确率。接下来，我们将探讨如何使用 CNN 对句子进行分类。特别地，我们要求 CNN 预测一个句子是否与对象、人物、位置等相关。

第 6 章介绍递归神经网络。递归神经网络（RNN）是一个可以模拟数据序列的强大的神经网络家族。首先讨论 RNN 背后的数学原理以及在学习期间随时间更新 RNN 的更新规则。然后，讨论 RNN 的不同变体及其应用（例如，一对一 RNN 和一对多 RNN）。最后，用 RNN 执行文本生成任务的练习。我们用童话故事训练 RNN，然后要求 RNN 生成一个新故事。我们将看到在持久的长期记忆方面 RNN 表现不佳。最后，讨论更高级的 RNN 变体，即 RNN-CF，它能够保持更长时间的记忆。

第 7 章介绍长短期记忆网络。RNN 在保持长期记忆方面效果较差，这使我们需要探索能在更长时间内记住信息的更强大技术。我们将在该章讨论一种这样的技术：长短期记忆网络（LSTM）。LSTM 功能更强大，并且在许多时间序列任务中表现得优于其他序列模型。首先通过一个例子，研究潜在的数学原理和 LSTM 的更新规则，以说明每个计算的重要性。然后，将了解为什么 LSTM 能够更长时间地保持记忆。接下来，将讨论如何进一步提高 LSTM 预测能力。最后，将讨论具有更复杂结构的几种 LSTM 变体（具有窥孔连接的 LSTM），以及简化 LSTM 门控循环单元（GRU）的方法。

第 8 章介绍 LSTM 的应用：文本生成。该章广泛评估 LSTM 在文本生成任务中的表现。我们将定性和定量地衡量 LSTM 产生的文本的好坏程度，还将比较 LSTM、窥孔连接 LSTM 和 GRU。最后，将介绍如何将词嵌入应用到模型中来改进 LSTM 生成的文本。

第 9 章转到对多模态数据（即图像和文本）的处理。在该章中，我们将研究如何自动生成给定图像的描述。这涉及将前馈模型（即 CNN）与词嵌入层及顺序模型（即 LSTM）组合，形成一个端到端的机器学习流程。

第 10 章介绍有关神经机器翻译（NMT）模型的应用。机器翻译指的是将句子或短语从

源语言翻译成目标语言。首先讨论机器翻译是什么并简单介绍机器翻译历史。然后，将详细讨论现代神经机器翻译模型的体系结构，包括训练和预测的流程。接下来，将了解如何从头开始实现 NMT 系统。最后，会探索改进标准 NMT 系统的方法。

第 11 章重点介绍 NLP 的现状和未来趋势。我们将讨论前面提到的系统的相关最新发现。该章将涵盖大部分令人兴奋的创新，并让你直观地感受其中的一些技术。

附录向读者介绍各种数学数据结构（例如，矩阵）和操作（例如，矩阵的逆），还将讨论概率中的几个重要概念。然后将介绍 Keras，它是在底层使用 TensorFlow 的高级库。Keras 通过隐藏 TensorFlow 中的一些有难度的细节使得神经网络的实现更简单。具体而言，通过使用 Keras 实现 CNN 来介绍如何使用 Keras。接下来，将讨论如何使用 TensorFlow 中的 seq2seq 库来实现一个神经机器翻译系统，所使用的代码比在第 11 章中使用的代码少得多。最后，将向你介绍如何使用 TensorBoard 可视化词嵌入的指南。TensorBoard 是 TensorFlow 附带的便捷可视化工具，可用于可视化和监视 TensorFlow 客户端中的各种变量。

如何充分利用本书

为了充分利用本书，读者需要具备以下能力：

- 有强烈的意愿和坚定的意志学习 NLP 的先进技术。
- 熟悉 Python 的基本语法和数据结构（例如，列表和字典）。
- 理解基本的数学原理（例如，矩阵或向量的乘法）。
- （可选）对于一些小节，需要高级的数学知识（例如，微分计算）来理解特定模型是如何在训练时克服潜在的实际问题的。
- （可选）对超出本书的内容，可以阅读相关论文以获取最新进展或细节。

下载示例代码及彩色图像

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。

这些代码还可在 GitHub 上获取，网址是：<https://github.com/PacktPublishing/Natural-Language-Processing-with-TensorFlow>。

ABOUT THE AUTHOR

关于作者

图珊·加内格达拉 (Thushan Ganegedara) 目前是澳大利亚悉尼大学第三年的博士生。他专注于机器学习和深度学习。他喜欢在未经测试的数据上运行算法。他还是澳大利亚初创公司 AssessThreat 的首席数据科学家。他在斯里兰卡莫拉图瓦大学获得了理学士学位。他经常撰写有关机器学习的技术文章和教程。此外，他经常通过游泳来努力营造健康的生活方式。

感谢我的父母、兄弟姐妹和我的妻子，感谢他们对我的信任以及给予我的支持，感谢我所有的老师和博士生导师提供的指导。

ABOUT THE REVIEWERS

关于审阅者

Motaz Saad 拥有洛林大学计算机专业博士学位，他喜欢数据并以此为乐。他在 NLP、计算语言学、数据科学和机器学习领域拥有超过 10 年的专业经验，目前担任 IUG 信息技术学院的助理教授。

Joseph O'Connor 博士是一名对深度学习充满热情的数据科学家。他的公司 Deep Learn Analytics 是一家总部位于英国的数据科学咨询公司，旨在与企业合作，开发从概念到部署的机器学习应用程序和基础架构。因为对 MINOS 高能物理实验的数据分析所做的研究，他获得伦敦大学学院授予的博士学位。从那时起，他为许多私营公司开发 ML 产品，主攻 NLP 和时间序列预测。你可以在 <http://deeplearnanalytics.com/> 找到他的相关信息。

目 录

译者序	1.6.3 安装 Jupyter Notebook	15
前言	1.6.4 安装 TensorFlow	16
关于作者	1.7 总结	17
关于审阅者		
第 1 章 自然语言处理简介	第 2 章 理解 TensorFlow	18
1.1 什么是自然语言处理	2.1 TensorFlow 是什么	18
1.2 自然语言处理的任务	2.1.1 TensorFlow 入门	19
1.3 传统的自然语言处理方法	2.1.2 TensorFlow 客户端详细介绍	21
1.3.1 理解传统方法	2.1.3 TensorFlow 架构：当你执行客户端时发生了什么	21
1.3.2 传统方法的缺点	2.1.4 Cafe Le TensorFlow：使用类比理解 TensorFlow	23
1.4 自然语言处理的深度学习方法	2.2 输入、变量、输出和操作	24
1.4.1 深度学习的历史	2.2.1 在 TensorFlow 中定义输入	25
1.4.2 深度学习和 NLP 的当前状况	2.2.2 在 TensorFlow 中定义变量	30
1.4.3 理解一个简单的深层模型——全连接神经网络	2.2.3 定义 TensorFlow 输出	31
1.5 本章之外的学习路线	2.2.4 定义 TensorFlow 操作	31
1.6 技术工具简介	2.3 使用作用域重用变量	40
1.6.1 工具说明	2.4 实现我们的第一个神经网络	42
1.6.2 安装 Python 和 scikit-learn	2.4.1 准备数据	43

2.4.2	定义 TensorFlow 图	43	4.1.1	实现原始 skip-gram 算法	73
2.4.3	运行神经网络	45	4.1.2	比较原始 skip-gram 算法 和改进的 skip-gram 算法	75
2.5	总结	46	4.2	比较 skip-gram 算法和 CBOW 算法	75
第 3 章 Word2vec——学习词 嵌入			4.2.1	性能比较	77
3.1	单词的表示或含义是什么	49	4.2.2	哪个更胜一筹: skip-gram 还是 CBOW	79
3.2	学习单词表示的经典方法	49	4.3	词嵌入算法的扩展	81
3.2.1	WordNet——使用外部 词汇知识库来学习单词 表示	50	4.3.1	使用 unigram 分布进行负 采样	81
3.2.2	独热编码表示方式	53	4.3.2	实现基于 unigram 的负 采样	81
3.2.3	TF-IDF 方法	53	4.3.3	降采样: 从概率上忽视 常用词	83
3.2.4	共现矩阵	54	4.3.4	实现降采样	84
3.3	Word2vec——基于神经网络学习 单词表示	55	4.3.5	比较 CBOW 及其扩展 算法	84
3.3.1	练习: queen = king - he + she 吗	56	4.4	最近的 skip-gram 和 CBOW 的 扩展算法	85
3.3.2	为学习词嵌入定义损失 函数	58	4.4.1	skip-gram 算法的限制	85
3.4	skip-gram 算法	59	4.4.2	结构化 skip-gram 算法	85
3.4.1	从原始文本到结构化的 数据	59	4.4.3	损失函数	86
3.4.2	使用神经网络学习词 嵌入	60	4.4.4	连续窗口模型	87
3.4.3	使用 TensorFlow 实现 skip-gram	67	4.5	GloVe: 全局向量表示	88
3.5	连续词袋算法	69	4.5.1	理解 GloVe	88
3.6	总结	71	4.5.2	实现 GloVe	89
第 4 章 高级 Word2vec			4.6	使用 Word2vec 进行文档分类	90
4.1	原始 skip-gram 算法	72	4.6.1	数据集	91
			4.6.2	用词向量进行文档分类	91

4.6.3	实现：学习词嵌入	92	第 6 章 递归神经网络	116
4.6.4	实现：词嵌入到文档 嵌入	92	6.1 理解递归神经网络	116
4.6.5	文本聚类以及用 t-SNE 可视化文档嵌入	93	6.1.1 前馈神经网络的问题	117
4.6.6	查看一些特异点	94	6.1.2 用递归神经网络进行 建模	118
4.6.7	实现：用 K-means 对文档 进行分类 / 聚类	95	6.1.3 递归神经网络的技术 描述	119
4.7	总结	96	6.2 基于时间的反向传播	119
第 5 章 用卷积神经网络进行 句子分类		97	6.2.1 反向传播的工作原理	120
5.1 介绍卷积神经网络		97	6.2.2 为什么 RNN 不能直接 使用反向传播	120
5.1.1 CNN 基础		97	6.2.3 基于时间的反向传播： 训练 RNN	121
5.1.2 卷积神经网络的力量		100	6.2.4 截断的 BPTT：更有效地 训练 RNN	121
5.2 理解卷积神经网络		100	6.2.5 BPTT 的限制：梯度消失 和梯度爆炸	122
5.2.1 卷积操作		100	6.3 RNN 的应用	123
5.2.2 池化操作		103	6.3.1 一对一 RNN	123
5.2.3 全连接层		104	6.3.2 一对多 RNN	123
5.2.4 组合成完整的 CNN		105	6.3.3 多对一 RNN	124
5.3 练习：在 MNIST 数据集上用 CNN 进行图片分类		105	6.3.4 多对多 RNN	124
5.3.1 关于数据		106	6.4 用 RNN 产生文本	125
5.3.2 实现 CNN		106	6.4.1 定义超参数	125
5.3.3 分析 CNN 产生的预测 结果		108	6.4.2 将输入随时间展开用于 截断的 BPTT	125
5.4 用 CNN 进行句子分类		109	6.4.3 定义验证数据集	126
5.4.1 CNN 结构		110	6.4.4 定义权重和偏置	126
5.4.2 随时间池化		112	6.4.5 定义状态持续变量	127
5.4.3 实现：用 CNN 进行句子 分类		112	6.4.6 用展开的输入计算隐藏 状态和输出	127
5.5 总结		115		

6.4.7	计算损失	128	第 8 章 LSTM 应用：文本生成	160	
6.4.8	在新文本片段的开头 重置状态	128	8.1 数据集	160	
6.4.9	计算验证输出	128	8.1.1 关于数据集	160	
6.4.10	计算梯度和优化	129	8.1.2 数据预处理	162	
6.4.11	输出新生成的文本块	129	8.2 实现 LSTM	162	
6.5	评估 RNN 的文本结果输出	130	8.2.1 定义超参数	163	
6.6	困惑度：衡量文本结果的质量	131	8.2.2 定义参数	163	
6.7	有上下文特征的递归神经网络： 更长记忆的 RNN	132	8.2.3 定义 LSTM 单元及操作	165	
6.7.1	RNN-CF 的技术描述	132	8.2.4 定义输入和标签	165	
6.7.2	实现 RNN-CF	133	8.2.5 定义处理序列数据所需的 序列计算	166	
6.7.3	RNN-CF 产生的文本	138	8.2.6 定义优化器	167	
6.8	总结	140	8.2.7 随时间衰减学习率	167	
第 7 章 长短期记忆网络	142	8.2.8 做预测	168	8.2.9 计算困惑度（损失）	168
7.1 理解长短期记忆网络	142	8.2.10 重置状态	169	8.2.11 贪婪采样避免单峰	169
7.1.1 LSTM 是什么	143	8.2.12 生成新文本	169	8.2.13 生成的文本样例	170
7.1.2 更详细的 LSTM	144	8.3 LSTM 与窥孔 LSTM 和 GRU 对比	171	8.3.1 标准 LSTM	171
7.1.3 LSTM 与标准 RNN 的 区别	149	8.3.2 门控循环单元 (GRU)	172	8.3.3 窥孔 LSTM	174
7.2 LSTM 如何解决梯度消失 问题	150	8.3.4 训练和验证随时间的 困惑度	175	8.4 改进 LSTM：集束搜索	176
7.2.1 改进 LSTM	152	8.4.1 实现集束搜索	177	8.4.2 集束搜索生成文本的 示例	179
7.2.2 贪婪采样	153				
7.2.3 集束搜索	153				
7.2.4 使用词向量	154				
7.2.5 双向 LSTM (BiLSTM)	155				
7.3 其他 LSTM 的变体	156				
7.3.1 窥孔连接	156				
7.3.2 门循环单元	157				
7.4 总结	159				

8.5 LSTM 改进：用单词替代 n-gram	9.8 定义 LSTM	201
生成文本	9.9 定量评估结果	203
8.5.1 维度灾难	9.9.1 BLEU	203
8.5.2 Word2vec 补救	9.9.2 ROUGE	204
8.5.3 使用 Word2vec 生成	9.9.3 METEOR	204
文本	9.9.4 CIDEr	206
8.5.4 使用 LSTM-Word2vec	9.9.5 模型随着时间变化的	
和集束搜索生成的文本	BLEU-4	206
示例	9.10 为测试图像生成标题	207
8.5.5 随时间困惑度	9.11 使用 TensorFlow RNN API 和	
8.6 使用 TensorFlow RNN API	预训练的 GloVe 词向量	210
8.7 总结	9.11.1 加载 GloVe 词向量	210
	9.11.2 清洗数据	212
	9.11.3 使用 TensorFlow RNN API	
	和预训练的词嵌入	213
	9.12 总结	218
第 9 章 LSTM 应用：图像标题		
生成		188
9.1 了解数据		188
9.1.1 ILSVRC ImageNet		
数据集		189
9.1.2 MS-COCO 数据集		189
9.2 图像标题生成实现路径		191
9.3 使用 CNN 提取图像特征		193
9.4 实现：使用 VGG-16 加载权重		
和推理		193
9.4.1 构建和更新变量		194
9.4.2 预处理输入		195
9.4.3 VGG-16 推断		196
9.4.4 提取图像的向量化表达		197
9.4.5 使用 VGG-16 预测类别		
概率		197
9.5 学习词嵌入		198
9.6 准备输入 LSTM 的标题		198
9.7 生成 LSTM 的数据		199
	第 10 章 序列到序列学习：神经	
	机器翻译	220
	10.1 机器翻译	220
	10.2 机器翻译简史	221
	10.2.1 基于规则的翻译	221
	10.2.2 统计机器翻译 (SMT)	222
	10.2.3 神经机器翻译 (NMT)	223
	10.3 理解神经机器翻译	225
	10.3.1 NMT 原理	225
	10.3.2 NMT 架构	226
	10.4 为 NMT 系统准备数据	228
	10.4.1 训练阶段	229
	10.4.2 反转源句	229
	10.4.3 测试阶段	230
	10.5 训练 NMT	230

10.6	NMT 推理	231	10.12	序列到序列模型的其他应用： 聊天机器人	256
10.7	BLEU 评分：评估机器翻译 系统	232	10.12.1	训练聊天机器人	256
10.7.1	修正的精确度	232	10.12.2	评估聊天机器人： 图灵测试	257
10.7.2	简短惩罚项	233	10.13	总结	258
10.7.3	最终 BLEU 得分	233	第 11 章 自然语言处理的现状 与未来		
10.8	从头开始实现 NMT：德语到 英语的翻译	233	11.1	NLP 现状	259
10.8.1	数据介绍	234	11.1.1	词嵌入	260
10.8.2	处理数据	234	11.1.2	神经机器翻译	264
10.8.3	学习词嵌入	235	11.2	其他领域的渗透	266
10.8.4	定义编码器和解码器	236	11.2.1	NLP 与计算机视觉 结合	266
10.8.5	定义端到端输出计算	238	11.2.2	强化学习	268
10.8.6	翻译结果	239	11.2.3	NLP 生成式对抗网络	269
10.9	结合词嵌入训练 NMT	241	11.3	走向通用人工智能	270
10.9.1	最大化数据集词汇表和 预训练词嵌入之间的 匹配	241	11.3.1	一个模型学习全部	271
10.9.2	将嵌入层定义为 TensorFlow 变量	243	11.3.2	联合多任务模型：为 多个 NLP 任务生成 神经网络	272
10.10	改进 NMT	245	11.4	社交媒体 NLP	273
10.10.1	教师强迫	246	11.4.1	社交媒体中的谣言 检测	274
10.10.2	深度 LSTM	247	11.4.2	社交媒体中的情绪 检测	274
10.11	注意力	247	11.4.3	分析推特中的政治 框架	274
10.11.1	突破上下文向量 瓶颈	247	11.5	涌现的新任务	275
10.11.2	注意力机制细节	248	11.5.1	讽刺检测	275
10.11.3	注意力 NMT 的翻译 结果	253	11.5.2	语言基础	276
10.11.4	源句子和目标句子 注意力可视化	254			