



TURING

图灵程序设计丛书

[PACKT]
PUBLISHING

Deep Learning with TensorFlow

TensorFlow深度学习

【意】 Giancarlo Zaccone

【孟加拉】 Md. Rezaul Karim 著

【埃及】 Ahmed Menshawy

李志 译

- 学习建模、数据采集和转换等实际操作
- 帮助想快速入门的新手获取深度学习实战经验



中国工信出版集团



人民邮电出版社
POSTS & TELECOM PRESS



图灵程序设计丛书

Deep Learning with TensorFlow

TensorFlow深度学习

【意】 Giancarlo Zaccone

【孟加拉】 Md. Rezaul Karim 著

【埃及】 Ahmed Menshawy

李志 译

人民邮电出版社

北 京

图书在版编目(CIP)数据

TensorFlow深度学习 / (意) 吉安卡洛·扎克尼,
(孟加拉) 穆罕默德·礼萨·卡里姆, (埃及) 艾哈迈德·
门沙维著; 李志译. — 北京: 人民邮电出版社, 2018.3

(图灵程序设计丛书)

ISBN 978-7-115-47877-1

I. ①T… II. ①吉… ②穆… ③艾… ④李… III. ①
人工智能—算法—研究 IV. ①TP18

中国版本图书馆CIP数据核字(2018)第023704号

内 容 提 要

本书共分5方面内容: 基础知识、关键模块、算法模型、内核揭秘、生态发展。前两方面由浅入深地介绍了TensorFlow平台, 算法模型方面依托TensorFlow讲解深度学习模型, 内核揭秘方面主要分析C++内核中的通信原理、消息管理机制等, 最后从生态发展的角度讲解以TensorFlow为中心的一套开源大数据分析解决方案。

本书适合所有对深度学习和TensorFlow感兴趣的开发人员和数据分析师阅读。

-
- ◆ 著 [意] Giancarlo Zaccone [孟加拉] Md. Rezaul Karim
[埃及] Ahmed Menshawy
译 李 志
责任编辑 陈 曦
责任印制 周昇亮
- ◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路11号
邮编 100164 电子邮件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
北京鑫正大印刷有限公司印刷
- ◆ 开本: 800×1000 1/16
印张: 15
字数: 355千字 2018年3月第1版
印数: 1—3 500册 2018年3月北京第1次印刷
著作权合同登记号 图字: 01-2017-6485号
-

定价: 49.00元

读者服务热线: (010)51095186转600 印装质量热线: (010)81055316

反盗版热线: (010)81055315

广告经营许可证: 京东工商广登字 20170147 号

版权声明

Copyright © 2017 Packt Publishing. First published in the English language under the title *Deep Learning with TensorFlow*.

Simplified Chinese-language edition copyright © 2018 by Posts & Telecom Press. All rights reserved.

本书中文简体字版由Packt Publishing授权人民邮电出版社独家出版。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。

版权所有，侵权必究。

前 言

机器学习关注的问题是如何使用算法将未经处理的原始数据转化为信息，进而转化为可供决策的情报。这一特性使得机器学习十分适用于大数据预测分析领域。可以说，没有机器学习，人们几乎不可能处理和整合当今世界的海量信息。另一方面，深度学习是机器学习算法的一个分支，其基本思想是学习出多个表示层，作为数据的模型。近年来，人们设计并开发了很多性能强大的深度学习算法，应用于图像识别、自然语言处理等大量复杂任务。深度学习算法本质上无非是复杂神经网络的一种实现，使其能够通过海量数据的分析进行学习。本书基于最新版TensorFlow，介绍深度学习中的核心概念。TensorFlow是谷歌2011年发布的一个开源框架，用于处理数学、机器学习和深度学习领域的问题。自发布以来，TensorFlow大受欢迎，广泛应用于学术界、科研领域乃至工业领域。TensorFlow版本1.0含有统一的API，它提供了恰到好处的灵活性。用户可以轻松实现和研究最前沿的算法架构，集中精力关注模型的组织结构，而不用为其中的数学细节而烦恼。在学习本书的过程中，你将通过建模、数据收集和转换等实际操作学习深度学习编程技术。

祝阅读愉快！

本书内容

第1章介绍机器学习和深度学习的架构，包括所谓的“深度神经网络”。这种神经网络与普通的单隐藏层神经网络的不同主要在于其“深度”。“深度”指的是节点层的数量，也就是说，深度神经网络拥有多个层。在模式识别的过程中，数据需要依次通过这些层。这一章将用一个表格对比各种深度学习架构的优劣，其中总结了各种神经网络模型，大部分深度学习算法都由这些神经网络模型发展而来。

第2章介绍TensorFlow 1.x版本的主要特性和功能：计算图、数据模型、编程模型和TensorBoard可视化模块等。后半部分实现一个单输入神经元，以展示TensorFlow的实际应用。最后介绍如何将TensorFlow从0.x版本升级到1.x版本。

第3章详细介绍前馈神经网络的概念。这一章采用前馈神经网络这一基本架构实现诸多应用范例，具有很强的实用性。

第4章介绍基于深度学习的图像分类器的基本构件——卷积神经网络 (CNN)。通过两个示例实现,第一个是经典的MNIST数字分类问题,第二个是基于一系列人脸图像训练一个网络,用来对表情进行分类。

第5章展示自编码器网络的概念。设计和训练自编码器网络的目的是对输入模式进行转化,当输入模式退化或不完整时,可以对其进行补全,获取原始模式。这一章会使用几个示例介绍自编码器的实际操作。

第6章解释循环神经网络这一基本架构。循环神经网络用于处理长短不一的数据,在自然语言处理领域中的应用十分广泛。这一章将实现文本处理和图像分类。

第7章介绍TensorFlow中用于GPU计算的工具,并探索TensorFlow中处理GPU任务的一些技术。

第8章介绍几个基于TensorFlow的库:Keras、Pretty Tensor和TFLearn。针对每个库介绍其主要特性,并分别给出应用示例。

第9章涵盖TensorFlow多媒体编程的一些高级技术和新兴技术。这一章将讨论应用于可伸缩对象检测的深度神经网络,以及Android系统上的TensorFlow深度学习,并给出示例及代码。还将讲解加速线性代数(XLA)及Keras,并使用一些示例使讨论更加具体。

第10章涵盖强化学习的基本概念,并介绍Q-learning算法。该算法是使用最广的强化学习算法之一。此外还将介绍OpenAI gym框架,该框架与TensorFlow兼容,可作为强化学习算法的开发和对比评价工具。

预备工具

书中所有示例均采用Python 2.7和3.5版本实现,操作系统为基于Linux的64位Ubuntu系统,TensorFlow库版本为1.0.1。书中展示的所有源代码均兼容Python 2.7。

你还需要安装以下Python模块(最好是最新版本):

- Pip
- Bazel
- Matplotlib
- Numpy
- Pandas
- mnist_data

阅读第8~10章时,还需要以下框架:

- Keras
- XLA
- Pretty Tensor
- TFLearn
- OpenAI gym

请注意, TensorFlow的GPU启用版本需要满足以下要求: 64位Linux、Python 2.7(若使用Python 3, 则要求3.3以上版本)、NVIDIA CUDA® 7.5(对于Pascal GPU, 则需要CUDA 8.0版本)、NVIDIA cuDNN v4.0(最低)或v5.1(推荐)。另外, 当前TensorFlow对GPU计算的支持仅限于NVIDIA的工具、驱动和软件。

读者对象

如果你是不具备太高深的数学背景, 但又想了解深度学习的开发人员、数据分析师, 或是对深度学习感兴趣的学习者, 那么本书非常适合你。书中内容主要针对想要快速入门的新手, 帮助他们获取一些深度学习的实战经验。读者需要具备基本的编程能力, 掌握至少一门程序设计语言, 并熟悉计算机科学技术的基础知识, 如基本的硬件、算法知识等。另外, 还需具备基本的数学能力, 熟悉基础的线性代数和微积分知识等。

排版约定

在本书中, 你将会看到几种不同的文本格式, 用以区分不同信息。下面是这些文本格式的示例及其相应的含义。

文本中的代码、数据库表名、文件夹名、文件名、文件扩展名、路径名、模拟URL、用户输入及Twitter句柄等内容, 显示为: “若要保存一个模型, 需要使用Saver()类。”

代码块格式如下:

```
saver = tf.train.Saver()
save_path = saver.save(sess, "softmax_mnist")
print("Model saved to %s" % save_path)
```

所有命令行输入/输出采用以下格式显示:

```
$ sudo apt-get install python-pip python-dev
```

屏幕上显示的词语, 例如菜单或对话框中的词, 在文本中显示如下: “点击**GRAPH**选项卡, 可以看到该模型的**计算图**及其中**继节点**。”



警告或重要的注意事项。



提示或小技巧。

读者反馈

欢迎各位提出宝贵意见，请让我们知道你对本书的看法——喜欢什么或者不喜欢什么。读者反馈对我们非常重要，因为这可以帮助我们发现对大家最有帮助的主题。

要想提供反馈，只需登录“图灵社区”本书页面（<http://www.ituring.com.cn/book/2420>）并留言。

客户支持

如果您购买了我们出版的图书，我们将提供一系列服务来使您获得最大收益。

下载示例代码

你可以从“图灵社区”本书页面（<http://www.ituring.com.cn/book/2420>）下载书中示例代码。

文件下载结束之后，请确定使用以下软件的最新版本解压或提取文件：

- ❑ Windows上使用WinRAR/7-Zip
- ❑ Mac上使用Zipeg/iZip/UnRarX
- ❑ Linux上使用7-Zip/PeaZip

本书的代码也被托管在GitHub上，地址为<https://github.com/PacktPublishing/Deep-Learning-with-TensorFlow>。<https://github.com/PacktPublishing/Deep-Learning-with-TensorFlow>这个地址还提供了其他种类丰富的图书和视频资料相关代码包，好好看一下吧！

下载本书彩色图片

我们也提供含有彩色截图/图表的PDF文件。彩色图片能帮助你更深入地理解输出的变化。可以从https://www.packtpub.com/sites/default/files/downloads/DeepLearningwithTensorFlow_ColorImages.pdf下载此文件。

勘误

尽管我们做了各种努力来保证内容的准确性，依然无法避免出现错误。如果您在书中发现文字或代码错误并告知我们，我们将非常感谢。通过勘误，您可以提高其他读者的阅读体验，并可以帮助我们在本书的后续版本中做出改进。不管您发现什么错误，都可以通过“图灵社区”本书页面（<http://www.ituring.com.cn/book/2420>）告诉我们。一旦勘误通过确认，将显示在页面上的勘误表中。

反盗版

互联网上针对有版权资料的盗版行为一直存在，并逐步扩展到所有媒体。图灵公司非常重视对自己版权和许可的保护，如果您在互联网上发现对于我们工作的任何形式的非法复制行为，请立即将地址或网站名通知我们，我们会采取对策。

请联系ebook@turingbook.com并提供有盗版嫌疑的链接。

如果我们在作者保护和造福读者方面得到您的帮助，我们将非常感谢。

问题

对本书有任何疑问，都可以登录“图灵社区”本书页面（<http://www.ituring.com.cn/book/2420>），我们会尽最大努力解决问题。

电子书

如需购买本书电子版，请扫描以下二维码。



致 谢

Md. Rezaul Karim

我要感谢我的父母（Razzaque先生和Monoara夫人），他们在我的人生中一直给予我鼓励和动力。我也要感谢我的妻子（Saroar）和孩子（Shadman），他们不懈的支持使我不断前行。我尤其要感谢本书的另外两位作者Ahmed Menshawy和Giancarlo Zaccone。没有他们的贡献，本书是不可能完成的。最后，我要将本书献给我的哥哥Md. Mamtaz Uddin（孟加拉国Biopharma公司国际部经理），他对我的人生做出了不可估量的贡献。

Ahmed Menshawy

我要感谢我的父母、我的妻子Sara和女儿Asma在本书写作过程中对我的支持和耐心。我也要向本书的另外两个作者Md. Rezaul Karim和Giancarlo Zaccone表示真诚的谢意。

共同致谢

感谢Packt出版社的采购、内容开发和技术编辑（以及所有为本书做过工作的人）的真诚合作和协调。另外，如果没有无数研究人员和深度学习应用者们的工作，如果没有他们无私贡献的专业出版物、课程和源代码，本书根本不可能存在。最后，我们要感谢TensorFlow社区的支持及其会员贡献的API，这些宝贵资源最终使得人人都能享受深度学习的乐趣。

目 录

第 1 章 深度学习入门	1	2.2 在 Linux 上安装 TensorFlow	19
1.1 机器学习简介	1	2.3 为 TensorFlow 启用 NVIDIA GPU	20
1.1.1 监督学习	2	2.3.1 第 1 步: 安装 NVIDIA CUDA	20
1.1.2 无监督学习	2	2.3.2 第 2 步: 安装 NVIDIA cuDNN	
1.1.3 强化学习	3	v5.1+	21
1.2 深度学习定义	3	2.3.3 第 3 步: 确定 GPU 卡的 CUDA	
1.2.1 人脑的工作机制	3	计算能力为 3.0+	22
1.2.2 深度学习历史	4	2.3.4 第 4 步: 安装 libcupti-dev 库	22
1.2.3 应用领域	5	2.3.5 第 5 步: 安装 Python	
1.3 神经网络	5	(或 Python 3)	22
1.3.1 生物神经元	5	2.3.6 第 6 步: 安装并升级 PIP	
1.3.2 人工神经元	6	(或 PIP3)	22
1.4 人工神经网络的学习方式	8	2.3.7 第 7 步: 安装 TensorFlow	23
1.4.1 反向传播算法	8	2.4 如何安装 TensorFlow	23
1.4.2 权重优化	8	2.4.1 直接使用 pip 安装	23
1.4.3 随机梯度下降法	9	2.4.2 使用 virtualenv 安装	24
1.5 神经网络架构	10	2.4.3 从源代码安装	26
1.5.1 多层感知器	10	2.5 在 Windows 上安装 TensorFlow	27
1.5.2 DNN 架构	11	2.5.1 在虚拟机上安装 TensorFlow	27
1.5.3 卷积神经网络	12	2.5.2 直接安装到 Windows	27
1.5.4 受限玻尔兹曼机	12	2.6 测试安装是否成功	28
1.6 自编码器	13	2.7 计算图	28
1.7 循环神经网络	14	2.8 为何采用计算图	29
1.8 几种深度学习框架对比	14	2.9 编程模型	30
1.9 小结	16	2.10 数据模型	33
第 2 章 TensorFlow 初探	17	2.10.1 阶	33
2.1 总览	17	2.10.2 形状	33
2.1.1 TensorFlow 1.x 版本特性	18	2.10.3 数据类型	34
2.1.2 使用上的改进	18	2.10.4 变量	36
2.1.3 TensorFlow 安装与入门	19	2.10.5 取回	37

2.10.6 注入	38	4.2 CNN 架构	84
2.11 TensorBoard	38	4.3 构建你的第一个 CNN	86
2.12 实现一个单输入神经元	39	4.4 CNN 表情识别	95
2.13 单输入神经元源代码	43	4.4.1 表情分类器源代码	104
2.14 迁移到 TensorFlow 1.x 版本	43	4.4.2 使用自己的图像测试模型	107
2.14.1 如何用脚本升级	44	4.4.3 源代码	109
2.14.2 局限	47	4.5 小结	111
2.14.3 手动升级代码	47	第 5 章 优化 TensorFlow 自编码器	112
2.14.4 变量	47	5.1 自编码器简介	112
2.14.5 汇总函数	47	5.2 实现一个自编码器	113
2.14.6 简化的数学操作	48	5.3 增强自编码器的鲁棒性	119
2.14.7 其他事项	49	5.4 构建去噪自编码器	120
2.15 小结	49	5.5 卷积自编码器	127
第 3 章 用 TensorFlow 构建前馈神经网络	51	5.5.1 编码器	127
3.1 前馈神经网络介绍	51	5.5.2 解码器	128
3.1.1 前馈和反向传播	52	5.5.3 卷积自编码器源代码	134
3.1.2 权重和偏差	53	5.6 小结	138
3.1.3 传递函数	53	第 6 章 循环神经网络	139
3.2 手写数字分类	54	6.1 RNN 的基本概念	139
3.3 探究 MNIST 数据集	55	6.2 RNN 的工作机制	140
3.4 softmax 分类器	57	6.3 RNN 的展开	140
3.5 TensorFlow 模型的保存和还原	63	6.4 梯度消失问题	141
3.5.1 保存模型	63	6.5 LSTM 网络	142
3.5.2 还原模型	63	6.6 RNN 图像分类器	143
3.5.3 softmax 源代码	65	6.7 双向 RNN	149
3.5.4 softmax 启动器源代码	66	6.8 文本预测	155
3.6 实现一个五层神经网络	67	6.8.1 数据集	156
3.6.1 可视化	69	6.8.2 困惑度	156
3.6.2 五层神经网络源代码	70	6.8.3 PTB 模型	156
3.7 ReLU 分类器	72	6.8.4 运行例程	157
3.8 可视化	73	6.9 小结	158
3.9 dropout 优化	76	第 7 章 GPU 计算	160
3.10 可视化	78	7.1 GPGPU 计算	160
3.11 小结	80	7.2 GPGPU 的历史	161
第 4 章 TensorFlow 与卷积神经网络	82	7.3 CUDA 架构	161
4.1 CNN 简介	82	7.4 GPU 编程模型	162

7.5 TensorFlow 中 GPU 的设置	163	9.2.2 使用重训练的模型	195
7.6 TensorFlow 的 GPU 管理	165	9.3 加速线性代数	197
7.7 GPU 内存管理	168	9.3.1 TensorFlow 的核心优势	197
7.8 在多 GPU 系统上分配单个 GPU	168	9.3.2 加速线性代数的准时编译	197
7.9 使用多个 GPU	170	9.4 TensorFlow 和 Keras	202
7.10 小结	171	9.4.1 Keras 简介	202
第 8 章 TensorFlow 高级编程	172	9.4.2 拥有 Keras 的好处	203
8.1 Keras 简介	172	9.4.3 视频问答系统	203
8.2 构建深度学习模型	174	9.5 Android 上的深度学习	209
8.3 影评的情感分类	175	9.5.1 TensorFlow 演示程序	209
8.4 添加一个卷积层	179	9.5.2 Android 入门	211
8.5 Pretty Tensor	181	9.6 小结	214
8.6 数字分类器	182	第 10 章 强化学习	215
8.7 TFLearn	187	10.1 强化学习基本概念	216
8.8 泰坦尼克号幸存者预测器	188	10.2 Q-learning 算法	217
8.9 小结	191	10.3 OpenAI Gym 框架简介	218
第 9 章 TensorFlow 高级多媒体编程	193	10.4 FrozenLake-v0 实现问题	220
9.1 多媒体分析简介	193	10.5 使用 TensorFlow 实现 Q-learning	223
9.2 基于深度学习的大型对象检测	193	10.6 小结	227
9.2.1 瓶颈层	195		

第1章

深度学习入门



本章会讨论深度学习的一些基本概念及其相关架构，这些内容在后续章节中都会有所涉及。首先简要介绍机器学习的定义。运用机器学习技术可以实现对大型数据集的分析，自动抽取信息并对后续的新数据进行预测。接着介绍深度学习的概念。深度学习是机器学习的一个分支，旨在使用一系列算法对高度抽象的数据进行建模。

随后介绍深度学习架构，也就是所谓的**深度神经网络（Deep Neural Networks, DNN）**。这种神经网络与普通的单隐藏层神经网络的不同主要在于“深度”。“深度”指的是节点层的数量。也就是说，深度神经网络拥有多个层，在模式识别的过程中，数据需要依次通过这些层。本章用一个表格总结了各种神经网络模型，大部分深度学习算法都是由这些神经网络模型发展而来的。

本章最后会针对多个特性简要了解并对比几种深度学习框架，比如框架使用的语言、多GPU支持及可用性等。

本章包含以下主题：

- 机器学习简介
- 深度学习定义
- 神经网络
- 人工神经网络的学习方式
- 神经网络架构
- DNN架构
- 几种深度学习框架的对比

1.1 机器学习简介

机器学习是计算机科学中的一个研究领域，其目标是识别并实现一些系统和算法，从而使计算机基于给定输入中的样例进行学习。机器学习的难点在于，如何使计算机通过学习自动识别复杂的模式，并尽可能智能地做出决策。完整的学习过程需要以下数据集。

- **训练集**：知识库，用于训练机器学习算法。在这一阶段，机器学习模型参数（超参数）可以根据随后获得的性能进行调整。
- **测试集**：仅用于评价模型对未知数据的预测性能。

学习理论采用的数学工具衍生自概率论、信息论等学科，可以实现对模型性能的对比。

机器学习的范式主要有三种：

- 监督学习
- 无监督学习
- 强化学习

下面简要叙述这三种学习范式。

1.1.1 监督学习

监督学习是一种相对简单、成熟的自动学习任务。它基于一系列预分类的样例，也就是说，预先知道作为输入的每个样例应该属于哪个类。在这种情况下，学习的关键问题就是“泛化”。对一个样本（通常比较小）进行分析后，系统应当能够生成一个对所有可能输入都可用的模型。

数据集包含有标签的数据，即“对象”及其对应的“类”。这个有标签属性的集合便组成了训练集。

大部分监督学习算法有一个共同的特点：定义某种“损失函数”或“代价函数”，对训练集进行处理时使其最小化。该函数代表系统的实际输出相对于正确输出的错误率，因为训练集给定的输出肯定是正确输出。

系统接下来会改变其内部的一些可调的参数，即权重，以最小化这个误差函数。随后，要评价模型的性能。给出另一组有标签的样例（测试集），检测被正确分类的样例和被错误分类的样例的百分比。

监督学习除了应用于分类场景，还可用于学习数值型预测函数。这种任务称为**回归**。在回归问题中，训练集是由对象及其对应的数值组成的二元组。许多监督学习算法已经实际应用于分类和回归问题。这些算法均可视为一套描述分类器或预测器的规则，其中包括决策树、决策法则、神经网络和贝叶斯网络。

1.1.2 无监督学习

无监督学习与监督学习相反，在训练过程中提供给系统的输入是没有标签的。这种学习范式非常重要，因为人脑处理的无监督学习任务要比监督学习任务多得多。

在这种情况下，学习模型中的唯一对象便是观察到的输入数据。一般假设这些输入数据为相互独立的样本，并服从某种概率分布。

无监督学习算法常用于处理聚类问题，即给定一系列对象，我们希望理解并展示它们之间的关系。一种标准的做法是为每两个对象定义一种相似度，据此找出簇的划分，使得簇内对象相似度较大，簇间对象相似度相对较小。

1.1.3 强化学习

强化学习是一种人工智能方法，强调利用系统与其环境之间的交互进行学习。通过强化学习，系统根据从环境获得的反馈动态调整其参数，调参的结果又进一步作为反馈指导决策。例如，用前面棋步的结果改进性能的国际象棋程序就是一个强化学习系统。目前，针对强化学习的研究涉及众多学科，涵盖了遗传算法、神经网络、心理学和控制工程等各种领域。

图1-1总结了上述三种学习范式，并列出了与之相关的问题。



图1-1 学习范式及其相关问题

1.2 深度学习定义

深度学习是机器学习中的一个研究方向，它基于一种特殊的学习机制。其特点是建立一个多层学习模型，深层级将浅层级的输出作为输入，将数据层层转化，使之越来越抽象。这种分层学习思想模拟的是人脑接受外界刺激时处理信息和学习的方式。

根据假设，每一个学习层对应大脑皮层的一个不同区域。

1.2.1 人脑的工作机制

负责解决图像识别问题的视觉皮质由一系列具有层级结构的扇区组成，每个区域从与之连接

的其他扇区接收信号流作为输入表示。

这种层级结构中的每一层代表不同程度的抽象，层级越高，抽象程度越大。大脑接收到一个输入图像时，会分成几个阶段对其进行处理，例如检测边缘或感知形状（任务由简单逐渐变得复杂）。

随着大脑根据经验一点一点地学习、激活新的神经元，抽象层会随着输入信息变化。深度学习架构就模仿了人脑的这一过程。

在图像分类任务中，这种层级结构叙述如下：一层层的处理块逐渐提取输入图像的特征，每一个块都会继续处理已被前面的块预处理过的数据，提取的特征也越来越抽象。这样，数据就被表示为一种层级结构，这也就是深度学习系统的基础。

更具体地说，可视化的层级建立过程如下所示。

- 第1层：系统开始识别明/暗像素
- 第2层：系统识别边缘和形状
- 第3层：系统学习到更为复杂的形状和物体
- 第4层：系统学习人脸由哪些物体定义

图1-2是该过程的可视化表示。

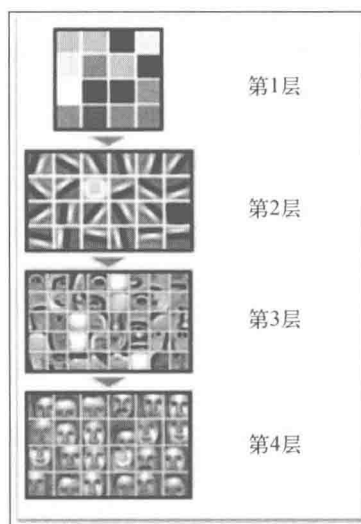


图1-2 人脸识别问题中的深度学习系统

1.2.2 深度学习历史

深度学习的发展与人工智能，尤其是神经网络的研究齐头并进。深度学习的研究始于20世纪