
机器人与自动驾驶中的感知原理、算法和 实践

Release 1.0.0

zhiyuan.zhang

Aug 29, 2023

前言	1
1 安装	3
2 符号	5
3 引言	9
4 预备知识	11
4.1 数据操作	12
4.1.1 入门	12
4.2 数据预处理	12
4.2.1 读取数据集	12
5 C++	15
6 软件工具	17

前言

几年前，在大公司和初创公司中，并没有大量的深度学习科学家开发智能产品和服务。我们中年轻人（作者）进入这个领域时，机器学习并没有在报纸上获得头条新闻。我们的父母根本不知道什么是机器学习，更不用说为什么我们可能更喜欢机器学习，而不是从事医学或法律职业。机器学习是一门具有前瞻性的学科，在现实世界的应用范围很窄。而那些应用，例如语音识别和计算机视觉，需要大量的领域知识，以至于它们通常被认为是完全独立的领域，而机器学习对这些领域来说只是一个小组件。因此，神经网络——我们在本书中关注的深度学习模型的前身，被认为是过时的工具。

就在过去的五年里，深度学习给世界带来了惊喜，推动了计算机视觉、自然语言处理、自动语音识别、强化学习和统计建模等领域的快速发展。有了这些进步，我们现在可以制造比以往任何时候都更自主的汽车（不过可能没有一些公司试图让大家相信的那么自主），可以自动起草普通邮件的智能回复系统，帮助人们从令人压抑的大收件箱中解放出来。在围棋等棋类游戏中，软件超越了世界上最优秀的人，这曾被认为是几十年后的事。这些工具已经对工业和社会产生了越来越广泛的影响，改变了电影的制作方式、疾病的诊断方式，并在基础科学中扮演着越来越重要的角色——从天体物理学到生物学。

关于本书

这本书代表了我们的尝试——让深度学习可平易近人，教会人们概念、背景和代码。

一种结合了代码、数学和HTML的媒介

任何一种计算技术要想发挥其全部影响力，都必须得到充分的理解、充分的文档记录，并得到成熟的、维护良好的工具的支持。关键思想应该被清楚地提炼出来，尽可能减少需要让新的从业者跟上时代的入门时间。成熟的库应该自动化常见的任务，示例代码应该使从业者可以轻松地修改、应用和扩展常见的应用程序，以满足他们的需求。以动态网页应用为例。尽管许多公司，如亚马逊，在20世纪90年代开发了成功的数据库驱动网页应用程序。但在过去的10年里，这项技术在帮助创造性企业家方面的潜力已经得到了更大程度的发挥，部分原因是开发了功能强大、文档完整的框架。

测试深度学习的潜力带来了独特的挑战，因为任何一个应用都会将不同的学科结合在一起。应用深度学习需要同时了解 (1) 以特定方式提出问题的动机；(2) 给定建模方法的数学；(3) 将模型拟合数据的优化算法；(4) 能够有效训练模型、克服数值计算缺陷并最大限度地利用现有硬件的工程方法。同时教授表述问题所需的批判性思维技能、解决问题所需的数学知识，以及实现这些解决方案所需的软件工具，这是一个巨大的挑战。

我们需要配置一个环境来运行 Python、Jupyter Notebook、相关库以及运行本书所需的代码，以快速入门并获得动手学习经验。

1.1 安装 Miniconda

最简单的方法就是安装依赖Python 3.x的Miniconda¹。如果已安装conda，则可以跳过以下步骤。访问Miniconda网站，根据Python3.x版本确定适合的版本。

如果我们使用macOS，假设Python版本是3.9（我们的测试版本），将下载名称包含字符串“MacOSX”的bash脚本，并执行以下操作：

```
# 以Intel处理器为例，文件名可能会更改
sh Miniconda3-py39_4.12.0-MacOSX-x86_64.sh -b
```

如果我们使用Linux，假设Python版本是3.9（我们的测试版本），将下载名称包含字符串“Linux”的bash脚本，并执行以下操作：

```
# 文件名可能会更改
sh Miniconda3-py39_4.12.0-Linux-x86_64.sh -b
```

接下来，初始化终端Shell，以便我们可以直接运行conda。

¹ <https://conda.io/en/latest/miniconda.html>

```
~/miniconda3/bin/conda init
```

现在关闭并重新打开当前的shell。并使用下面的命令创建一个新的环境：

```
conda create --name d2l python=3.9 -y
```

现在激活 d2l 环境：

```
conda activate d2l
```

1.2 安装深度学习框架和d2l软件包

在安装深度学习框架之前，请先检查计算机上是否有可用的GPU。例如可以查看计算机是否装有NVIDIA GPU并已安装CUDA²。如果机器没有任何GPU，没有必要担心，因为CPU在前几章完全够用。但是，如果想流畅地学习全部章节，请提早获取GPU并且安装深度学习框架的GPU版本。

² <https://developer.nvidia.com/cuda-downloads>

本书中使用的符号概述如下。

2.1 数字

- x : 标量
- \mathbf{x} : 向量
- \mathbf{X} : 矩阵
- χ : 张量
- \mathbf{I} : 单位矩阵
- $x_i, [\mathbf{x}]_i$: 向量 \mathbf{x} 第 i 个元素
- $x_{ij}, [\mathbf{X}]_{ij}$: 矩阵 \mathbf{X} 第 i 行第 j 列的元素

2.2 集合论

- \mathcal{X} : 集合
- \mathbb{Z} : 整数集合
- \mathbb{R} : 实数集合
- \mathbb{R}^n : n 维实数向量集合
- $\mathbb{R}^{a \times b}$: 包含 a 行和 b 列的实数矩阵集合
- $\mathcal{A} \cup \mathcal{B}$: 集合 \mathcal{A} 和 \mathcal{B} 的并集
- $\mathcal{A} \cap \mathcal{B}$: 集合 \mathcal{A} 和 \mathcal{B} 的交集
- $\mathcal{A} \setminus \mathcal{B}$: 集合 \mathcal{A} 与集合 \mathcal{B} 相减, \mathcal{B} 关于 \mathcal{A} 的相对补集

2.3 函数和运算符

- $f(\cdot)$: 函数
- $\log(\cdot)$: 自然对数
- $\exp(\cdot)$: 指数函数
- $\mathbf{1}_{\mathcal{X}}$: 指示函数
- $(\cdot)^{\top}$: 向量或矩阵的转置
- \mathbf{X}^{-1} : 矩阵的逆
- \odot : 按元素相乘
- $[\cdot, \cdot]$: 连结
- $|\mathcal{X}|$: 集合的基数
- $\|\cdot\|_p$: L_p 正则
- $\|\cdot\|$: L_2 正则
- $\langle \mathbf{x}, \mathbf{y} \rangle$: 向量 \mathbf{x} 和 \mathbf{y} 的点积
- \sum : 连加
- \prod : 连乘
- $\stackrel{\text{def}}{=}$: 定义

2.4 微积分

- $\frac{dy}{dx}$: y 关于 x 的导数
- $\frac{\partial y}{\partial x}$: y 关于 x 的偏导数
- $\nabla_{\mathbf{x}} y$: y 关于 \mathbf{x} 的梯度
- $\int_a^b f(x) dx$: f 在 a 到 b 区间上关于 x 的定积分
- $\int f(x) dx$: f 关于 x 的不定积分

2.5 概率与信息论

- $P(\cdot)$: 概率分布
- $z \sim P$: 随机变量 z 具有概率分布 P
- $P(X | Y)$: $X | Y$ 的条件概率
- $p(x)$: 概率密度函数
- $E_x[f(x)]$: 函数 f 对 x 的数学期望
- $X \perp Y$: 随机变量 X 和 Y 是独立的
- $X \perp Y | Z$: 随机变量 X 和 Y 在给定随机变量 Z 的条件下是独立的
- $\text{Var}(X)$: 随机变量 X 的方差
- σ_X : 随机变量 X 的标准差
- $\text{Cov}(X, Y)$: 随机变量 X 和 Y 的协方差
- $\rho(X, Y)$: 随机变量 X 和 Y 的相关性
- $H(X)$: 随机变量 X 的熵
- $D_{\text{KL}}(P \| Q)$: P 和 Q 的KL-散度

2.6 复杂度

- \mathcal{O} : 大O标记

引言

时至今日，人们常用的计算机程序几乎都是软件开发人员从零编写的。比如，现在开发人员要编写一个程序来管理网上商城。经过思考，开发人员可能提出如下一个解决方案：首先，用户通过Web浏览器（或移动应用程序）与应用程序进行交互；紧接着，应用程序与数据库引擎进行交互，以保存交易历史记录并跟踪每个用户的动态；其中，这个应用程序的核心——“业务逻辑”，详细说明了应用程序在各种情况下进行的操作。

为了完善业务逻辑，开发人员必须细致地考虑应用程序所有可能遇到的边界情况，并为这些边界情况设计合适的规则。当买家单击将商品添加到购物车时，应用程序会向购物车数据库表中添加一个条目，将该用户ID与商品ID关联起来。虽然一次编写出完美应用程序的可能性微乎其微，但在大多数情况下，开发人员可以从上述的业务逻辑出发，编写出符合业务逻辑的应用程序，并不断测试直到满足用户的需求。根据业务逻辑设计自动化系统，驱动正常运行的产品和系统，是一个人类认知上的非凡壮举。

幸运的是，对日益壮大的机器学习科学家群体来说，实现很多任务的自动化并不再屈从于人类所能考虑到的逻辑。想象一下，假如开发人员要试图解决以下问题之一：

- 编写一个应用程序，接受地理信息、卫星图像和一些历史天气信息，并预测明天的天气；
- 编写一个应用程序，接受自然文本表示的问题，并正确回答该问题；
- 编写一个应用程序，接受一张图像，识别出该图像所包含的人，并在每个人周围绘制轮廓；
- 编写一个应用程序，向用户推荐他们可能喜欢，但在自然浏览过程中不太可能遇到的产品。

在这些情况下，即使是顶级程序员也无法提出完美的解决方案，原因可能各不相同。有时任务可能遵循一种随着时间推移而变化的模式，我们需要程序来自动调整。有时任务内的关系可能太复杂（比如像素和抽象类别之间的关系），需要数千或数百万次的计算。即使人类的眼睛能毫不费力地完成这些难以提出完美解决方案的任务，这其中的计算也超出了人类意识理解范畴。机器学习（machine learning, ML）是一类强大的可以从经验中学习的技术。通常采用观测数据或与环境交互的形式，机器学习算法会积累更多的经验，其性能

也会逐步提高。相反，对于刚刚所说的电子商务平台，如果它一直执行相同的业务逻辑，无论积累多少经验，都不会自动提高，除非开发人员认识到问题并更新软件。本书将带读者开启机器学习之旅，并特别关注深度学习（deep learning, DL）的基础知识。深度学习是一套强大的技术，它可以推动计算机视觉、自然语言处理、医疗保健和基因组学等不同领域的创新。

3.1 日常生活中的机器学习

机器学习应用在日常生活中的方方面面。现在，假设本书的作者们一起驱车去咖啡店。阿斯顿拿起一部iPhone，对它说道：“Hey Siri!” 手机的语音识别系统就被唤醒了。接着，李沐对Siri说道：“去星巴克咖啡店。” 语音识别系统就自动触发语音转文字功能，并启动地图应用程序，地图应用程序在启动后筛选了若干条路线，每条路线都显示了预计的通行时间……由此可见，机器学习渗透在生活中的方方面面，在短短几秒钟的时间里，人们与智能手机的日常互动就可以涉及几种机器学习模型。

预备知识

要学习深度学习，首先需要先掌握一些基本技能。所有机器学习方法都涉及从数据中提取信息。因此，我们先学习一些关于数据的实用技能，包括存储、操作和预处理数据。

机器学习通常需要处理大型数据集。我们可以将某些数据集视为一个表，其中表的行对应样本，列对应属性。线性代数为人们提供了一些用来处理表格数据的方法。我们不会太深究细节，而是将重点放在矩阵运算的基本原理及其实现上。

深度学习是关于优化的学习。对于一个带有参数的模型，我们想要找到其中能拟合数据的最好模型。在算法的每个步骤中，决定以何种方式调整参数需要一点微积分知识。本章将简要介绍这些知识。幸运的是，`autograd`包会自动计算微分，本章也将介绍它。

机器学习还涉及如何做出预测：给定观察到的信息，某些未知属性可能的值是多少？要在不确定的情况下进行严格的推断，我们需要借用概率语言。

最后，官方文档提供了本书之外的大量描述和示例。在本章的结尾，我们将展示如何在官方文档中查找所需信息。

本书对读者数学基础无过分要求，只要可以正确理解深度学习所需的数学知识即可。但这并不意味着本书中不涉及数学方面的内容，本章会快速介绍一些基本且常用的数学知识，以便读者能够理解书中的大部分数学内容。如果读者想要深入理解全部数学内容，可以进一步学习本书数学附录中给出的数学基础知识。

4.1 数据操作

为了能够完成各种数据操作，我们需要某种方法来存储和操作数据。通常，我们需要做两件重要的事：(1) 获取数据；(2) 将数据读入计算机后对其进行处理。如果没有某种方法来存储数据，那么获取数据是没有意义的。

首先，我们介绍 n 维数组，也称为张量（tensor）。使用过Python中NumPy计算包的读者会对本部分很熟悉。无论使用哪个深度学习框架，它的张量类（在MXNet中为ndarray，在PyTorch和TensorFlow中为Tensor）都与Numpy的ndarray类似。但深度学习框架又比Numpy的ndarray多一些重要功能：首先，GPU很好地支持加速计算，而NumPy仅支持CPU计算；其次，张量类支持自动微分。这些功能使得张量类更适合深度学习。如果没有特殊说明，本书中所说的张量均指的是张量类的实例。

4.1.1 入门

本节的目标是帮助读者了解并运行一些在阅读本书的过程中会用到的基本数值计算工具。如果你很难理解一些数学概念或库函数，请不要担心。后面的章节将通过一些实际的例子来回顾这些内容。如果你已经具有相关经验，想要深入学习数学内容，可以跳过本节。

首先，我们从MXNet导入np（numpy）模块和npx（numpy_extension）模块。np模块包含NumPy支持的函数；而npx模块包含一组扩展函数，用来在类似NumPy的环境中实现深度学习开发。当使用张量时，几乎总是会调用set_np函数，这是为了兼容MXNet的其他张量处理组件。:end_tab:

4.2 数据预处理

为了能用深度学习来解决现实世界的问题，我们经常从预处理原始数据开始，而不是从那些准备好的张量格式数据开始。在Python中常用的数据分析工具中，我们通常使用pandas软件包。像庞大的Python生态系统中的许多其他扩展包一样，pandas可以与张量兼容。本节我们将简要介绍使用pandas预处理原始数据，并将原始数据转换为张量格式的步骤。后面的章节将介绍更多的数据预处理技术。

4.2.1 读取数据集

举一个例子，我们首先创建一个人工数据集，并存储在CSV（逗号分隔值）文件../data/house_tiny.csv中。以其他格式存储的数据也可以通过类似的方式进行处理。下面我们将数据集按行写入CSV文件中。

```
#@tab all
import os

os.makedirs(os.path.join '..', 'data'), exist_ok=True
data_file = os.path.join '..', 'data', 'house_tiny.csv'
with open(data_file, 'w') as f:
    f.write('NumRooms,Alley,Price\n') # 列名
```

(continues on next page)

(continued from previous page)

```
f.write('NA,Pave,127500\n') # 每行表示一个数据样本
f.write('2,NA,106000\n')
f.write('4,NA,178100\n')
f.write('NA,NA,140000\n')
```

要从创建的CSV文件中加载原始数据集，我们导入pandas包并调用read_csv函数。该数据集有四行三列。其中每行描述了房间数量（“NumRooms”）、巷子类型（“Alley”）和房屋价格（“Price”）。

```
#@tab all
# 如果没有安装pandas，只需取消对以下行的注释来安装pandas
# !pip install pandas
import pandas as pd

data = pd.read_csv(data_file)
print(data)
```


5

C++

软件工具
