深度学习系列
Deep Learning

掌握深度学习数学原理、编程实战经验
轻松构建复杂实际项目的深度学习方案

Apress®

TensorFlow 深度学习

数学原理与Python实战进阶

Pro Deep Learning with TensorFlow

A Mathematical Approach to Advanced Artificial Intelligence in Python

塔努·帕塔纳雅克（Santanu Pattanayak） 著
魏国强 倪晨杰 李杨 厉高远 罗佳程 译

机械工业出版社
CHINA MACHINE PRESS

深度学习系列

TensorFlow 深度学习：数学原理与 Python 实战进阶

[印] 桑塔努·帕塔纳雅克（Santanu Pattanayak） 著
魏国强 倪晨杰 李杨 厉高远 罗佳程 译

机械工业出版社

本书重点在帮你掌握深度学习所要求的数学原理和编程实战经验，使你能快速使用 TensorFlow 轻松部署产品中的深度学习解决方案，并形成开发深度学习架构和解决方案时所需的数学理解和直觉。

本书提供了丰富的理论和实战动手经验，使你可以从零开始掌握深度学习，并能快速部署有价值的深度学习解决方案。本书重点讲解了与多个行业相关的深度学习实践方面的专业知识。通过这些实战经验，你将能够使用原型来构建新的深度学习应用程序。

图书在版编目（CIP）数据

TensorFlow 深度学习：数学原理与 Python 实战进阶/（印）桑塔努·帕塔纳雅克（Santanu Pattanayak）著；魏国强等译. —北京：机械工业出版社，2020.2

（深度学习系列）

书名原文：Pro Deep Learning with TensorFlow：A Mathematical Approach to Advanced Artificial Intelligence in Python

ISBN 978-7-111-64584-9

Ⅰ.①T… Ⅱ.①桑…②魏… Ⅲ.①人工智能 - 算法 Ⅳ.①TP18

中国版本图书馆 CIP 数据核字（2020）第 017166 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：林 桢 责任编辑：闾洪庆 林 桢

责任校对：肖 琳 封面设计：鞠 杨

责任印制：李 昂

唐山三艺印务有限公司印刷

2020 年 4 月第 1 版第 1 次印刷

184mm × 240mm · 20 印张 · 457 千字

标准书号：ISBN 978-7-111-64584-9

定价：99.00 元

电话服务

客服电话：010 - 88361066

010 - 88379833

010 - 68326294

封底无防伪标均为盗版

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

原书前言

本书是使用 TensorFlow 实现深度学习的数学原理和编程实战指南。深度学习是机器学习的一个分支，你可以在其中根据概念的层次结构为世界建模。这种学习模式类似于人脑的学习方式，它允许计算机对复杂的概念进行建模，而这些概念在其他传统的建模方法中通常没有被注意到。因此，在现代计算范式中，深度学习在对复杂的实际问题进行建模时扮演着至关重要的角色，尤其是通过利用如今大量可用的非结构化数据。

由于深度学习模型涉及内容的复杂性，很多时候使用它的人们都将其视为黑盒子。但是，为了从机器学习这一分支中获得最大的收益，需要通过研究与之相关的科学和数学来发现隐藏的奥秘。在本书中，我们从数学和科学的角度都非常谨慎地解释了与深度学习相关的概念和技术。此外，第 1 章专注于构建轻松理解深度学习概念所需的数学基础。TensorFlow 被选为深度学习软件包，是因为它在用于研究目的上的灵活性以及易用性。选择 TensorFlow 的另一个原因是它能够使用其服务功能轻松地在实时生产环境中加载模型。

总而言之，本书提供了大量实用的实战知识，因此你可以从零开始学习深度学习并部署有价值的深度学习解决方案。本书将能帮助你迅速上手使用 TensorFlow，并优化了实际应用中不同的深度学习架构。本书重点关注与多个行业相关的深度学习的实践方面的内容。你将能够使用演示的原型来构建新的深度学习应用程序。本书中的代码以 iPython 笔记本和脚本的形式提供，让你可以尝试示例并以有趣的方式扩展它们。阅读学习本书后，你将具备数学基础和专业知识，可以从事该领域的研究并回馈社区。

读者对象

- 本书面向正在研究深度学习解决方案以解决复杂业务问题的数据科学家和机器学习专业人员。
- 本书适用于通过 TensorFlow 开发深度学习解决方案的软件开发人员。
- 本书也适合渴望不断学习的师生和人工智能爱好者。

本书内容

本书涵盖的章节如下：

第 1 章数学基础，在该章中，详细讨论线性代数、概率、微积分、优化和机器学习公式的相关数学概念，从而为深度学习奠定数学基础。在讲解了各种概念后，重点是关注它们在机器学习和深度学习领域中的用法。

第 2 章深度学习概念和 TensorFlow 介绍，该章介绍深度学习的世界，并讨论了其多年来的发展。详细讲解了神经网络的关键组成部分以及几种学习方法，例如感知器学习规则和反向传播算法。此外，该章还介绍了 TensorFlow 编码的范例，以便你在转入 TensorFlow 并涉及更多的实践前熟悉基本语法。

第 3 章卷积神经网络，该章讨论用于图像处理的卷积神经网络。图像处理是计算机视觉领域的一个重要研究课题，在将卷积神经网络用于对象识别和检测、对象分类、定位和分割等领域后，性能得到了极大的提升。该章首先详细说明卷积的操作，然后继续讲解卷积神经网络的工作原理。重点介绍了卷积神经网络的组成部分，从而为你提供了以有趣的方式进行实验和扩展其网络所需的工具。此外，将详细阐述通过卷积和池化层的反向传播，以帮助你全面了解卷积神经网络的训练过程。该章还介绍了平移同变性和平移不变性的属性，它们对于卷积神经网络的成功至关重要。

第 4 章基于循环神经网络的自然语言处理，该章讲解使用深度学习进行自然语言处理的内容。首先从用于文本处理的不同向量空间模型开始，之后是词到向量的嵌入模型，例如连续词袋方法和 Skip - gram，然后转到涉及循环神经网络、LSTM、门控循环单元和双向循环神经网络的更高级的主题。该章详细介绍了语言建模，以帮助你在涉及该网络的实际问题中利用这些网络。此外，还详细讨论了循环神经网络和 LSTM 情况下的反向传播机制以及梯度消失问题。

第 5 章用受限玻尔兹曼机和自编码器进行无监督学习，在该章中，你将学习使用受限玻尔兹曼机和自编码器的深度学习中的无监督方法。另外，该章还将讲解贝叶斯推断和 MCMC 方法，例如 Metropolis 算法和吉布斯采样，因为受限玻尔兹曼机训练过程需要一些采样知识。此外，该章还将讨论对比散度，这是吉布斯采样的定制版本，可以对受限玻尔兹曼机进行实际训练。我们将进一步讨论受限玻尔兹曼机如何在推荐系统中用于协作过滤，以及如何在深度置信网络的无监督预训练中使用。

该章后半部分介绍了各种自编码器，例如稀疏自编码器、去噪自编码器等。此外，你还会学习如何将从自编码器中学到的内部特征用于降维以及监督学习。最后，该章简要介绍了数据预处理技术，例如 PCA 白化和 ZCA 白化。

第 6 章高级神经网络，在该章中，你将学习一些高级神经网络，例如全卷积神经网络、U - Net、R - CNN、Fast R - CNN、Faster R - CNN 等，处理图像的语义分割、对象检测和定位。该章还将讲解传统的图像分割方法，以便可以适当地结合两个方面的优点。在该章的后半部分，你将学习生成式对抗网络，这是一种用于生成合成数据（如给定分布所生成的数据）的生成模型的新模式。生成式对抗网络在多个领域具有用途和潜力，例如在图像生成、图像修复、抽象推理、语义分割、视频生成、域间样式迁移以及文本到图像生成应用程序等领域。

总而言之，你可以从本书中学到如下主要知识：

- 理解使用 TensorFlow 的全栈深度学习，并为深度学习奠定坚实的数学基础。
- 使用 TensorFlow 在产品中部署复杂的深度学习解决方案。
- 进行深度学习研究并使用 TensorFlow 进行实验。

译者简介

魏国强

中国科学技术大学与微软亚洲研究院联合培养博士在读，研究方向为姿态估计及其应用等。熟悉 TensorFlow 等深度学习框架。曾参与 PyTorch 官方文档汉化等翻译工作。

倪晨杰

毕业于多伦多大学，本科阶段专修计算机科学，主要研究方向是人工智能和图像处理。了解主流深度学习框架（如 PyTorch 和 TensorFlow）的原理以及应用。擅长将人工智能与云计算平台相结合，使机器学习算法的开发变得更加灵活、高效。

李杨

目前在初创公司做软件开发工作，硕士研究生，毕业于天津大学，主要从事视觉人工智能系统开发。研究兴趣广泛，希望和大家一起探讨新技术新知识。联系方式：lwkj. liyang@gmail. com。

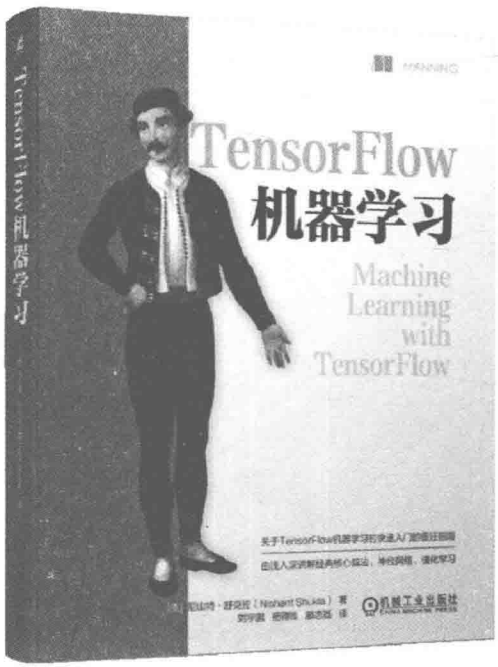
厉高远

一知智能服务端工程师。喜欢天马行空的 idea，喜欢新奇锐意的产品，喜欢探寻未知。

罗佳程

毕业于云南大学，获工学硕士学位。长期从事推荐系统研究，活跃于各大技术社区。

——推荐阅读——



TensorFlow 机器学习

［美］尼山特·舒克拉（Nishant Shukla）著　刘宇鹏　杨锦锋　滕志扬　译　定价：69 元

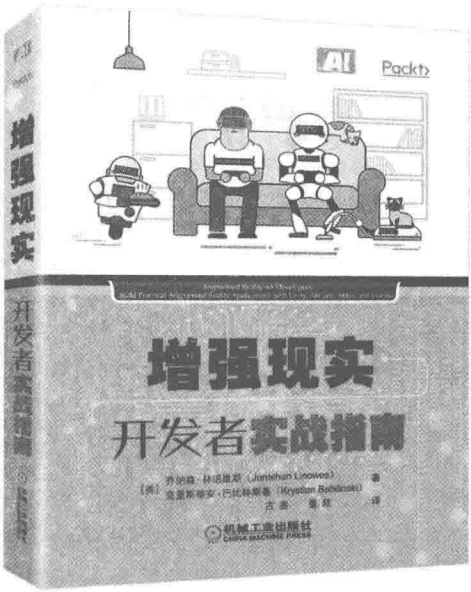
关于 TensorFlow 机器学习的快速入门的极好指南。由浅入深讲解经典核心算法、神经网络、强化学习。

为你提供了机器学习概念的坚实基础，以及使用 Python 编写 TensorFlow 的实战经验。

本书由浅入深地对 TensorFlow 进行了介绍，并对 TensorFlow 的本质、核心学习算法（线性回归、分类、聚类、隐马尔可夫模型）和神经网络的类型（自编码器、强化学习、卷积神经网络和循环神经网络）都进行了详细介绍，同时配以代码实现。

你将通过经典的预测、分类和聚类算法等快速学习掌握基础知识。然后，继续学习具有深度价值的内容：探索深度学习的概念，例如自动编码器、递归神经网络和强化学习等。通过本书，你将会准备好将 TensorFlow 用于自己的机器学习和深度学习应用程序中。

本书可作为人工智能、机器学习、深度学习相关行业的从业者和爱好者的重要参考书。



增强现实开发者实战指南

阿里、微软、百度及学界专家联合推荐。

随着几年的蛰伏，即将到来的5G技术，将极大促进增强现实、虚拟现实（AR/VR）行业的突破性发展，学习增强现实开发正当时。

作为一本适合AR开发者的实战案头书，采用逐步教学的实战方式详解如何使用Unity 3D、Vuforia、ARToolkit、HoloLens、Apple ARKit和Google ARCore等主流开发工具。

助你快速掌握并在移动智能设备和可穿戴设备上构建激动人心的实用AR应用程序。

本书适合想要在各平台上开发AR项目的开发人员、设计人员等从业者，AR技术的研究者、相关专业师生，以及对AR技术感兴趣的人员阅读。

实感交互：人工智能下的人机交互技术

人工智能赋能人机交互技术，智能+交互，深入探讨解读人工智能下的人机交互技术。

分析基于触摸、手势、语音和视觉等自然人机交互领域的技术、应用和未来趋势。

- 有关触控技术的明确指导，包括优点、局限性和未来的趋势。
- 基于语音交互的语音输入、处理和识别技术的原理和应用案例讲解。
- 新兴的基于视觉感知技术和手势、身体、面部、眼球追踪交互的详解说明。
- 讨论多模式自然用户交互方案，直观地将触摸、语音和视觉结合在一起，实现真实感互动。
- 审视实现真正3D沉浸式显示和交互的要求和技术现状。

目　录

第1章
数学基础

深度学习是机器学习的一个分支，它由多层人工神经元相互堆叠组成，用于识别输入数据的复杂特征并解决现实中的实际问题。它可以用于监督学习和无监督学习任务。深度学习目前应用于计算机视觉、视频分析、模式识别、异常检测、文本处理、情感分析以及推荐系统等领域。此外，它还广泛应用于机器人、自动驾驶的汽车以及人工智能系统。

数学是任何机器学习算法的核心。对核心数学概念的深入理解能够帮助人们为特定的机器学习问题选择正确的算法，并同时牢记最终目标。此外，它还使人们能够更好地调整机器学习/深度学习模型，并理解算法未尽如人意的可能原因。作为机器学习的一个分支，深度学习相较于其他机器学习任务而言需要同样，甚至更多的专业性数学知识。数学是一门非常广泛的学科，但其中只有一些具体的主题需要机器学习或深度学习专业人士/爱好者注意，这为的是从机器学习、深度学习和人工智能这些奇妙的领域中获得最大收益。图 1-1 所示为数学的不同分支，以及它们在机器学习和深度学习领域中的重要性。本章将讨论以下每个分支的相关概念：

- 线性代数
- 概率和统计
- 微积分
- 机器学习算法的设计与优化

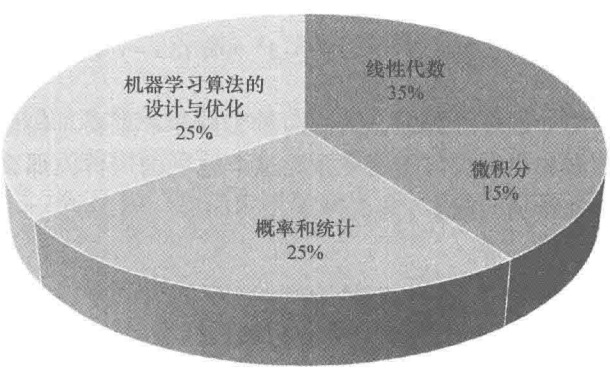


图 1-1　数学对于机器学习和数据科学的重要性

■ 注意：已经熟悉这些话题的读者可以选择跳过本章或对内容进行大概浏览。

1.1 线性代数

线性代数是数学的一个分支，它是关于向量以及它从一个向量空间到另一个向量空间的变换。由于在机器学习和深度学习中我们通常处理多维数据并对其进行运算操作，所以线性代数在几乎所有机器学习和深度学习算法中都起着至关重要的作用。图 1-2 所示为一个三维向量空间，其中 v_1、v_2 和 v_3 是向量，P 是三维向量空间内的二维平面。

1.1.1 向量

连续或离散的数组称为向量，由向量组成的空间称为向量空间。向量空间的维度可以是有限的，也可以是无限的，但大多数机器学习或数据科学问题都处理固定长度的向量，例如，汽车在 x 和 y 方向分别以速度 v_x 和 v_y 在平面内移动（见图 1-3）。

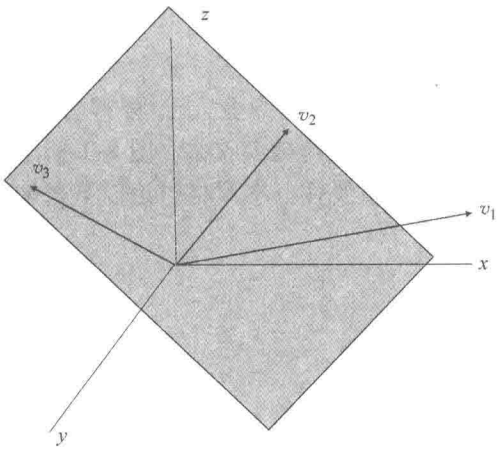


图 1-2 三维向量空间和一个二维平面

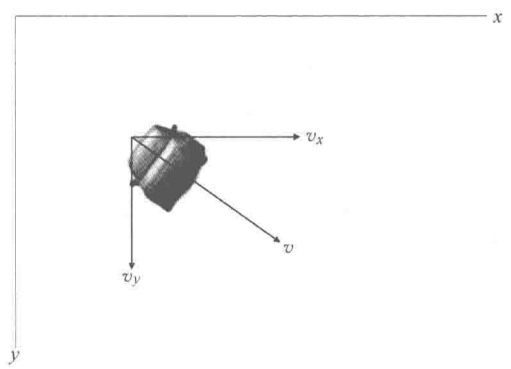


图 1-3 小车在 $x-y$ 坐标平面内以 v_x 和 v_y 的速度移动

在机器学习中，我们处理多维数据，因此向量变得非常重要。假设我们试图根据房屋面积、卧室数量、浴室数量和当地人口密度来预测某个地区的房价，那么所有这些特征构成了这个房价预测问题的输入特征向量。

1.1.2 标量

一维向量即是标量。正如我们在高中所学到的，标量是一个数量，只有大小，没有方向。这是因为，由于它只有一个可以移动的方向，所以它的方向并不重要，我们只关心它的大小。例如，儿童的身高、水果的重量等。

1.1.3 矩阵

矩阵是一个数字以行和列进行排列的二维数组。矩阵的大小由行和列的长度决定。如果矩阵 A 有 m 行和 n 列，那么它可以表示为一个具有 $m \times n$ 个元素的矩形对象（见图 1-4a），记作 $A_{m \times n}$。

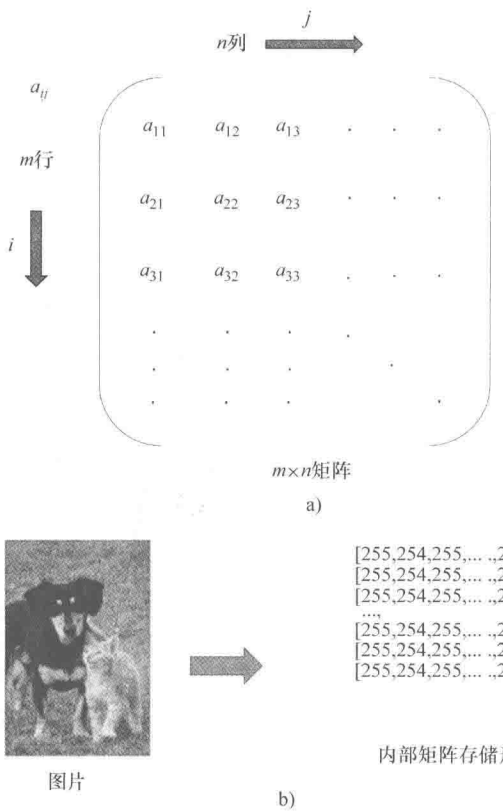


图 1-4 矩阵的结构

属于同一个向量空间的几个向量组合形成一个矩阵。

例如，灰度（黑白）图像以矩阵形式存储，图像的大小决定了矩阵的大小，并且每个矩阵中的元素数值介于 0 ~ 255 之间，它代表了像素亮度。图 1-4b 所示为灰度图像，随后是矩阵表示形式。

1.1.4 张量

张量（Tensor）是一个多维数组。事实上，向量和矩阵可以分别看作一维和二维张量。在深度学习中，张量主要用于存储和处理数据。例如，RGB 图像存储在三维张量中，其中沿着一个维度是水平数轴 x 轴，沿着另一个维度是垂直数轴 y 轴，第三个维度对应的是三个颜色通道，即红绿蓝三原色。另一个例子是在卷积神经网络中通过小批量提供图像的四维张

量：第一个维度是小批量中的图像编号，第二个维度是颜色通道，第三个和第四个维度分别对应水平和垂直方向上的像素位置。

1.1.5 矩阵的运算和操作

大多数深度学习的计算活动都是通过基本矩阵运算来完成的，例如乘法、加法、减法、转置等。因此，回顾基本的矩阵运算是很有意义的。

我们可以将 m 行 n 列的矩阵 A 看作包含 n 个并排堆叠的 m 维列向量的矩阵。我们将矩阵表示为

$$A_{m\times n}\in\mathbb{R}^{m\times n}$$

1. 矩阵加法

A 和 B 两个矩阵相加意味着它们每个对应元素相加。我们只能对两个维度相同的矩阵做加法运算。如果 C 是矩阵 A 和 B 的和，那么

$$c_{ij}=a_{ij}+b_{ij}\quad \forall i\in\{1,2,\cdots,m\},\forall j\in\{1,2,\cdots,n\}$$

式中，$a_{ij}\in A$；$b_{ij}\in B$；$c_{ij}\in C$。

例如，$A=\begin{bmatrix}1 & 2\\ 3 & 4\end{bmatrix}$，$B=\begin{bmatrix}5 & 6\\ 7 & 8\end{bmatrix}$，那么 $A+B=\begin{bmatrix}1+5 & 2+6\\ 3+7 & 4+8\end{bmatrix}=\begin{bmatrix}6 & 8\\ 10 & 12\end{bmatrix}$。

2. 矩阵减法

A 和 B 两个矩阵相减意味着它们每个对应元素相减。我们只能对两个维度相同的矩阵做减法运算。如果矩阵 C 代表 $A-B$，那么

$$c_{ij}=a_{ij}-b_{ij}\quad \forall i\in\{1,2,\cdots,m\},\forall j\in\{1,2,\cdots,n\}$$

式中，$a_{ij}\in A$；$b_{ij}\in B$；$c_{ij}\in C$。

例如，$A=\begin{bmatrix}1 & 2\\ 3 & 4\end{bmatrix}$，$B=\begin{bmatrix}5 & 6\\ 7 & 8\end{bmatrix}$，那么 $A-B=\begin{bmatrix}1-5 & 2-6\\ 3-7 & 4-8\end{bmatrix}=\begin{bmatrix}-4 & -4\\ -4 & -4\end{bmatrix}$。

3. 矩阵乘法

对于两个矩阵 $A\in\mathbb{R}^{m\times n}$ 和 $B\in\mathbb{R}^{p\times q}$，为了使它们可乘，$n$ 和 p 必须相等。作为运算结果的矩阵 $C\in\mathbb{R}^{m\times q}$，其元素可以表示为

$$c_{ij}=\sum_{k=1}^{n}a_{ik}b_{kj}\quad \forall i\in\{1,2,\cdots,m\},\forall j\in\{1,2,\cdots,q\}$$

例如，矩阵 A、$B\in\mathbb{R}^{2\times 2}$，相乘的计算步骤如下：

$$A=\begin{bmatrix}1 & 2\\ 3 & 4\end{bmatrix},\quad B=\begin{bmatrix}5 & 6\\ 7 & 8\end{bmatrix}$$

$$c_{11}=\begin{bmatrix}1 & 2\end{bmatrix}\begin{bmatrix}5\\ 7\end{bmatrix}=1\times5+2\times7=19\quad c_{12}=\begin{bmatrix}1 & 2\end{bmatrix}\begin{bmatrix}6\\ 8\end{bmatrix}=1\times6+2\times8=22$$

$$c_{21}=\begin{bmatrix}3 & 4\end{bmatrix}\begin{bmatrix}5\\ 7\end{bmatrix}=3\times5+4\times7=43\quad c_{22}=\begin{bmatrix}3 & 4\end{bmatrix}\begin{bmatrix}6\\ 8\end{bmatrix}=3\times6+4\times8=50$$

$$C=\begin{bmatrix}c_{11} & c_{12}\\ c_{21} & c_{22}\end{bmatrix}=\begin{bmatrix}19 & 22\\ 43 & 50\end{bmatrix}$$

4. 矩阵转置

矩阵 $A \in \mathbb{R}^{m \times n}$ 的转置一般用 $A^{\mathrm{T}} \in \mathbb{R}^{n \times m}$ 来表示，它通过交换行向量和列向量来获得。

$$a'_{ji} = a_{ij} \quad \forall i \in \{1,2,\cdots,m\}, \forall j \in \{1,2,\cdots,n\}$$

式中，$a'_{ji} \in A^{\mathrm{T}}$，$a_{ij} \in A$。

例如，$A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$，那么 $A^{\mathrm{T}} = \begin{bmatrix} 1 & 3 \\ 2 & 4 \end{bmatrix}$。

A 和 B 两个矩阵乘积的转置是 A 和 B 以相反顺序转置的乘积，即 $(AB)^{\mathrm{T}} = B^{\mathrm{T}}A^{\mathrm{T}}$。

举个例子，如果我们有两个矩阵 $A = \begin{bmatrix} 19 & 22 \\ 43 & 50 \end{bmatrix}$ 和 $B = \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix}$，然后 $(AB) = \begin{bmatrix} 19 & 22 \\ 43 & 50 \end{bmatrix}\begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix} = \begin{bmatrix} 95 & 132 \\ 301 & 400 \end{bmatrix}$，那么 $(AB)^{\mathrm{T}} = \begin{bmatrix} 95 & 301 \\ 132 & 400 \end{bmatrix}$。

现在 $A^{\mathrm{T}} = \begin{bmatrix} 19 & 43 \\ 22 & 50 \end{bmatrix}$，$B^{\mathrm{T}} = \begin{bmatrix} 5 & 7 \\ 6 & 8 \end{bmatrix}$

$$B^{\mathrm{T}}A^{\mathrm{T}} = \begin{bmatrix} 5 & 7 \\ 6 & 8 \end{bmatrix}\begin{bmatrix} 19 & 43 \\ 22 & 50 \end{bmatrix} = \begin{bmatrix} 95 & 301 \\ 132 & 400 \end{bmatrix}$$

所以，等式 $(AB)^{\mathrm{T}} = B^{\mathrm{T}}A^{\mathrm{T}}$ 成立。

5. 两个向量的点积（数量积）

任何一个 n 维向量都能表示为矩阵 $v \in \mathbb{R}^{n \times 1}$。让我们定义两个 n 维向量 $v_1 \in \mathbb{R}^{n \times 1}$、$v_2 \in \mathbb{R}^{n \times 1}$。

$$v_1 = \begin{bmatrix} v_{11} \\ v_{12} \\ \cdot \\ \cdot \\ \cdot \\ v_{1n} \end{bmatrix}, \quad v_2 = \begin{bmatrix} v_{21} \\ v_{22} \\ \cdot \\ \cdot \\ \cdot \\ v_{2n} \end{bmatrix}$$

两个向量的点积是它们（相同维度上）对应元素乘积的和，它可以表示为

$$v_1 \cdot v_2 = v_1^{\mathrm{T}}v_2 = v_2^{\mathrm{T}}v_1 = v_{11}v_{21} + v_{12}v_{22} + \cdots + v_{1n}v_{2n} = \sum_{k=1}^{n} v_{1k}v_{2k}$$

例如，

$$v_1 = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}, \quad v_2 = \begin{bmatrix} 3 \\ 5 \\ -1 \end{bmatrix}, \quad v_1 \cdot v_2 = v_1^{\mathrm{T}}v_2 = 1 \times 3 + 2 \times 5 - 3 \times 1 = 10。$$

6. 矩阵和向量之间的运算

当一个矩阵乘以一个向量时，结果是另一个向量。比方说，矩阵 $A \in \mathbb{R}^{m \times n}$ 乘以向量 $x \in \mathbb{R}^{n \times 1}$，其结果将会是另一个向量 $b \in \mathbb{R}^{m \times 1}$。