

从线性代数到张量计算

陈新宇（蒙特利尔大学）& 赵熙乐（电子科技大学）

2022 年 12 月

目录

第一章 代数结构	7
1.1 向量与矩阵	7
1.2 张量	7
1.3 特殊代数结构	7
1.3.1 循环矩阵	7
第二章 Kronecker 分解	9
2.1 朴素 Kronecker 分解	9
2.1.1 定义	9
2.1.2 引入 permute 概念	9
2.1.3 求解过程	10
2.2 广义 Kronecker 分解	11
2.3 模型参数压缩	11
第三章 低秩线性回归	13

前言

在过去的数十年间，随着信号处理、机器学习与数值计算等领域的快速发展，张量计算已从以线性代数为基础的矩阵计算中逐步拓展开来，相关研究贯穿信号处理、机器学习等众多领域。随着大量张量计算算法涌现出来，我们不难发现：这些算法大多建立在张量分解的基础上。本文以张量计算这一概念为核心，将从线性代数出发，讲述张量计算相关的一系列内容。为了提高读者的阅读体验，笔者进行了以下尝试：

- **化繁为简**。将线性代数以及张量计算的范畴限定在实空间中。另外，严格来说，向量和矩阵属于低阶张量，为区分概念，我们默认常提到的张量特指高阶张量（阶数大于或等于 3）。
- **由浅入深**。从基本的线性代数内容展开，通过循序渐进的方式引出一系列矩阵分解与张量分解技术，使读者体会到线性代数的巨大价值。
- **熟能生巧**。本文在撰写过程中尽可能考虑初学者的学习历程，在全文中设计一系列难度适中的例题让读者更直观理解一系列理论，并通过练习熟练掌握相应内容。

笔者深感自身才疏学浅，对于线性代数和张量计算的认识具有一定的局限性，请广大读者批评指正。另外，全文内容设置的合理性也有待考究，需要等待读者的检验。尽管如此，笔者愿竭心力，在后续版本中逐步更新与完善本文，如有建议或疑问，请在 GitHub 开源项目<https://github.com/xinychen/tensor-book>的问答区与笔者进行互动交流。

作者声明：

- 撰写本文的初衷在于传播知识，为感兴趣的读者提供参考素材。
- 禁止将本文放在其他网站上供人下载，唯一下载网站为https://xinychen.github.io/books/tensor_book.pdf。
- 禁止将本文用于任何形式的商业活动。

第一章 代数结构

长期以来，线性代数一直作为机器学习中最为重要的数学工具之一，被人们广泛用于开发各类机器学习算法。线性代数本质上是以向量与矩阵为基本代数结构，本书要讨论的张量分解等模型则主要以张量为基本代数结构。在过去的数十年间，借助线性代数这一基本数学工具，机器学习中涌现出了很多经典的代数模型，这其中不乏矩阵分解、主成分分析，而张量分解在某种程度上可看作是矩阵分解的一种衍生物。

近年来，张量分解在机器学习的众多问题中得到了很好的应用，但关于张量的一些计算与我们所熟悉的线性代数却大相径庭，同时，张量计算相比以矩阵计算为主导的线性代数更为抽象，这使得很多与张量分解相关的内容看起来晦涩难懂。实际上，向量与矩阵都是张量的特例，可以被定义为低阶张量。一般而言，向量是第 1 阶张量，英文表述为 first-order tensor；矩阵是第 2 阶张量，英文表述为 second-order tensor；第 3 阶或者更高阶数的张量被称为高阶张量，英文表述为 higher-order tensor。在各类文献中，通常提到的张量都是特指高阶张量，当然，这在本书的叙述中也不例外。需要注意的是，在各类程序语言中，人们更愿意将张量称为多维数组。

在一个矩阵中，某一元素的位置可以说是“第 i 行、第 j 列”，即要描述某一元素的位置需用到行和列索引构成的组合 (i, j) 。类似地，在一个第 3 阶张量中，描述某一元素的位置需用到三个索引构成的组合，例如 (i, j, k) 。在处理稀疏矩阵或稀疏张量时，用索引来标记元素的位置会节省下一些不必要的存储开支。

1.1 向量与矩阵

1.2 张量

1.3 特殊代数结构

1.3.1 循环矩阵

循环矩阵 (circulant matrix) 是一种特殊的代数结构，广泛应用于信号处理等。从定义出发，给定任意向量 $\mathbf{x} = (x_1, x_2, \dots, x_T)^\top \in \mathbb{R}^T$ ，其对应的循环矩阵可写作如下形式：

$$\mathcal{C}(\mathbf{x}) \triangleq \begin{bmatrix} x_1 & x_T & \cdots & x_2 \\ x_2 & x_1 & \cdots & x_3 \\ \vdots & \vdots & \ddots & \vdots \\ x_T & x_{T-1} & \cdots & x_1 \end{bmatrix} \in \mathbb{R}^{T \times T} \quad (1.1)$$

其中， $\mathcal{C} : \mathbb{R}^T \rightarrow \mathbb{R}^{T \times T}$ 表示循环算子 (circulant operator)。该循环矩阵的第一列为向量 \mathbf{x} 本身，对角线元素均为 x_1 。

例 1. 给定任意向量 $\mathbf{x} = (x_1, x_2, x_3, x_4, x_5)^\top \in \mathbb{R}^5$ ，试写出其对应的循环矩阵。

解. 向量 \mathbf{x} 对应的循环矩阵为

$$\mathcal{C}(\mathbf{x}) = \begin{bmatrix} x_1 & x_5 & x_4 & x_3 & x_2 \\ x_2 & x_1 & x_5 & x_4 & x_3 \\ x_3 & x_2 & x_1 & x_5 & x_4 \\ x_4 & x_3 & x_2 & x_1 & x_5 \\ x_5 & x_4 & x_3 & x_2 & x_1 \end{bmatrix} \in \mathbb{R}^{5 \times 5} \quad (1.2)$$

例 2. 给定任意向量 $\mathbf{x} = (x_1, x_2, \dots, x_T)^\top \in \mathbb{R}^T$ 与 $\mathbf{y} = (y_1, y_2, \dots, y_T)^\top \in \mathbb{R}^T$, 若两者之间的循环卷积 (*circular convolution*) 为 $\mathbf{z} = \mathbf{x} \star \mathbf{y} \in \mathbb{R}^T$, 其中, 符号 \star 表示卷积运算, 则向量 \mathbf{z} 的任意元素为

$$z_t = \sum_{k=1}^T x_{t-k+1} y_k, \forall t \in \{1, 2, \dots, T\} \quad (1.3)$$

其中, 当 $t+1 \leq k$ 时, 则令 $x_{t-k+1} = x_{t-k+1+T}$ 。试根据循环矩阵的定义写出循环卷积。

解. 在这里, 循环卷积可写作如下形式:

$$\mathbf{z} = \mathbf{x} \star \mathbf{y} = \begin{bmatrix} x_1 & x_T & \cdots & x_2 \\ x_2 & x_1 & \cdots & x_3 \\ \vdots & \vdots & \ddots & \vdots \\ x_T & x_{T-1} & \cdots & x_1 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_T \end{bmatrix} = \mathcal{C}(\mathbf{x})\mathbf{y} \quad (1.4)$$

第二章 Kronecker 分解

Kronecker 分解是一种以 Kronecker 积为基础的分解形式，又被称为 Kronecker 积分解、Kronecker 积逼近 (Kronecker product approximation)、最近 Kronecker 积 (nearest Kronecker product) 等，它是矩阵计算与张量计算中十分重要的逼近问题。本章主要讲述 Kronecker 分解的一般形式、逼近问题、求解过程等。

2.1 朴素 Kronecker 分解

2.1.1 定义

一般而言，给定任意矩阵 $\mathbf{X} \in \mathbb{R}^{(mp) \times (nq)}$ ，若 $\mathbf{A} \in \mathbb{R}^{m \times n}$, $\mathbf{B} \in \mathbb{R}^{p \times q}$ 为朴素 Kronecker 分解中的待定参数，则可将分解过程描述为如下逼近问题：

$$\min_{\mathbf{A}, \mathbf{B}} \|\mathbf{X} - \mathbf{A} \otimes \mathbf{B}\|_F^2 \quad (2.1)$$

其中，我们的建模目标是寻找最佳的矩阵 \mathbf{A}, \mathbf{B} 使得损失函数最小化。

为便于理解该逼近问题，不妨用一组小矩阵一窥究竟，令 $m = 3, n = p = q = 2$ ，则此时的目标函数为

$$\|\mathbf{X} - \mathbf{A} \otimes \mathbf{B}\|_F^2 = \left\| \begin{bmatrix} x_{11} & x_{12} & x_{13} & x_{14} \\ x_{21} & x_{22} & x_{23} & x_{24} \\ x_{31} & x_{32} & x_{33} & x_{34} \\ x_{41} & x_{42} & x_{43} & x_{44} \\ x_{51} & x_{52} & x_{53} & x_{54} \\ x_{61} & x_{62} & x_{63} & x_{64} \end{bmatrix} - \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \\ a_{31} & a_{32} \end{bmatrix} \otimes \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} \right\|_F^2 \quad (2.2)$$

2.1.2 引入 permute 概念

在这里，我们引入 permute 是为了对矩阵的维度按照特定规则进行调整，这一做法最早是由 Van Loan 和 Pitsianis 于 1993 年提出¹。在公式(2.2)中，首先使用分块矩阵表示矩阵 $\mathbf{X} \in \mathbb{R}^{6 \times 4}$ ：

$$\begin{bmatrix} x_{11} & x_{12} & x_{13} & x_{14} \\ x_{21} & x_{22} & x_{23} & x_{24} \\ x_{31} & x_{32} & x_{33} & x_{34} \\ x_{41} & x_{42} & x_{43} & x_{44} \\ x_{51} & x_{52} & x_{53} & x_{54} \\ x_{61} & x_{62} & x_{63} & x_{64} \end{bmatrix} = \begin{bmatrix} \mathbf{X}_{11} & \mathbf{X}_{12} \\ \mathbf{X}_{21} & \mathbf{X}_{22} \\ \mathbf{X}_{31} & \mathbf{X}_{32} \end{bmatrix} \quad (2.3)$$

¹C. Van Loan, N. Pitsianis (1993). Approximation with Kronecker products. Linear Algebra for Large Scale and Real-Time Applications, 232: 293-314.

其中, 分块矩阵为

$$\begin{aligned} \mathbf{X}_{11} &= \begin{bmatrix} x_{11} & x_{12} \\ x_{21} & x_{22} \end{bmatrix} & \mathbf{X}_{12} &= \begin{bmatrix} x_{13} & x_{14} \\ x_{23} & x_{24} \end{bmatrix} \\ \mathbf{X}_{21} &= \begin{bmatrix} x_{31} & x_{32} \\ x_{41} & x_{42} \end{bmatrix} & \mathbf{X}_{22} &= \begin{bmatrix} x_{33} & x_{34} \\ x_{43} & x_{44} \end{bmatrix} \\ \mathbf{X}_{31} &= \begin{bmatrix} x_{51} & x_{52} \\ x_{61} & x_{62} \end{bmatrix} & \mathbf{X}_{32} &= \begin{bmatrix} x_{53} & x_{54} \\ x_{63} & x_{64} \end{bmatrix} \end{aligned} \quad (2.4)$$

对这些分块矩阵进行向量化:

$$\text{vec}(\mathbf{X}_{11}) = \begin{bmatrix} x_{11} \\ x_{21} \\ x_{12} \\ x_{22} \end{bmatrix} \quad \text{vec}(\mathbf{X}_{21}) = \begin{bmatrix} x_{31} \\ x_{41} \\ x_{32} \\ x_{42} \end{bmatrix} \quad \cdots \quad \text{vec}(\mathbf{X}_{32}) = \begin{bmatrix} x_{53} \\ x_{63} \\ x_{54} \\ x_{64} \end{bmatrix} \quad (2.5)$$

使用这些向量构造如下矩阵:

$$\tilde{\mathbf{X}} = \begin{bmatrix} \text{vec}(\mathbf{X}_{11})^\top \\ \text{vec}(\mathbf{X}_{21})^\top \\ \text{vec}(\mathbf{X}_{31})^\top \\ \text{vec}(\mathbf{X}_{12})^\top \\ \text{vec}(\mathbf{X}_{22})^\top \\ \text{vec}(\mathbf{X}_{32})^\top \end{bmatrix} \in \mathbb{R}^{6 \times 4} \quad (2.6)$$

在这里, 将矩阵 \mathbf{X} 构造成矩阵 $\tilde{\mathbf{X}}$ 的过程通常被称为 permute。

由于

$$\begin{aligned} \text{vec}(\mathbf{X}_{11}) &= a_{11} \cdot \text{vec}(\mathbf{B}) \\ \text{vec}(\mathbf{X}_{21}) &= a_{21} \cdot \text{vec}(\mathbf{B}) \\ &\vdots \\ \text{vec}(\mathbf{X}_{32}) &= a_{32} \cdot \text{vec}(\mathbf{B}) \end{aligned} \quad (2.7)$$

此时, Kronecker 分解的逼近问题可写作:

$$\arg \min_{\mathbf{A}, \mathbf{B}} \|\mathbf{X} - \mathbf{A} \otimes \mathbf{B}\|_F^2 = \arg \min_{\mathbf{A}, \mathbf{B}} \|\tilde{\mathbf{X}} - \text{vec}(\mathbf{A})\text{vec}(\mathbf{B})^\top\|_F^2 \quad (2.8)$$

2.1.3 求解过程

对于公式(2.1)中的 Kronecker 分解逼近问题, 可根据 Eckhart-Young 定理对如下优化问题进行求解:

$$\min_{\mathbf{A}, \mathbf{B}} \|\tilde{\mathbf{X}} - \text{vec}(\mathbf{A})\text{vec}(\mathbf{B})^\top\|_F^2 \quad (2.9)$$

若 $\tilde{\mathbf{X}}$ 的奇异值分解为 $\tilde{\mathbf{X}} = \sum_{r=1}^{\min\{mn, pq\}} \sigma_r \mathbf{u}_r \mathbf{v}_r^\top$, 其中, 奇异值为 $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_{\min\{mn, pq\}}$, 则矩阵 \mathbf{A} 与 \mathbf{B} 的最优解为

$$\begin{cases} \text{vec}(\hat{\mathbf{A}}) = \sqrt{\sigma_1} \cdot \mathbf{u}_1 \\ \text{vec}(\hat{\mathbf{B}}) = \sqrt{\sigma_2} \cdot \mathbf{v}_1 \end{cases} \quad (2.10)$$

2.2 广义 Kronecker 分解

在《Convolutional neural network compression through generalized Kronecker product decomposition》中，作者给出了一种广义 Kronecker 分解。形式上说，给定任意矩阵 $\mathbf{X} \in \mathbb{R}^{(mp) \times (nq)}$ ，若 $\mathbf{A}_r \in \mathbb{R}^{m \times n}$, $\mathbf{B}_r \in \mathbb{R}^{p \times q}$, $r = 1, 2, \dots, R$ 为广义 Kronecker 分解中的待定参数，则可将分解过程描述为如下逼近问题：

$$\min_{\{\mathbf{A}_r, \mathbf{B}_r\}_{r=1}^R} \left\| \mathbf{X} - \sum_{r=1}^R \mathbf{A}_r \otimes \mathbf{B}_r \right\|_F^2 \quad (2.11)$$

其中，我们的建模目标是寻找最佳的矩阵 $\{\mathbf{A}_r, \mathbf{B}_r\}_{r=1}^R$ 使得损失函数最小化。

与朴素 Kronecker 分解类似，可先将广义 Kronecker 分解的逼近问题写作如下形式：

$$\arg \min_{\{\mathbf{A}_r, \mathbf{B}_r\}_{r=1}^R} \left\| \mathbf{X} - \sum_{r=1}^R \mathbf{A}_r \otimes \mathbf{B}_r \right\|_F^2 = \arg \min_{\{\mathbf{A}_r, \mathbf{B}_r\}_{r=1}^R} \left\| \mathbf{X} - \sum_{r=1}^R \text{vec}(\mathbf{A}_r) \text{vec}(\mathbf{B}_r)^\top \right\|_F^2 \quad (2.12)$$

其中，矩阵 $\tilde{\mathbf{X}}$ 是由矩阵 \mathbf{X} 进行 permute 构造得到。

根据 Eckhart-Young 定理对上述优化问题进行求解，若矩阵 $\tilde{\mathbf{X}}$ 的奇异值分解为 $\tilde{\mathbf{X}} = \sum_{r=1}^{\min\{mn, pq\}} \sigma_r \mathbf{u}_r \mathbf{v}_r^\top$ ，其中，奇异值为 $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_{\min\{mn, pq\}}$ ，则矩阵 \mathbf{A}_r 和 \mathbf{B}_r 的最优解为

$$\begin{cases} \text{vec}(\hat{\mathbf{A}}_r) = \sqrt{\sigma_r} \mathbf{u}_r \\ \text{vec}(\hat{\mathbf{B}}_r) = \sqrt{\sigma_r} \mathbf{v}_r \end{cases} \quad (2.13)$$

2.3 模型参数压缩

Kronecker 分解的一个重要用途是压缩模型参数。以多元线性回归 (multivariate linear regression) 为例，给定输入、输出数据为 $\mathcal{D} = \{(\mathbf{x}_1, \mathbf{y}_1), \dots, (\mathbf{x}_N, \mathbf{y}_N)\} \in \mathbb{R}^{nq} \times \mathbb{R}^{mp}$ ，则多元线性回归的优化问题为

$$\min_{\mathbf{W}} \frac{1}{2} \sum_{n=1}^N \|\mathbf{y}_n - \mathbf{W} \mathbf{x}_n\|_2^2 \quad (2.14)$$

令

$$\mathbf{X} = \begin{bmatrix} | & & | \\ \mathbf{x}_1 & \cdots & \mathbf{x}_N \\ | & & | \end{bmatrix} \in \mathbb{R}^{nq \times N} \quad (2.15)$$

$$\mathbf{Y} = \begin{bmatrix} | & & | \\ \mathbf{y}_1 & \cdots & \mathbf{y}_N \\ | & & | \end{bmatrix} \in \mathbb{R}^{mp \times N}, \quad (2.16)$$

则此时多元线性回归的等价优化问题为

$$\min_{\mathbf{W}} \frac{1}{2} \|\mathbf{Y} - \mathbf{W} \mathbf{X}\|_F^2 \quad (2.17)$$

不妨假设这里的系数矩阵 $\mathbf{W} \in \mathbb{R}^{(mp) \times (nq)}$ 存在一个广义 Kronecker 分解，且由 R 个成分构成，则基于广义 Kronecker 分解的多元线性回归可写作如下形式：

$$\min_{\{\mathbf{A}_r, \mathbf{B}_r\}_{r=1}^R} \frac{1}{2} \left\| \mathbf{Y} - \sum_{r=1}^R (\mathbf{A}_r \otimes \mathbf{B}_r) \mathbf{X} \right\|_F^2 \quad (2.18)$$

将优化问题改写为如下形式即可得到一个标准的广义 Kronecker 分解：

$$\min_{\{\mathbf{A}_r, \mathbf{B}_r\}_{r=1}^R} \frac{1}{2} \left\| \mathbf{Y} \mathbf{X}^\dagger - \sum_{r=1}^R (\mathbf{A}_r \otimes \mathbf{B}_r) \right\|_F^2 \quad (2.19)$$

从而可根据广义 Kronecker 分解的求解方法对该多元线性回归问题进行求解。

例 3 (矩阵自回归模型²). 对于多维时间序列 (*multidimensional time series*), 若任意时刻 t 对应的观测数据为矩阵 $\mathbf{X}_t \in \mathbb{R}^{M \times N}$, 则矩阵自回归的表达式为

$$\mathbf{X}_t = \mathbf{A} \mathbf{X}_{t-1} \mathbf{B}^\top + \mathbf{E}_t, \quad t = 2, 3, \dots, T \quad (2.20)$$

其中, $\mathbf{A} \in \mathbb{R}^{M \times M}$ 与 $\mathbf{B} \in \mathbb{R}^{N \times N}$ 为自回归过程的系数矩阵 (*coefficient matrix*); 矩阵 $\mathbf{E}_t \in \mathbb{R}^{M \times N}$ 为自回归过程的残差矩阵 (*residual matrix*)。

²http://www.stat.rutgers.edu/home/rongchen/publications/20JoE_Matrix_AR.pdf

第三章 低秩线性回归

线性回归是机器学习中的一个基本模型，常用于各类回归问题，其建模思路是采用线性方程对给定的变量建立线性关系。本章以线性回归模型为基础，将介绍低秩线性回归模型、低秩自回归模型、时变低秩自回归模型等，这些模型的核心是借助矩阵分解或张量分解对模型参数进行压缩。