第一讲 代数与矩阵基础（1）

李宇峰

liyf@nju.edu.cn

人工智能学院

矩阵计算的历史和发展

矩阵计算的历史和发展历程：

- 古代起源
  - 中国：成书于西汉末、东汉初的《九章算术》用分离系数法表示线性方程组，得到其增广矩阵；但当时仅作为线性方程组的标准表示与处理方式，没有形成独立的矩阵理论

  - 古希腊：古希腊数学家如毕达哥拉斯、欧几里得等人对线性方程组和行列式性质的研究，为矩阵的发展奠定了一定基础

矩阵计算的历史和发展

- 近代发展
  - <u>行列式理论奠基</u>：1693年，莱布尼茨在写给洛比达的一封信中使用并给出了行列式，并给出方程组的系数行列式为零的条件；同时代的日本数学家关孝和在其著作《解伏题元法》中也提出了行列式的概念与算法。1750年，瑞士数学家克莱姆发现了克莱姆法则。
  - <u>矩阵概念形成</u>：1801年，德国数学家高斯把线性变换的全部系数作为了一个整体。1844年，德国数学家爱森斯坦讨论了“变换”（矩阵）及其乘积。1850年，英国数学家西尔维斯特首先使用“矩阵”一词。
  - <u>矩阵理论创立</u>：1858年，英国数学家凯莱发表《关于矩阵理论的研究报告》，他首先将矩阵作为一个独立的数学对象加以研究，给出了两矩阵相等、零矩阵、单位矩阵等一系列定义，以及矩阵的基本运算。1879年，费罗贝尼乌斯引入矩阵秩的概念，至此，矩阵的体系基本建立。

矩阵计算的历史和发展

- 现代拓展
  - **理论完善**：20世纪初，美国数学家台尔曼提出了矩阵的谱定理，为解决线性微分方程组的问题提供了新的工具。英国数学家哈密尔顿和德国数学家弗罗贝尼乌斯进一步发展了矩阵的理论，引入了矩阵的指数、对数以及三角函数等概念，并研究了矩阵的相似、合同以及特征值等问题。
  - **应用拓展**：矩阵在物理学中用于描述量子力学中的波函数等；在计算机科学中用于表示图像、声音等数据，进行数值计算和图形渲染等；在经济学中用于描述统计数据和模型等。随着机器学习等领域兴起，矩阵在其中也扮演着重要角色，如线性回归、神经网络等算法都依赖矩阵运算
  - **芯片领域**：以英伟达的 GPU 架构发展为例，2017 年推出的 volta 架构引入张量核心，专门针对深度学习计算中的矩阵乘法和累加操作进行优化；2018 年的 turing 架构增强了 tensorcore 功能，新增对多种整数格式支持；2020 年 ampere 架构支持 tf32 和 bf16 数据格式，还引入对稀疏矩阵计算的支持；2022 年 hopper 架构引入 fp8 张量核心；2024 年 blackwell 架构推出第二代 transformer 引擎

矩阵计算的应用

- 矩阵计算在大模型和人工智能领域应用广泛，涵盖神经网络、自然语言处理、计算机视觉等多个方面，例如，
- 神经网络
  - 神经元连接权重表示：神经网络中，神经元之间的连接权重通常用矩阵表示。例如在一个具有n个输入神经元和m个输出神经元的全连接层中，连接权重可以表示为一个$n \times m$ 的矩阵。通过调整这些矩阵中的元素（即权重值），模型能够学习到输入数据与输出之间的复杂映射关系
  - 前向传播计算：在神经网络的前向传播过程中，矩阵乘法是核心计算操作。以多层感知机（MLP）为例，每一层的输入向量与该层的权重矩阵相乘，再加上偏置向量，经过激活函数得到该层的输出，这个过程可以用矩阵运算高效地实现。假设输入向量为$\mathbf{x} \in \mathbb{R}^n$，权重矩阵为$\mathbf{W} \in \mathbb{R}^{n\times m}$ ，偏置向量为 $\mathbf{b} \in \mathbb{R}^m$，则该层的输出$\mathbf{y} = \sigma(\mathbf{W}\mathbf{x} + \mathbf{b})$ ，其中 σ 为激活函数
  - 反向传播求导：在反向传播算法中，需要计算损失函数关于权重矩阵的梯度，以更新权重。这涉及到对矩阵乘法和激活函数的求导，同样依赖矩阵计算来高效地实现梯度的计算和传播，例如计算梯度时会用到矩阵的转置、Hadamard积等操作。

矩阵计算的应用

- 自然语言处理
  - 词向量表示：将单词映射为低维向量空间中的向量，词向量矩阵将每个词的向量组合在一起。如在Word2Vec模型中，通过训练得到的词向量矩阵可以捕捉单词之间的语义关系。当进行文本分类、情感分析等任务时，将输入文本中的词转换为对应的词向量，再通过矩阵运算进行特征提取和模型预测。
  - 注意力机制：在Transformer架构中，注意力机制通过计算查询向量（Query）、键向量（Key）和值向量（Value）之间的关系来动态地关注输入序列中的不同部分。其中，查询、键和值通常是通过对输入序列进行线性变换得到的，这些线性变换可以用矩阵乘法来实现。具体来说，计算注意力得分的公式为

$$Attention(Q,K,V) = \mathrm{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$

其中Q、K、V分别是查询、键、值矩阵，d_k是键向量的维度

矩阵计算的应用

- 计算机视觉
  - 图像特征提取：卷积神经网络（CNN）在计算机视觉中广泛用于图像特征提取。卷积层中的卷积核可以看作是小型的权重矩阵，通过在图像上滑动卷积核进行卷积操作，实际上就是对图像的局部区域与卷积核矩阵进行乘法和求和运算，从而提取出图像的各种特征，如边缘、纹理等。
  - 图像变换：在计算机视觉中，经常需要对图像进行各种变换，如旋转、缩放、平移等。这些变换可以用齐次坐标和变换矩阵来表示。例如，二维图像的旋转和平移可以用一个3X3的变换矩阵来实现，通过将图像中的每个像素点的坐标与变换矩阵相乘，得到变换后的像素点坐标。

矩阵计算的应用

- 强化学习
  - 状态价值函数估计：在强化学习中，状态价值函数V(s)表示在状态s下的长期累积奖励的期望。在基于模型的强化学习方法中，通常会用矩阵来表示状态转移概率和奖励函数，通过矩阵运算来求解最优策略和估计状态价值函数。例如，使用动态规划算法求解最优策略时，需要对状态转移矩阵和奖励矩阵进行迭代计算，找到最优价值函数和策略。
  - 策略评估与改进：在策略迭代算法中，需要对当前策略进行评估，计算策略的价值函数，这涉及到对状态转移矩阵和奖励矩阵的多次乘法运算。然后根据价值函数来改进策略，通过不断迭代这个过程，最终收敛到最优策略。

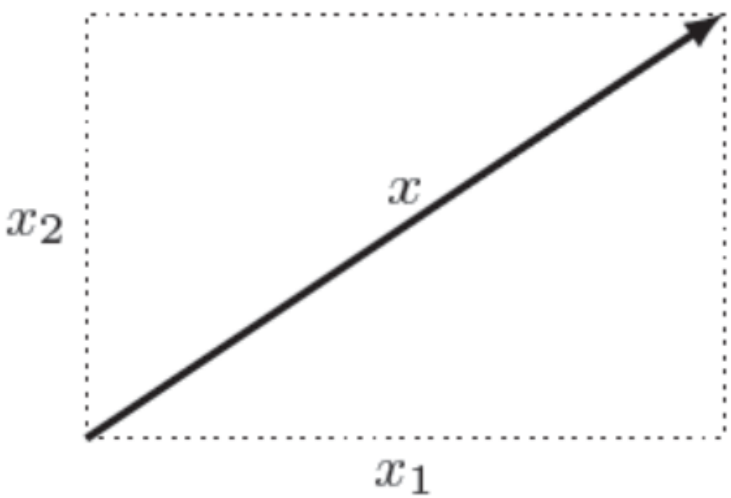
代数与矩阵基础

- 向量，矩阵
- 线性空间，线性子空间
  - 零空间，像空间
- 线性相关，线性无关
  - 空间的维数与基底
- 矩阵的基本运算
  - 转置，共轭转置
  - 相乘，求逆
- 矩阵的数值特征
  - 行列式，二次型，特征值，trace，rank
- 内积与范数
  - 向量范数
  - 矩阵范数

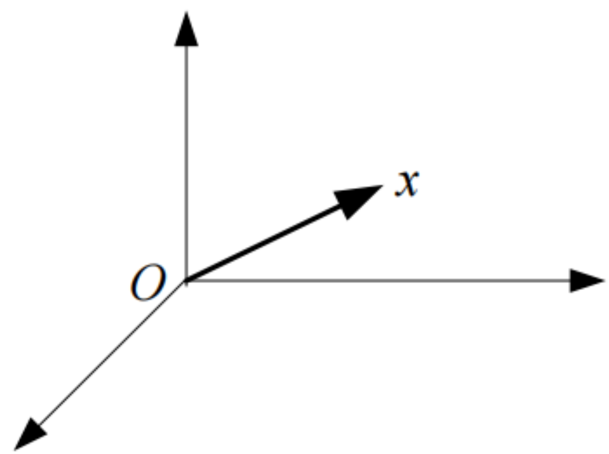
列向量与行向量

- 基本符号：P代表数域，R代表实数域，C代表复数域，Z代表整数域。
- 一个m维向量（m-vector）定义为：$x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{bmatrix}$.
- 若其元素$x_i\,(i = 1,2,\cdots, m)$取实数，即$x_i \in R$，称m维实向量，也简记为$x \in R^m = R^{m\times 1}$。
- 如果其元素是复数，就是m维复向量，记成$x \in C^m$.
- 一个m维行向量定义为$x = [x_1, x_2, \dots, x_m]$，记作$x \in R^{1\times m}$，或者$x \in R^{1\times m}$.
- 一个m维的列向量也可以写出m维行向量的转置形式，即$x = [x_1, x_2, \dots, x_m]^T$.

列向量与行向量



A 2-vector



A 3-vector

矩阵

- 一个$m \times n$矩阵(matrix)定义为

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix} = [a_{ij}]_{m\times n}$$

若其元素$a_{ij} \in R$，则就称A为$m \times n$实矩阵，表示为$A \in R^{m\times n}$.

- $m \times n$矩阵A是由m行n列所组成，其可以表示成n个列向量的形式，就是

$$A = [a_1, a_2, \dots, a_n],$$

其第i列向量为$a_i = (a_{1i}, a_{2i}, \dots, a_{mi})^T$，$i = 1, 2, \dots, n$.

线性空间

学习矩阵计算中的线性空间具有多方面的重要意义，例如：

理论基础方面

- **统一数学结构**：线性空间是一种抽象的数学结构，它为矩阵计算以及许多其他数学领域提供了一个统一的框架。通过定义线性空间的基本性质和运算规则，如向量的加法和数乘满足的一系列公理，可以将各种看似不同但本质上具有线性性质的对象和问题统一进行研究和处理。矩阵作为线性空间中的一种重要元素，在这个框架下可以更清晰地理解其性质和运算规律。

- **深化数学理解**：学习线性空间有助于深入理解线性代数的基本概念和原理。矩阵计算中的很多概念，如矩阵的秩、线性方程组的解、特征值和特征向量等，都与线性空间的理论紧密相关。例如，矩阵的秩可以理解为矩阵所对应的线性变换的值域空间的维数，通过线性空间的视角能更直观地理解秩的本质和意义，从而为解决矩阵计算问题提供更深刻的理论支持。

线性空间

实际应用方面
- 数据分析与机器学习

- **数据降维**：在数据分析中，经常需要对高维数据进行降维处理，以提取关键信息和降低计算复杂度。主成分分析（PCA）就是一种基于线性空间的方法，它通过对数据的协方差矩阵进行特征值分解，找到数据在低维线性子空间中的最佳表示，实现数据的降维。

- **机器学习算法**：许多机器学习算法都依赖于矩阵计算和线性空间的理论。例如，在支持向量机（SVM）中，通过将数据映射到高维特征空间，然后利用线性空间中的内积运算和矩阵求解来构建分类超平面；在神经网络中，神经元之间的连接权重可以用矩阵表示，通过矩阵乘法和加法实现信息的传递和处理。

线性空间

- 科学与工程计算

- **物理问题求解：**在量子力学、电磁学等物理学领域，很多问题都可以用线性空间和矩阵来描述。例如，量子力学中的态矢量可以看作是希尔伯特空间（一种特殊的线性空间）中的向量，而物理算符则对应着矩阵。通过矩阵计算在这些线性空间中的应用，可以求解物理系统的能量、状态演化等重要问题。

- **工程系统建模：**在控制工程、信号处理等领域，线性空间和矩阵计算是系统建模和分析的重要工具。例如，在控制系统中，状态空间模型就是基于线性空间理论建立的，通过矩阵描述系统的状态转移和输入输出关系，从而实现对系统的稳定性分析、控制器设计等。

线性空间

培养思维能力方面

- **抽象思维培养**：线性空间的学习需要从具体的向量和矩阵实例中抽象出一般的概念和性质，这有助于培养抽象思维能力。能够将具体问题抽象为线性空间中的数学模型，然后运用线性空间的理论和方法进行分析和解决，是一种重要的数学思维方式，对于解决各种复杂的科学和工程问题具有重要意义。

- **逻辑推理训练**：线性空间的理论体系具有严密的逻辑性，从基本定义和公理出发，通过一系列的逻辑推理和证明，可以得出许多重要的定理和结论。学习过程中需要进行大量的逻辑推理和证明工作，这有助于训练逻辑思维能力，提高数学证明和问题求解的能力。

线性空间

线性空间(linear space).

定义1.1 线性空间是指在一个集合S上定义了加法和数乘运算，且数乘满足下列线性规则，即$\forall \alpha, \beta \in P, \forall x, y \in S$，均有

$$\alpha(x+y)=\alpha x+\alpha y$$

$$(\alpha+\beta)x=\alpha x+\beta x$$

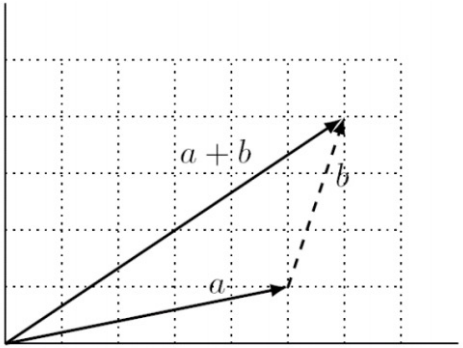
$$\alpha(\beta x)=(\alpha\beta)x$$

线性空间

其实我们已经学过许多线性空间的例子。譬如

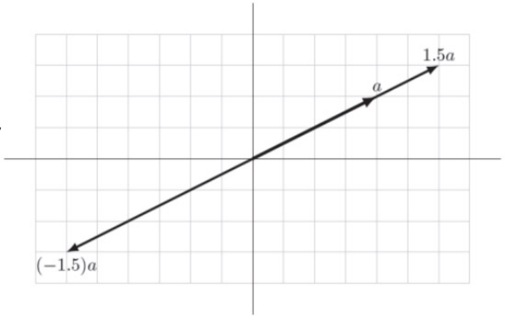
例1.1 R^n是线性空间，其中“加法”运算定义为

$$a+b=(a_1+b_1,a_2+b_2,\dots,a_n+b_n)^T$$



数乘运算为

$$\lambda a=(\lambda a_1,\lambda a_2,\dots,\lambda a_n)^T$$

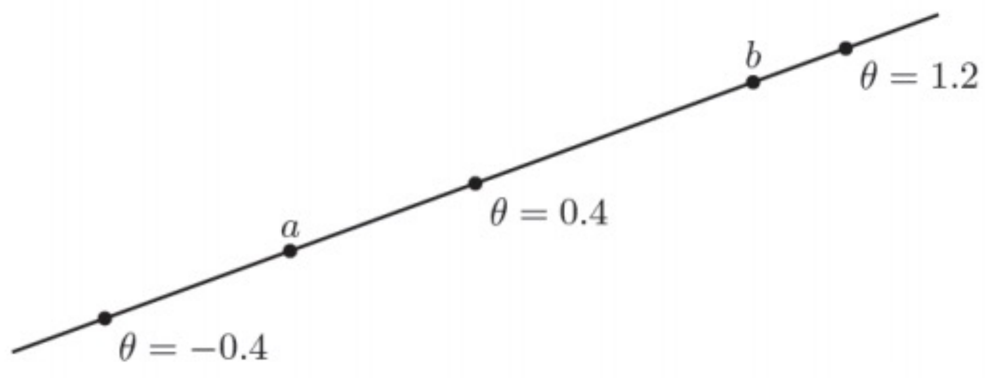


其中$a=(a_1,a_2,\dots,a_n)^T, b=(b_1,b_2,\dots,b_n)^T$

线性空间

思考：在n维实空间中，经过两个向量a和b的直线怎么表示？连接它们的线段怎么表示？

线性空间



$(1-\theta)a + \theta b$ for different values of θ.

线段：$\mathbf{a}+\theta\cdot(\mathbf{b}-\mathbf{a}),\ \theta\in[0, 1]$

直线：$\mathbf{a}+\theta\cdot(\mathbf{b}-\mathbf{a}),\ \theta\in R$

线性空间

- 若S和Q同为线性空间，且$Q \subset S$，则称Q为S的线性子空间（linear subspace）.

例1.2 次数$\leq n$的实系数多项式的全体

$$P_n(\lambda) = \{a \mid a = a_n\lambda^n + a_{n-1}\lambda^{n-1} + \cdots + a_1\lambda + a_0, a_i \in R\}$$

是线性空间。

思考题：最小的线性空间是什么？次数为n的多项式的集合是线性空间吗？

零空间与像空间

零空间和像空间是矩阵的两个重要子空间，它们能帮助我们更深入地理解矩阵的本质和性质。

零空间由所有满足Ax=0的向量构成，它体现了矩阵A的“核”，即那些被矩阵映射到零向量的向量集合，反映了矩阵在变换过程中“丢失”信息的情况。

像空间则是由矩阵A的列向量张成的空间，它代表了矩阵A作用于向量后所有可能的输出结果，展示了矩阵的“作用范围”和“影响力”。

通过研究这两个空间，能更全面地把握矩阵在向量空间中的作用和性质。

零空间与像空间

- **应用层面：数据分析与机器学习领域**
  - **信号处理与滤波：**在信号处理中，矩阵常常用于对信号进行变换和处理。零空间可以用于设计信号滤波器，例如在图像去噪中，通过分析噪声在某个变换矩阵下的零空间特性，设计滤波器去除噪声，保留有用信号。像空间则可以用来表示信号经过变换后的特征空间，有助于提取信号的关键特征，进行信号的分类和识别。

  - **数据降维与特征提取：**在主成分分析（PCA）中，像空间的概念起着关键作用。通过对数据的协方差矩阵进行特征分解，得到的主成分就是像空间的一组基向量，它们能够最大程度地保留数据的方差信息，实现数据的降维。而零空间在一些异常检测算法中有所应用，例如，如果某个数据点在经过某种变换后落入了零空间附近，可能意味着它是一个异常点，因为正常数据点经过变换后通常不会在零空间中。

零空间与像空间

- 函数$f(x)$有定义域(domain)与值域(range)。定义域是自变量x所有可取值的集合；值域是由定义域中一切元素所能得到的所有函数值的集合。

- 设$A \in R^{m\times n}$为一个$m\times n$矩阵，其所对应的变换$y = Ax(x \in R^n)$的值域(像空间)定义为

$$R(A) = \{y \in R^m \mid y = Ax, x \in R^n\}$$

零空间（核空间）：$Ax = 0$的所有解向量$x \in R^n$组成的集合

$$N(A) = \{x \in R^n \mid Ax = 0, x \in R^n\}$$

零空间与像空间

例1.3 给定$A \in R^{m\times n}$，求证$R(A)$是R^m的子空间；$N(A)$是R^n的子空间。

证明：要证明实空间S是否是线性子空间，只要证明

$\forall x, y \in S, \forall \alpha, \beta \in R$，都有$\alpha x + \beta y \in S$.

下面分别证明$R(A)$是线性空间.

设$y_1, y_2 \in R(A) \subset R^m, \alpha, \beta \in R$，根据$R(A)$的定义，故必有

$x_1, x_2 \in R^n$使得$Ax_1 = y_1, Ax_2 = y_2$

$\Rightarrow \alpha y_1 + \beta y_2 = \alpha A x_1 + \beta A x_2 = A(\alpha x_1 + \beta x_2)$

故推得$\forall y_1, y_2 \in R(A) \subset R^m, \forall \alpha, \beta \in R$得到$\alpha y_1 + \beta y_2 \in R(A)$.

所以$R(A)$是一个子空间。同理可以证明$N(A)$也是线性空间。

零空间与像空间

如果$A \in R^{m\times n}$，则

(1) A的零空间 $N(A) = \{x \in R^n \mid Ax = 0, x \in R^n\} \subset R^n$，就是说$N(A)$是$n$维实向量空间的子空间，$N(A)$中的向量是$n$维向量；

(2) A的像空间 $R(A) = \{x \in R^m \mid y = Ax, \forall x \in R^n\} \subset R^m$，就是说$R(A)$是$m$维实向量空间的子空间，$R(A)$中的向量是$m$维向量。

- 若已知向量$b \in R(A)$，则一定存在向量x，使得$b = Ax$.
- 因此，线性方程组$Ax = b$是否有解最本质的条件是：右端向量是否有$b \in R(A)$.

零空间与像空间

“**像空间，也是向量系的生成空间**”

设有一组向量系$x_1, x_2, \ldots, x_k$，其中$x_i \in R^n, i = 1,2, \ldots, k$，则这一组向量系的所有线性组合的集合：

$$S = \{y \in R^n \mid y = \alpha_1 x_1 + \alpha_2 x_2 + \cdots + \alpha_k x_k, \forall \alpha_1, \alpha_2, \ldots, \alpha_k\}$$

容易证明S是R^n的一个子空间，称为这一组向量系的生成空间（$Span$），因此**A的像空间，就是A的列向量张成的空间**。

线性相关与线性无关

学习矩阵计算中的线性相关和线性无关具有极其重要的意义，体现在理论构建、实际应用以及思维提升等多个层面，具体如下：

理论基础方面

•**理解向量空间结构：**线性相关与线性无关是刻画向量空间中向量之间关系的基本概念。向量空间中的向量通过线性相关和线性无关的关系，形成了特定的结构。例如，一组线性无关的向量可以作为向量空间的基，它们能够张成整个向量空间，而线性相关的向量则意味着存在冗余信息，无法构成向量空间的最简表示。理解这些概念有助于深入把握向量空间的本质和结构，为进一步研究矩阵在向量空间中的作用奠定基础。

•**构建线性代数理论体系：**线性相关和线性无关是线性代数理论体系的重要基石。许多重要的线性代数概念和定理都与它们密切相关，如矩阵的秩、行列式、线性方程组的解等。例如，矩阵的秩等于其列向量组中线性无关向量的最大个数，通过判断向量组的线性相关性可以确定矩阵的秩，进而判断线性方程组是否有解以及解的个数等问题。

线性相关与线性无关

实际应用方面，数据处理与分析

- **特征提取与降维**：在数据分析和机器学习中，常常需要对高维数据进行处理。如果数据的特征向量之间存在线性相关关系，那么这些特征中就存在冗余信息。通过识别和去除线性相关的特征，可以实现数据的降维，提取出更具有代表性和独立性的特征，提高模型的训练效率和准确性。例如，在主成分分析（PCA）中，就是基于寻找数据中线性无关的主成分来实现降维的。

- **数据建模与预测**：在建立数据模型时，确保输入变量之间线性无关可以避免模型出现多重共线性问题。多重共线性会导致模型参数估计不稳定，影响模型的准确性和可靠性。通过判断变量的线性相关性，合理选择变量，可以构建更有效的数据模型，提高对数据的预测和解释能力。

线性相关与线性无关

- 向量系的线性相关和线性无关，是矩阵论中最重要的概念之一。

对于给定的一组向量系$\{x_1, x_2, \dots, x_m\}, \forall x_i \in R^n$，如果仅仅当$\alpha_1, \alpha_2, \dots, \alpha_m$都为零时，其线性组合

$$\alpha_1 x_1 + \alpha_2 x_2 + \cdots + \alpha_m x_m = 0,$$

则就称向量系$\{x_1, x_2, \dots, x_m\}$是线性无关的，反之就是线性相关的。

- 向量系不是线性相关的，就是线性无关的，而线性无关的概念和基底、矩阵的秩(*rank*)，空间的维数紧密相关。

线性相关与线性无关

- 如果在线性空间S中存在有n个向量线性无关，而任何n+1个向量均线性相关，则称空间S的维数（dimension）为n，并记为dim(S)=n。

- 若S是定义在数域R上的线性空间，且S中的向量系$\{x_1, x_2, \ldots, x_n\}$满足以下两条

1. $\{x_1, x_2, \ldots, x_n\}$线性无关；
2. $\forall x \in S$, x都可以表示成$x = \alpha_1 x_1 + \alpha_2 x_2 + \cdots + \alpha_n x_n$，则称向量系$x_1, x_2, \ldots, x_n$是空间$S$的一组基底（basis）。

问题：基底上的表示 $x = \alpha_1 x_1 + \alpha_2 x_2 + \cdots + \alpha_n x_n$ 是否唯一？

$$\sum_i a_i x_i = \sum_i b_i x_i$$

$$\sum_i (a_i - b_i) \cdot x_i = 0$$

$$a_i = b_i, \forall i$$

思考

• 线性空间的“交”、“并”、“和”、“直接和”四种运算的定义如下：

$$T\cap V=\{x\mid x\in T\text{且}x\in V\}$$

$$T\cup V=\{x\mid x\in T\text{或}x\in V\}$$

$$T+V=\{x\mid x=y+z,y\in T,z\in V\}$$

$$T\oplus V=\{y\mid y=x+v,x\in T,v\in V,T\cap V=\{0\}\}$$

思考：以上四个集合中，都是子空间吗？

思考

- 首先看$T\cap V$，

设$\forall x,y\in T\cap V\Rightarrow x,y\in T$且$x,y\in V$，故$\forall\alpha,\beta\in R,\alpha x+\beta y\in T$且$\alpha x+\beta y\in V\ \Rightarrow\alpha x+\beta y\in T\cap V\Rightarrow T\cap V$是子空间。

- 注意$T\cup V$，

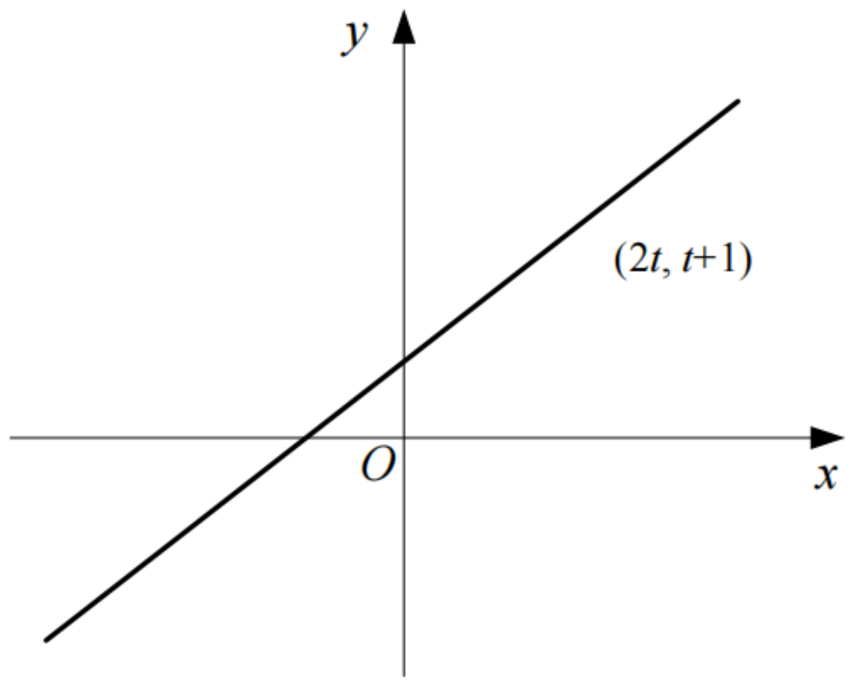
比如设 $T=\{x\mid x=(x_1,0),x_1\in R\},V=\{x\mid x=(0,x_2),x_2\in R\}$，则$\forall x,y\in T\cup V$，就有可能$x+y\notin T\cup V$，对于通常定义的加法不具有封闭性。所以$T\cup V$不是$R^2$的子空间。

$$(1,0)+(0,1)=(1,1)$$
$$(1,1)\notin T,(1,1)\notin V$$

思考题：集合$T+V$、$T\oplus V$是子空间吗？

思考

例1.5 考虑二维几何空间的直线 $V_1 = \{(2t, t+1) \mid t \in R\}$，它是二维几何空间的子空间吗？



矩阵与线性方程组

线性方程组对于矩阵计算的意义

提供实际背景和问题来源

线性方程组是实际问题中广泛存在的数学模型，它为矩阵计算赋予了实际意义。许多科学和工程领域的问题，如物理中的电路分析、力学中的平衡问题、经济中的投入-产出模型等，都可以转化为线性方程组的形式。例如，在电路分析中，根据基尔霍夫定律建立的电流和电压方程就是一个线性方程组。这些实际问题促使我们去研究矩阵的各种运算和性质，以找到线性方程组的有效解法。矩阵计算中的许多概念和方法，如矩阵的乘法、行列式、逆矩阵等，都是为了解决线性方程组而发展起来的。

检验矩阵计算方法的有效性

线性方程组的求解过程是检验矩阵计算方法正确性和有效性的重要途径。通过将矩阵计算方法应用于线性方程组的求解，并与已知的解析解或实际问题的预期结果进行比较，可以验证矩阵计算方法的准确性和稳定性。例如，高斯消元法是一种常用的求解线性方程组的矩阵计算方法，我们可以通过求解不同类型的线性方程组，观察其计算结果的精度和计算效率，来评估该方法的性能。同时，线性方程组的特殊结构和性质也可以帮助我们发现矩阵计算方法中的潜在问题，并对其进行改进和优化。

矩阵与线性方程组

矩阵计算对于线性方程组的意义

简化线性方程组的表示

矩阵是线性方程组的一种简洁表示方式。对于一个含有n个未知数$x_1, x_2, \cdots, x_n$的m元线性方程组

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n = b_2 \\ \cdots \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n = b_m \end{cases}$$

，可以用矩阵方程$Ax = b$来表示，其中$A = (a_{ij})$是$m \times n$的系数矩阵，$x = (x_1, x_2, \cdots, x_n)^T$是未知数向量，$b = (b_1, b_2, \cdots, b_m)^T$是常数项向量。这种表示方式不仅简洁明了，而且便于进行理论分析和计算。

提供有效的求解方法

矩阵计算为线性方程组的求解提供了一系列系统而有效的方法。例如，高斯消元法通过对增广矩阵进行初等行变换，将线性方程组化为行阶梯形矩阵或行最简形矩阵，从而方便地求出方程组的解；矩阵的逆运算可以用于求解系数矩阵可逆的线性方程组，若A可逆，则线性方程组$Ax = b$的解为$x = A^{-1}b$；克莱姆法则则利用行列式来求解系数矩阵为方阵且行列式不为零的线性方程组。这些方法都基于矩阵的运算和性质，大大提高了线性方程组求解的效率和准确性。

矩阵与线性方程组

帮助分析线性方程组的解的性质

矩阵的秩、行列式等概念可以帮助我们分析线性方程组解的存在性、唯一性和通解结构等性质。对于线性方程组$Ax = b$，通过比较系数矩阵A的秩$r(A)$和增广矩阵$\overline{A} = (A|b)$的秩$r(\overline{A})$，可以判断方程组是否有解：当$r(A) = r(\overline{A}) = n$时，方程组有唯一解；当$r(A) = r(\overline{A}) < n$时，方程组有无穷多解；当$r(A) \neq r(\overline{A})$时，方程组无解。此外，对于齐次线性方程组$Ax = 0$，矩阵$A$的秩与解空间的维数之间存在着密切的关系，解空间的维数等于$n - r(A)$，这有助于我们深入理解齐次线性方程组解的结构。

矩阵与线性方程组

- 《九章算术》：提出最早的线性方程组解法

今有上禾(上等稻)三秉(捆)，中禾二秉，下禾一秉，实(谷子) 三十九斗；上禾二秉，中禾三秉，下禾一秉，实三十四斗；上禾一秉，中禾二秉，下禾三秉，实二十六斗。问上、中、下禾一秉各几何？

今有上等稻3捆、中等稻2捆、下等稻1捆，共打出39斗米；有上等稻2捆、中等稻3捆、下等稻1捆，共打出34斗米；有上等稻1捆、中等稻2捆、下等稻3捆，共打出26斗米。问上等稻、中等稻、下等稻各1捆能打出多少斗米？

$$
\begin{aligned}
3x + 2y + z &= 39 \\
2x + 3y + z &= 34 \\
x + 2y + 3z &= 26.
\end{aligned}
$$

解法：算筹法（高斯消去法）

矩阵与线性方程组

科学与工程中的许多问题都可以通过数学建模转化成一个线性方程组

$$\begin{cases} \alpha_{11}x_1 + \alpha_{12}x_2 + \cdots + \alpha_{1n}x_n = b_1 \\ \alpha_{21}x_1 + \alpha_{22}x_2 + \cdots + \alpha_{2n}x_n = b_2 \\ \vdots \\ \alpha_{m1}x_1 + \alpha_{m2}x_2 + \cdots + \alpha_{mn}x_n = b_m \end{cases} \tag{2.1}$$

这是一个由m个方程n个未知量组成的线性方程组。

线性方程组(2. 1)可以表示成矩阵和向量的简洁的形式

$$Ax = b \tag{2.2}$$

矩阵与线性方程组

其中

$$A=\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}_{m\times n} \quad x=\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}_{n\times 1} \quad b=\begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}_{m\times 1}$$

A称为系数矩阵，b称为右端项，x称为要求的未知向量。

求解线性方程组从数学的角度看，就是要确定已知向量b在A的列向量上的线性表示系数，因为若把A表示成列向量的形式$A=[\alpha_1,\alpha_2,\dots,\alpha_n]$，则方程组$Ax=b$就是求$x_1,x_2,\dots,x_n$使得

$$\alpha_1 x_1+\alpha_2 x_2+\cdots+\alpha_n x_n=b \qquad (2.3)$$

矩阵与线性方程组

线性方程组$Ax = b$从工程问题的角度看，矩阵A往往是某个物理线性系统的符号表示，x表示该系统的输入激励（不可观测），b表示输出响应（可观测）。

于是线性方程组$Ax = b$的求解问题就可以叙述为：根据已知的线性系统参数和输出观察值，求未知的输入激励。

矩阵与线性方程组

对于线性方程组的系数矩阵A常有三种形式：

（1）$m>n$时常称矩阵A是“高矩阵”，相应的线性方程组为超定方程组(overdetermined systems)；

（2）$m=n$时矩阵A称为正方矩阵(square matrix)；

（3）$m<n$时矩阵A常称为“宽矩阵”，相应的线性方程组为亚定方程组(underdetermined systems)。

但方程组$Ax=b$中矩阵的上面三种形式，仅仅是形式上的差别，并不是影响其解(有解、无解等)的本质差别。

矩阵的基本运算

矩阵基本运算（如加法、乘法、求逆等）是《矩阵计算》这一领域的基石，对其学习和研究有着不可替代的重要意义，以下从理论基础、算法设计、问题求解等多个角度详细阐述：

奠定理论基础

- **理解矩阵空间结构**：矩阵加法和数乘运算满足一定的运算律，这些运算律定义了矩阵构成的线性空间结构。例如，矩阵加法的交换律（$A+B=B+A$）和结合律（$(A+B)+C=A+(B+C)$），以及数乘对加法的分配律（$k(A+B)=kA+kB$）等性质，是研究矩阵空间的维度、基和子空间等概念的基础。掌握这些基本运算，有助于深入理解矩阵空间的代数结构，为进一步学习矩阵的特征值、特征向量等高级理论提供支撑。
- **构建矩阵运算体系**：矩阵乘法是矩阵计算中最重要的运算之一，它不仅定义了矩阵之间的一种二元运算关系，还与线性变换的复合紧密相关。通过矩阵乘法，可以将线性变换表示为矩阵形式，从而方便地研究线性变换的性质和组合。而矩阵求逆运算则是矩阵乘法的逆运算，可逆矩阵的存在使得我们可以在矩阵运算中进行“除法”操作，进一步丰富了矩阵运算的体系。这些基本运算相互关联，共同构成了矩阵计算的理论框架。

支持算法设计与优化

- **设计高效算法的基础**：许多矩阵计算算法都是基于基本运算构建的。例如，高斯 - 约旦消元法用于求解线性方程组和求矩阵的逆，其核心步骤就是矩阵的初等行变换，而初等行变换本质上是矩阵加法和数乘运算的组合。又如，矩阵乘法的 Strassen 算法是一种用于提高矩阵乘法效率的分治算法，它通过巧妙地将矩阵分块并进行特定的矩阵运算组合，降低了矩阵乘法的时间复杂度。掌握矩阵基本运算，能够帮助我们理解这些算法的设计思路和原理，进而进行算法的改进和创新。

矩阵的基本运算

解决实际问题

- **建模与分析**：在科学和工程领域的众多实际问题中，常常需要将问题转化为矩阵形式，然后通过矩阵运算进行求解和分析。例如，在电路分析中，利用基尔霍夫定律建立的电路方程可以表示为矩阵方程，通过矩阵运算求解该方程可以得到电路中各节点的电压和各支路的电流。在图像处理中，图像可以表示为像素矩阵，通过对矩阵进行加法、乘法等运算，可以实现图像的滤波、增强等处理操作。掌握矩阵基本运算，能够帮助我们准确地建立问题的数学模型，并有效地进行求解和分析。
- **数据处理与挖掘**：在大数据时代，矩阵计算在数据处理和挖掘中发挥着重要作用。数据通常以矩阵的形式存储和表示，通过矩阵运算可以对数据进行降维、聚类、分类等处理。例如，主成分分析（PCA）是一种常用的数据降维方法，它通过对数据矩阵进行特征值分解和矩阵乘法运算，找到数据的主成分，从而实现数据的压缩和特征提取。掌握矩阵基本运算，有助于我们在数据处理和挖掘中更好地应用这些方法，从海量数据中提取有价值的信息。

矩阵的基本运算

1.1.4 向量运算

设 $a \in \mathbb{R}$, $\boldsymbol{x} \in \mathbb{R}^n$ 和 $\boldsymbol{y} \in \mathbb{R}^n$. 基本向量运算包括标量与向量的乘法,

$$\boldsymbol{z} = a\boldsymbol{x} \quad \Longrightarrow \quad z_i = a x_i ,$$

向量加法,

$$\boldsymbol{z} = \boldsymbol{x} + \boldsymbol{y} \quad \Longrightarrow \quad z_i = x_i + y_i ,$$

点积（或称内积）,

$$c = \boldsymbol{x}^{\mathrm{T}}\boldsymbol{y} \quad \Longrightarrow \quad c = \sum_{i=1}^{n} x_i y_i .$$

逐点向量运算也很有用，包括向量乘法,

$$\boldsymbol{z} = \boldsymbol{x} .* \boldsymbol{y} \quad \Longrightarrow \quad z_i = x_i y_i$$

和向量除法,

$$\boldsymbol{z} = \boldsymbol{x} ./ \boldsymbol{y} \quad \Longrightarrow \quad z_i = x_i / y_i .$$

矩阵的基本运算

1.1.2 矩阵运算

基本矩阵运算包括转置（$\mathbb{R}^{m\times n}\to\mathbb{R}^{n\times m}$），

$$\boldsymbol{C}=\boldsymbol{A}^{\mathrm{T}}\quad\Longrightarrow\quad c_{ij}=a_{ji},$$

加法（$\mathbb{R}^{m\times n}\times\mathbb{R}^{m\times n}\to\mathbb{R}^{m\times n}$），

$$\boldsymbol{C}=\boldsymbol{A}+\boldsymbol{B}\quad\Longrightarrow\quad c_{ij}=a_{ij}+b_{ij},$$

标量与矩阵的乘法（$\mathbb{R}\times\mathbb{R}^{m\times n}\to\mathbb{R}^{m\times n}$），

$$\boldsymbol{C}=\alpha\boldsymbol{A}\quad\Longrightarrow\quad c_{ij}=\alpha a_{ij},$$

矩阵与矩阵的乘法（$\mathbb{R}^{m\times p}\times\mathbb{R}^{p\times n}\to\mathbb{R}^{m\times n}$），

$$\boldsymbol{C}=\boldsymbol{A}\boldsymbol{B}\quad\Longrightarrow\quad c_{ij}=\sum_{k=1}^{p}a_{ik}b_{kj}.$$

逐点矩阵运算偶尔也有用，特别是逐点乘法（$\mathbb{R}^{m\times n}\times\mathbb{R}^{m\times n}\to\mathbb{R}^{m\times n}$），

$$\boldsymbol{C}=\boldsymbol{A}.*\boldsymbol{B}\quad\Longrightarrow\quad c_{ij}=a_{ij}b_{ij}$$

和逐点除法（$\mathbb{R}^{m\times n}\times\mathbb{R}^{m\times n}\to\mathbb{R}^{m\times n}$），

$$\boldsymbol{C}=\boldsymbol{A}./\boldsymbol{B}\quad\Longrightarrow\quad c_{ij}=a_{ij}/b_{ij}.$$

当然，为了使逐点除法有意义，“分母矩阵”必须都是非零元素.

矩阵的基本运算

1.1.10 矩阵与矩阵的乘法

考虑 2×2 矩阵与矩阵的乘法问题. 在点积形式下，每个元素都按点积计算:

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}\begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix} = \begin{bmatrix} 1\times 5 + 2\times 7 & 1\times 6 + 2\times 8 \\ 3\times 5 + 4\times 7 & 3\times 6 + 4\times 8 \end{bmatrix}.$$

在 saxpy 形式下，乘积的每列可看成是左边矩阵的列的线性组合:

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}\begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix} = \left[\, 5\begin{bmatrix} 1 \\ 3 \end{bmatrix} + 7\begin{bmatrix} 2 \\ 4 \end{bmatrix},\ 6\begin{bmatrix} 1 \\ 3 \end{bmatrix} + 8\begin{bmatrix} 2 \\ 4 \end{bmatrix} \,\right].$$

最后，在外积形式下，矩阵相乘可视为一组外积之和:

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}\begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix} = \begin{bmatrix} 1 \\ 3 \end{bmatrix}\begin{bmatrix} 5 & 6 \end{bmatrix} + \begin{bmatrix} 2 \\ 4 \end{bmatrix}\begin{bmatrix} 7 & 8 \end{bmatrix}.$$

请思考

这几种矩阵相乘的形式虽然在数学上等价，但事实证明，由于不同矩阵乘法的内存开销不同，计算表现上会有巨大差异. 对此，我们将在 1.5 节继续讨论. 在

矩阵的基本运算

矩阵与矩阵相乘 $m \times n$矩阵A与$r \times s$矩阵B的乘积AB只有当n=r时才存在，它是一个$m \times s$矩阵也就是说两个矩阵相乘只有当A的列数与B的行数相等时才可乘，并且不满足交换律，即$AB \neq BA$.

$$\left[(AB)_{ij}\right]_{m\times s}=\sum_{k=1}^{n}a_{ik}b_{kj}\,,i=1,2,\dots,m;j=1,2,\dots,s.$$

满足结合律，即$(AB)C = A(BC)$.

矩阵乘法更多的材料

参阅学习更多矩阵乘法的高效计算

矩阵的基本运算

$$\mathbf{B}\mathbf{x} = \mathbf{y}$$

$$b_{11}x_1 + b_{12}x_2 = y_1$$
$$b_{21}x_1 + b_{32}x_2 = y_2$$
$$b_{31}x_1 + b_{32}x_2 = y_3$$

$$\underbrace{\mathbf{A} \cdot \mathbf{B}}_{\mathbf{C}}\mathbf{x} = \mathbf{A}\mathbf{y}$$

$$c_{11}x_1 + c_{12}x_2 = a_{11}y_1 + a_{12}y_2 + a_{13}y_3$$
$$c_{21}x_1 + c_{22}x_2 = a_{21}y_1 + a_{22}y_2 + a_{23}y_3$$
$$c_{31}x_1 + c_{32}x_2 = a_{31}y_1 + a_{32}y_2 + a_{33}y_3$$

$$\mathbf{C}_{ij} = (\mathbf{AB})_{ij} = \sum_{k=1}^{n} a_{ik} b_{kj}$$

问题背景

矩阵链乘法问题是指给定一系列矩阵 $A_1, A_2, \cdots, A_n$，其中 A_i 的维度为 $p_{i-1} \times p_i$，我们要找出一种矩阵相乘的括号化方案，使得进行标量乘法的总次数最少。由于矩阵乘法满足结合律，不同的括号化方式会导致不同的计算复杂度。

具体例子

假设我们有 4 个矩阵 A_1、A_2、A_3、A_4，它们的维度分别如下：

- A_1：10×20
- A_2：20×30
- A_3：30×40
- A_4：40×50

这里对应的维度数组 $p = [10, 20, 30, 40, 50]$，其中 p_{i-1} 和 p_i 分别表示矩阵 A_i 的行数和列数。

一、状态定义

设

$$m[i][j]$$

表示计算矩阵链 $A_i A_{i+1} \cdots A_j$ 所需的**最少标量乘法次数。**

转移方程:

$$m[i][j] = \min_{i \le k < j} \{m[i][k] + m[k+1][j] + p_{i-1} p_k p_j\}$$

其中最后一项 $p_{i-1} p_k p_j$ 是把两部分结果矩阵相乘的代价。

二、先算长度为 2 的子问题

1. $m[1][2]$

$$A_1A_2 : 10 \times 20 \cdot 20 \times 30$$

代价:

$$10 \cdot 20 \cdot 30 = 6000$$

所以:

$$m[1][2] = 6000$$

2. $m[2][3]$

$$A_2A_3 : 20 \times 30 \cdot 30 \times 40$$

代价:

$$20 \cdot 30 \cdot 40 = 24000$$

所以:

$$m[2][3] = 24000$$

3. $m[3][4]$

$$A_3A_4 : 30 \times 40 \cdot 40 \times 50$$

代价:

$$30 \cdot 40 \cdot 50 = 60000$$

所以:

$$m[3][4] = 60000$$

三、算长度为 3 的子问题

1. $m[1][3]$

有两种分法:

分法一: $(A_1)(A_2A_3)$

$$0+24000+10\cdot20\cdot40=24000+8000=32000$$

分法二: $(A_1A_2)(A_3)$

$$6000+0+10\cdot30\cdot40=6000+12000=18000$$

取较小值:

$$m[1][3]=18000$$

最优括号化为:

$$(A_1A_2)A_3$$

2. $m[2][4]$

有两种分法:

分法一: $(A_2)(A_3A_4)$

$$0+60000+20\cdot30\cdot50=60000+30000=90000$$

分法二: $(A_2A_3)(A_4)$

$$24000+0+20\cdot40\cdot50=24000+40000=64000$$

取较小值:

$$m[2][4]=64000$$

最优括号化为:

$$(A_2A_3)A_4$$

四、算最终结果 $m[1][4]$

有三种分法:

1. $A_1 \mid (A_2A_3A_4)$

$$0 + 64000 + 10 \cdot 20 \cdot 50 = 64000 + 10000 = 74000$$

2. $(A_1A_2) \mid (A_3A_4)$

$$6000 + 60000 + 10 \cdot 30 \cdot 50 = 66000 + 15000 = 81000$$

3. $(A_1A_2A_3) \mid A_4$

$$18000 + 0 + 10 \cdot 40 \cdot 50 = 18000 + 20000 = 38000$$

取最小值:

$$m[1][4] = 38000$$

五、最终答案

最优括号化方案:

$$((A_1A_2)A_3)A_4$$

矩阵的基本运算

矩阵求逆(inverse)

对于$n \times n$矩阵A，如果存在$n \times n$矩阵B，使得$AB = BA = I_n$，则B就称为A的逆矩阵，记成A^{-1} 。这样对于线性方程组$Ax = b$的解就可以写成$x = A^{-1}b$.

对于矩阵的共轭、转置和共轭转置满足以下分配律：

$$(A+B)^* = A^* + B^* \quad (A+B)^T = A^T + B^T \quad (A+B)^H = A^H + B^H$$

对于矩阵的共轭、转置和共轭转置满足关系式：

$$(AB)^T = B^T A^T \qquad (AB)^H = B^H A^H$$

$$(AB)^{-1} = B^{-1} A^{-1}$$ （A,B为可逆的同阶正方阵）

矩阵的基本运算

2.1.4 Sherman-Morrison-Woodbury 公式

恒等式

$$B^{-1} = A^{-1} - B^{-1}(B-A)A^{-1} \tag{2.1.3}$$

表明了逆的变化和矩阵的变化的关系. 假设 $\boldsymbol{A} \in \mathbb{R}^{n\times n}$, $\boldsymbol{U}$ 和 $\boldsymbol{V}$ 都是 $n \times k$ 矩阵, Sherman-Morrison-Woodbury 公式给出 $(\boldsymbol{A}+\boldsymbol{U}\boldsymbol{V}^{\mathrm{T}})$ 的逆的一个简便表达式:

$$(\boldsymbol{A}+\boldsymbol{U}\boldsymbol{V}^{\mathrm{T}})^{-1} = \boldsymbol{A}^{-1} - \boldsymbol{A}^{-1}\boldsymbol{U}(\boldsymbol{I}+\boldsymbol{V}^{\mathrm{T}}\boldsymbol{A}^{-1}\boldsymbol{U})^{-1}\boldsymbol{V}^{\mathrm{T}}\boldsymbol{A}^{-1}. \tag{2.1.4}$$

矩阵的秩 k 的变化导致逆矩阵的秩 k 的变化. 在 (2.1.4) 中, 我们假定 $\boldsymbol{A}$ 和 $(\boldsymbol{I}+\boldsymbol{V}^{\mathrm{T}}\boldsymbol{A}^{-1}\boldsymbol{U})$ 都是非奇异的.

$k=1$ 的例子非常有用. 若 $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ 是非奇异的, $\boldsymbol{u}, \boldsymbol{v} \in \mathbb{R}^n$, 且 $\alpha = 1 + \boldsymbol{v}^{\mathrm{T}}\boldsymbol{A}^{-1}\boldsymbol{u} \neq 0$, 则

$$(\boldsymbol{A}+\boldsymbol{u}\boldsymbol{v}^{\mathrm{T}})^{-1} = \boldsymbol{A}^{-1} - \frac{1}{\alpha}\boldsymbol{A}^{-1}\boldsymbol{u}\boldsymbol{v}^{\mathrm{T}}\boldsymbol{A}^{-1}. \tag{2.1.5}$$

这称为 Sherman-Morrison 公式.

矩阵的性能指标

表 1.3.1　矩阵的性能指标

性能指标	描述的矩阵性能
二次型	矩阵的正定性与负定性
行列式	矩阵的奇异性
特征值	矩阵的奇异性、正定性和对角元素的结构
迹	矩阵对角元素之和、特征值之和
秩	行 (或列) 之间的线性无关性；线性方程组的适定性

矩阵的行列式

学习行列式有多方面的重要意义，主要体现在以下几个方面：

- **判定矩阵的可逆性：**行列式的值可以直接用于判断矩阵是否可逆。对于一个方阵行列式不为零，则矩阵可逆；若行列式为零，则矩阵不可逆。这为研究矩阵的性质和求解矩阵方程等问题提供了重要的依据。

- **为其他数学工具提供基础：**行列式的计算方法和性质为矩阵的特征值、特征向量等概念的研究提供了重要的基础。在求矩阵的特征值时，需要计算特征方程 $|\lambda I - A| = 0$ ，其中λ为特征值，I为单位矩阵，A为原矩阵。特征值和特征向量在许多领域都有广泛的应用，如数据分析中的主成分分析、动力系统的稳定性分析等

矩阵的行列式

- 一个$n \times n$正方矩阵A的行列式（determinant）记作$det(A)$或$|A|$.

对于标量a，有$\det(a) = a$.

- 矩阵A的行列式可由递推得出：

$$\det(A) = \sum_{j=1}^{n} (-1)^{i+j} a_{ij} \det(A_{ij})$$

其中A_{ij}为矩阵A删去第i行和第j列后的$(n-1) \times (n-1)$子矩阵。

矩阵的行列式

- 定义：行列式不等于零的矩阵称为非奇异矩阵。

- 行列式的性质：

（1） $\det(I) = 1$；

（2） $\det(A) = \det(A^T)$, 但是 $\det(A^H) = [\det(A^T)]^* = \det(A)^*$；

（3） $\det(AB) = \det(A)\det(B)$；

（4） $\det(cA) = c^n\det(A)$；

（5） 若A非奇异，$\det(A^{-1}) = 1/\det(A)$

矩阵的行列式

行列式等于所有特征值的乘积

设A是n阶方阵，$\lambda_1, \lambda_2, \cdots, \lambda_n$是$A$的$n$个特征值（重根按重数计算），则方阵$A$的行列式$|A|$等于其所有特征值的乘积，即$|A| = \lambda_1\lambda_2\cdots\lambda_n$。这一关系可以通过矩阵的特征方程和行列式的性质来证明。

- **Sylvester's Determinant Theorem**

  for A, an $m \times n$ matrix, and B, an $n \times m$ matrix

$$\det(I_m + AB) = \det(I_n + BA)$$

- 推论：
  - For the case of column vector c and r, each with m components

$$\det(I_m + cr^T) = 1 + r^T c$$

  - For any invertible $m \times m$ matrix X

$$\det(X + AB) = \det(X)\det(I_n + BX^{-1}A)$$

矩阵的二次型

在矩阵计算中，学习二次型具有重要意义，体现在理论研究、实际应用和优化问题等多个方面，具体如下：

- **深入理解矩阵性质：** 二次型与对称矩阵存在一一对应的关系，通过研究二次型可以更深入地理解对称矩阵的性质和结构。例如，正定二次型对应的矩阵是正定矩阵，负定二次型对应负定矩阵等。**利用二次型的性质可以对矩阵进行分类和刻画，为矩阵理论的研究提供了新的视角和方法。**

矩阵的二次型

优化问题方面

- **求解极值问题：许多优化问题可以转化为二次型的极值问题**。例如，在无约束优化问题中，目标函数如果是二次函数，那么可以通过对二次型的矩阵进行分析，利用矩阵的特征值和特征向量等知识来求解函数的极值点和极值；在约束优化问题中，也可以通过拉格朗日乘数法等方法将约束问题转化为无约束问题，再利用二次型的性质进行求解。这种将优化问题转化为二次型问题的方法在数学规划、机器学习等领域有着广泛的应用，为解决各种实际优化问题提供了有效的手段。

- **凸优化问题：**二次型在凸优化理论中具有重要地位。**如果一个二次型是正定的，那么它对应的函数是凸函数，凸优化问题具有良好的性质和有效的求解算法。**许多实际问题，如资源分配、网络规划等，都可以建模为凸优化问题，通过将问题转化为二次型的形式，利用凸优化的理论和算法来寻找最优解，提高问题的求解效率和准确性。

矩阵的二次型

矩阵的二次型(Quadratic form)

- 任意一个实对称矩阵或复共轭矩阵(即Hermitian)矩阵A的二次型定义为$x^H Ax$，其中x可以是任意的非零复向量。

- 矩阵的二次型$x^H Ax$取实数，可以同零比较大小。

矩阵的二次型

例：设待设计的线性系统用$m \times n$矩阵A建模，其输入向量为x，输出信号向量$y = Ax$，为了抑制噪声或者干扰，线性系统的设计常采用最大输出能量准则：使输出能量

$$J(x) = \|y\|^2 = y^H y = (Ax)^H (Ax) = x^H A^H A x$$

最大化。此时目标函数$J(x) = x^H B x$即为二次型函数，其中$B = A^H A$为$n \times n$ Hermitian矩阵。

- 通常将大于零的二次型$x^H A x$称为正定二次型，与之对应的$n \times n$ Hermitian矩阵称为正定矩阵。类似的可由二次型$x^H A x$定义Hermitian矩阵的半正定性、负定性和半负定性。

矩阵的二次型

定义4.1 一个$n \times n$的复共轭对称矩阵A称为：

(1) 正定矩阵(positive definete)：若 $\forall x \neq 0$， 二次型 $x^H A x > 0$，记作A>0；

(2) 半正定矩阵(positive semi-definite, PSD)：若对$\forall x \neq 0$，二次型$x^H A x \geq 0$，记作A≥0；

(3) 负定矩阵：若对$\forall x \neq 0$，二次型$x^H A x < 0$，记作A<0；

(4) 半负定矩阵：若对$\forall x \neq 0$，二次型$x^H A x \leq 0$，记作A≤0；

(5) 不定矩阵：若二次型$x^H A x$既可取正值，也可取负值。

代数与矩阵基础

- 向量，矩阵
- 线性空间，线性子空间
  - 零空间，像空间
- 线性相关，线性无关
  - 空间的维数与基底
- 矩阵的基本运算
  - 转置，共轭转置
  - 相乘，求逆
- 矩阵的数值特征
  - 行列式，二次型，特征值，trace，rank
- 内积与范数
  - 向量范数
  - 矩阵范数