Linear Algebra

Matrices and Vectors. First Introduction, Basic Operations.

Gleb Karpov

MDS FCS HSE

Matrix

- **Definition**

  A **matrix** is an ordered array of numbers arranged in n rows and m columns.

$$A_{n\times m} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & a_{22} & \cdots & a_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nm} \end{bmatrix}$$

Matrix

- **Definition**

  A **matrix** is an ordered array of numbers arranged in n rows and m columns.

  $$A_{n\times m} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & a_{22} & \cdots & a_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nm} \end{bmatrix}$$

- We usually denote it as $A_{n\times m}$ or, to emphasize the nature of numbers in the matrix, we write $A \in \mathbb{R}^{n\times m}$.

Matrix

- **Definition**

  A **matrix** is an ordered array of numbers arranged in n rows and m columns.

  $$A_{n \times m} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & a_{22} & \cdots & a_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nm} \end{bmatrix}$$

- We usually denote it as $A_{n \times m}$ or, to emphasize the nature of numbers in the matrix, we write $A \in \mathbb{R}^{n \times m}$.
- If $n = m$, the matrix is called square; if $n \neq m$, it's called rectangular

Basic operations: transpose

Definition

Transpose of a matrix is an operation where rows and columns are swapped. If $A \in \mathbb{R}^{n \times m}$, then $B = A^T \in \mathbb{R}^{m \times n}$, where $b_{ij} = a_{ji}$

$$\underset{A_{2\times 3}}{\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{bmatrix}} \xrightarrow{A^T} \underset{B_{3\times 2} = A^T}{\begin{bmatrix} a_{11} & a_{21} \\ a_{12} & a_{22} \\ a_{13} & a_{23} \end{bmatrix}}$$

Basic operations: matrix addition

-

> ℹ Definition
>
> **Matrix addition** is only possible for matrices of the same size. The result is obtained by adding corresponding elements. If $A, B \in \mathbb{R}^{n \times m}$, then $C = A + B$, where $c_{ij} = a_{ij} + b_{ij}$

Example:

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} + \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix} = \begin{bmatrix} 1+5 & 2+6 \\ 3+7 & 4+8 \end{bmatrix} = \begin{bmatrix} 6 & 8 \\ 10 & 12 \end{bmatrix}$$

Basic operations: scalar multiplication

- **Definition**

  Scalar multiplication - each element of the matrix is multiplied by the given number. If $A \in \mathbb{R}^{n\times m}$ and $\alpha \in \mathbb{R}$, then $C = \alpha A$, where $c_{ij} = \alpha \cdot a_{ij}$

Example:

$$3 \cdot \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} = \begin{bmatrix} 3\cdot 1 & 3\cdot 2 \\ 3\cdot 3 & 3\cdot 4 \end{bmatrix} = \begin{bmatrix} 3 & 6 \\ 9 & 12 \end{bmatrix}$$

Another example - Linear combination:

$$2\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + 3\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix} + \begin{bmatrix} 3 & 3 \\ 3 & 3 \end{bmatrix} = \begin{bmatrix} 5 & 3 \\ 3 & 5 \end{bmatrix}$$

Vector

In the simplest representation, we treat a **vector** as a special case of a matrix:

Column vector

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \in \mathbb{R}^{n\times 1}$$

Dimension: $n \times 1$ (matrix with one column)

Row vector

$$\mathbf{y}^T = \begin{bmatrix} y_1 & y_2 & \cdots & y_m \end{bmatrix} \in \mathbb{R}^{1\times m}$$

Dimension: $1 \times m$ (matrix with one row)

Notation

- **Vectors:** usually denoted by lowercase letters x, v or $\mathbf{u}$

Vector

In the simplest representation, we treat a **vector** as a special case of a matrix:

Column vector

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \in \mathbb{R}^{n\times 1}$$

Dimension: $n \times 1$ (matrix with one column)

Row vector

$$\mathbf{y}^T = \begin{bmatrix} y_1 & y_2 & \cdots & y_m \end{bmatrix} \in \mathbb{R}^{1\times m}$$

Dimension: $1 \times m$ (matrix with one row)

Notation

- **Vectors:** usually denoted by lowercase letters x, v or $\mathbf{u}$
- **Matrices:** usually denoted by uppercase letters A, B, C

Vector

In the simplest representation, we treat a **vector** as a special case of a matrix:

Column vector

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \in \mathbb{R}^{n \times 1}$$

Dimension: $n \times 1$ (matrix with one column)

Row vector

$$\mathbf{y}^T = \begin{bmatrix} y_1 & y_2 & \cdots & y_m \end{bmatrix} \in \mathbb{R}^{1 \times m}$$

Dimension: $1 \times m$ (matrix with one row)

Notation

- **Vectors:** usually denoted by lowercase letters x, v or $\mathbf{u}$
- **Matrices:** usually denoted by uppercase letters A, B, C
- **By default:** vector is considered a **column vector**

Vector

In the simplest representation, we treat a **vector** as a special case of a matrix:

Column vector

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \in \mathbb{R}^{n\times 1}$$

Dimension: $n \times 1$ (matrix with one column)

Row vector

$$\mathbf{y}^T = \begin{bmatrix} y_1 & y_2 & \cdots & y_m \end{bmatrix} \in \mathbb{R}^{1\times m}$$

Dimension: $1 \times m$ (matrix with one row)

Notation

- **Vectors:** usually denoted by lowercase letters x, v or $\mathbf{u}$
- **Matrices:** usually denoted by uppercase letters A, B, C
- **By default:** vector is considered a **column vector**
- **Transpose:** $\mathbf{x}^\top$ converts column to row

Matrix-by-vector multiplication (matvec)

Standard approach

$$\underbrace{\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1,m-1} & a_{1m} \\ a_{21} & a_{22} & \cdots & a_{2,m-1} & a_{2m} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{n,m-1} & a_{nm} \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{m-1} \\ x_m \end{bmatrix}}_{\mathbf{x}} = \underbrace{\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}}_{\mathbf{y}}$$

$$y_1 = a_{11}x_1 + a_{12}x_2 + \cdots + a_{1,m-1}x_{m-1} + a_{1m}x_m$$

$$\boxed{y_j = \sum_{k=1}^{m} a_{jk}x_k}$$

Matrix-by-vector multiplication (matvec)

Computational complexity and a brief look at parallel computing

Recall the general formula:

$$\boxed{y_j = \sum_{k=1}^{m} a_{jk} x_k, \quad j = 1, 2, \dots, n}$$

Operations analysis:

- For computing one element y_j: m multiplications (each $a_{jk} \cdot x_k$), $m - 1$ additions (summing m products)
- For the entire vector $\mathbf{y}$ (n elements): $n \cdot (2m - 1)$ total operations.

Time complexity:

- For square matrix $n \times n$: $\mathcal{O}(2n^2 - n) = \mathcal{O}(n^2)$
- For rectangular matrix $n \times m$: $\mathcal{O}(nm)$

Natural parallelism

Computing each element y_j is **independent** of other elements!

- **Row-wise parallelization:** each processor computes its own y_j

Matrix-by-vector multiplication (matvec)

Guru approach

$$\underbrace{\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}}_{\mathbf{x}} = \underbrace{\begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix}}_{\mathbf{y}}$$

Matrix-by-vector multiplication (matvec)

Guru approach

$$\underbrace{\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}}_{\mathbf{x}} = \underbrace{\begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix}}_{\mathbf{y}}$$

$$y_1 = a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + a_{14}x_4$$

Matrix-by-vector multiplication (matvec)

Guru approach

$$\underbrace{\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}}_{\mathbf{x}} = \underbrace{\begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix}}_{\mathbf{y}}$$

$$y_1 = a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + a_{14}x_4$$

$$y_2 = a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + a_{24}x_4$$

Matrix-by-vector multiplication (matvec)

Guru approach

$$\underbrace{\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}}_{\mathbf{x}} = \underbrace{\begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix}}_{\mathbf{y}}$$

$$\begin{aligned} y_1 &= a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + a_{14}x_4 \\ y_2 &= a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + a_{24}x_4 \\ y_3 &= a_{31}x_1 + a_{32}x_2 + a_{33}x_3 + a_{34}x_4 \end{aligned}$$

Matrix-by-vector multiplication (matvec)

Guru approach

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix}$$

$A \qquad \mathbf{x} \qquad \mathbf{y}$

$$\mathbf{y} = x_1 \begin{bmatrix} a_{11} \\ a_{21} \\ a_{31} \end{bmatrix} + x_2 \begin{bmatrix} a_{12} \\ a_{22} \\ a_{32} \end{bmatrix} + x_3 \begin{bmatrix} a_{13} \\ a_{23} \\ a_{33} \end{bmatrix} + x_4 \begin{bmatrix} a_{14} \\ a_{24} \\ a_{34} \end{bmatrix}$$

Matrix-by-vector product

Visual comparison

$$\underbrace{\begin{bmatrix} 3 & 1 \\ 2 & 4 \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} 2 \\ 1 \end{bmatrix}}_{\mathbf{x}} = \underbrace{\begin{bmatrix} 7 \\ 8 \end{bmatrix}}_{\mathbf{y}}$$

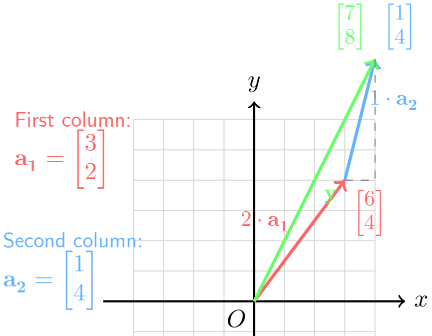
Traditional View: Row-wise

Calculations:

- $y_1 = 3 \cdot 2 + 1 \cdot 1 = 7$
- $y_2 = 2 \cdot 2 + 4 \cdot 1 = 8$

Guru View: Column Linear Combination

Linear combination:



Matrix-by-matrix multiplication (matmul, General Matrix Multiplication - GEMM)

Standard approach

$$
\begin{array}{ccccc}
 & & j\text{-th column} & & \\
i\text{-th row} \rightarrow &
\underset{A}{\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1k} \\ \vdots & \vdots & \ddots & \vdots \\ a_{i1} & a_{i2} & \cdots & a_{ik} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nk} \end{bmatrix}} &
\underset{B}{\begin{bmatrix} b_{11} & \cdots & b_{1j} & \cdots & b_{1m} \\ b_{21} & \cdots & b_{2j} & \cdots & b_{2m} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ b_{k1} & \cdots & b_{kj} & \cdots & b_{km} \end{bmatrix}} &
= &
\underset{C}{\begin{bmatrix} c_{11} & \cdots & c_{1j} & \cdots & c_{1m} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ c_{i1} & \cdots & c_{ij} & \cdots & c_{im} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ c_{n1} & \cdots & c_{nj} & \cdots & c_{nm} \end{bmatrix}}
\end{array}
$$

$$c_{ij} = a_{i1} \cdot b_{1j} + a_{i2} \cdot b_{2j} + \cdots + a_{ik} \cdot b_{kj}$$

$$\boxed{c_{ij} = \sum_{l=1}^{k} a_{il} b_{lj}}$$

Matrix multiplication (matmul, General Matrix Multiplication - GEMM)

Column-wise approach

$$\underbrace{\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1k} \\ \vdots & \vdots & \ddots & \vdots \\ a_{i1} & a_{i2} & \cdots & a_{ik} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nk} \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} b_{11} & b_{12} & b_{13} & \cdots & b_{1m} \\ b_{21} & b_{22} & b_{23} & \cdots & b_{2m} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ b_{k1} & b_{k2} & b_{k3} & \cdots & b_{km} \end{bmatrix}}_{B} = \underbrace{\begin{bmatrix} c_{11} & c_{12} & c_{13} & \cdots & c_{1m} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ c_{i1} & c_{i2} & c_{i3} & \cdots & c_{im} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ c_{n1} & c_{n2} & c_{n3} & \cdots & c_{nm} \end{bmatrix}}_{C}$$

Each column of $C = A\times$ corresponding column of B

$$\boxed{\mathbf{C_j} = A\mathbf{B_j}, \quad j = 1, 2, \ldots, m}$$

Matrix multiplication (matmul, General Matrix Multiplication - GEMM)

Properties of matrix multiplication

1. **Associativity:**

$$(AB)C = A(BC)$$

Matrix multiplication (matmul, General Matrix Multiplication - GEMM)

Properties of matrix multiplication

1. **Associativity:**

$$(AB)C = A(BC)$$

2. **Distributivity:**

$$A(B + C) = AB + AC$$

$$(A + B)C = AC + BC$$

Matrix multiplication (matmul, General Matrix Multiplication - GEMM)

Properties of matrix multiplication

1. **Associativity:**

$$(AB)C = A(BC)$$

2. **Distributivity:**

$$A(B + C) = AB + AC$$

$$(A + B)C = AC + BC$$

3. **Scalar multiplication:**

$$\alpha(AB) = (\alpha A)B = A(\alpha B)$$

Matrix multiplication (matmul, General Matrix Multiplication - GEMM)

Important limitations

4. **Non-commutativity:**

$$AB \neq BA \quad \text{(in general)}$$

Example: For 2×2 matrices:

$$\begin{bmatrix} 1 & 2 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} = \begin{bmatrix} 3 & 2 \\ 1 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 2 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 2 \\ 1 & 3 \end{bmatrix}$$

Matrix multiplication (matmul, General Matrix Multiplication - GEMM)

Important limitations

4. **Non-commutativity:**

$$AB \neq BA \quad \text{(in general)}$$

   Example: For 2×2 matrices:

$$\begin{bmatrix} 1 & 2 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} = \begin{bmatrix} 3 & 2 \\ 1 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 2 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 2 \\ 1 & 3 \end{bmatrix}$$

5. **Transpose of product:**

$$(AB)^T = B^T A^T$$

> **Уведомление**
>
> Note the **reversed order** of matrices when transposing!

Matrix multiplication (matmul, General Matrix Multiplication - GEMM)

Computational complexity and parallelization

Recall the general formula:

$$\boxed{c_{ij} = \sum_{l=1}^{k} a_{il} b_{lj}, \quad i = 1, 2, \dots, n, \quad j = 1, 2, \dots, m}$$

Operations analysis for matrices $A_{n\times k}$ and $B_{k\times m}$:

- For computing one element c_{ij}: k multiplications, $k-1$ additions
- Total number of operations: $n \times m \times (2k-1)$

Time complexity:

- For square matrices $n \times n$: $\mathcal{O}(2n^3 - n^2) = \mathcal{O}(n^3)$
- For rectangular matrices: $\mathcal{O}(nmk)$

Natural parallelism

- **By elements:** each c_{ij} is computed independently

Matrix multiplication (matmul, General Matrix Multiplication - GEMM)

Computational complexity and parallelization

Recall the general formula:

$$\boxed{c_{ij} = \sum_{l=1}^{k} a_{il} b_{lj}, \quad i = 1, 2, \ldots, n, \quad j = 1, 2, \ldots, m}$$

Operations analysis for matrices $A_{n \times k}$ and $B_{k \times m}$:

- For computing one element c_{ij}: k multiplications, $k - 1$ additions
- Total number of operations: $n \times m \times (2k - 1)$

Time complexity:

- For square matrices $n \times n$: $\mathcal{O}(2n^3 - n^2) = \mathcal{O}(n^3)$
- For rectangular matrices: $\mathcal{O}(nmk)$

Natural parallelism

- **By elements:** each c_{ij} is computed independently
- **By rows:** each processor handles its rows of matrix A

Matrix multiplication (matmul, General Matrix Multiplication - GEMM)

Computational complexity and parallelization

Recall the general formula:

$$\boxed{c_{ij} = \sum_{l=1}^{k} a_{il} b_{lj}, \quad i = 1, 2, \ldots, n, \quad j = 1, 2, \ldots, m}$$

Operations analysis for matrices $A_{n \times k}$ and $B_{k \times m}$:

- For computing one element c_{ij}: k multiplications, $k-1$ additions
- Total number of operations: $n \times m \times (2k-1)$

Time complexity:

- For square matrices $n \times n$: $\mathcal{O}(2n^3 - n^2) = \mathcal{O}(n^3)$
- For rectangular matrices: $\mathcal{O}(nmk)$

Natural parallelism

- **By elements:** each c_{ij} is computed independently
- **By rows:** each processor handles its rows of matrix A
- **By columns:** each processor handles its columns of matrix B

Matrix multiplication (matmul, General Matrix Multiplication - GEMM)

Computational complexity and parallelization

Recall the general formula:

$$\boxed{c_{ij} = \sum_{l=1}^{k} a_{il} b_{lj}, \quad i = 1, 2, \dots, n, \quad j = 1, 2, \dots, m}$$

Operations analysis for matrices $A_{n \times k}$ and $B_{k \times m}$:

- For computing one element c_{ij}: k multiplications, $k - 1$ additions
- Total number of operations: $n \times m \times (2k - 1)$

Time complexity:

- For square matrices $n \times n$: $\mathcal{O}(2n^3 - n^2) = \mathcal{O}(n^3)$
- For rectangular matrices: $\mathcal{O}(nmk)$

Natural parallelism

- **By elements:** each c_{ij} is computed independently
- **By rows:** each processor handles its rows of matrix A
- **By columns:** each processor handles its columns of matrix B
- **Block approach:** dividing matrices into blocks for efficient cache usage

Example. Simple but important idea about matrix computations.

Suppose you have the following expression

$$b = A_1 A_2 A_3 x,$$

where $A_1, A_2, A_3 \in \mathbb{R}^{3\times 3}$ are random square dense (fully filled with numbers) matrices, and $x \in \mathbb{R}^3$ is a vector. You need to compute b.

Which approach is best to use?

1. $A_1 A_2 A_3 x$ (left to right)
2. $\left(A_1 \left(A_2 \left(A_3 x\right)\right)\right)$ (right to left)
3. It doesn't matter
4. The results of the first two options will not be the same.

Check the attached .ipynb file in the repository.