3 The Magical World of Matrices: Building Blocks for Machine Learning

3.1 Discovering Matrices: The Playground of Numbers

3.1.1 What Is a Matrix? Rows, Columns, and More

As shown in Figure 1, a matrix $\boldsymbol{A}$ is a rectangular array of numbers arranged in rows and columns. Its shape is denoted as $m \times p$, where m represents the number of rows and p represents the number of columns.

Sometimes this is written as $\boldsymbol{A}_{m \times p}$ to make the dimensions explicit. The total number of elements (also called **entries** or **components**) in the matrix is simply the product $m \times p$.

Each element of the matrix is indexed by its row and column position. The element in the i-th row and j-th column is written as $a_{i,j}$.

For example, $a_{1,1}$ refers to the element in the first row and first column, while $a_{2,3}$ refers to the element in the second row and third column.

In fact, row vectors and column vectors discussed in the previous section are special cases of matrices — a row vector has only one row, and a column vector has only one column.

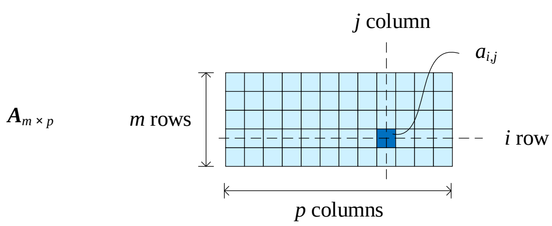


Figure 1. The shape of matrix $\boldsymbol{A}$

3.1.2 Matrices as Row and Column Collections

A matrix can be viewed as a collection of **row vectors** stacked on top of one another, as illustrated in Figure 2. For instance, we can write the matrix $\boldsymbol{A}$ as a sequence of its rows:

$$\boldsymbol{A}_{m\times p} = \begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} & \cdots & a_{1,p} \\ a_{2,1} & a_{2,2} & a_{2,3} & \cdots & a_{2,p} \\ a_{3,1} & a_{3,2} & a_{3,3} & \cdots & a_{3,p} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{m,1} & a_{m,2} & a_{m,3} & \cdots & a_{m,p} \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} & \cdots & a_{1,p} \end{bmatrix} \\ \begin{bmatrix} a_{2,1} & a_{2,2} & a_{2,3} & \cdots & a_{2,p} \end{bmatrix} \\ \begin{bmatrix} a_{3,1} & a_{3,2} & a_{3,3} & \cdots & a_{3,p} \end{bmatrix} \\ \vdots \\ \begin{bmatrix} a_{m,1} & a_{m,2} & a_{m,3} & \cdots & a_{m,p} \end{bmatrix} \end{bmatrix} = \begin{bmatrix} \boldsymbol{a}^{(1)} \\ \boldsymbol{a}^{(2)} \\ \boldsymbol{a}^{(3)} \\ \vdots \\ \boldsymbol{a}^{(m)} \end{bmatrix} \tag{1}$$

where $\boldsymbol{a}^{(i)}$ represents the i-th row vector of $\boldsymbol{A}$.

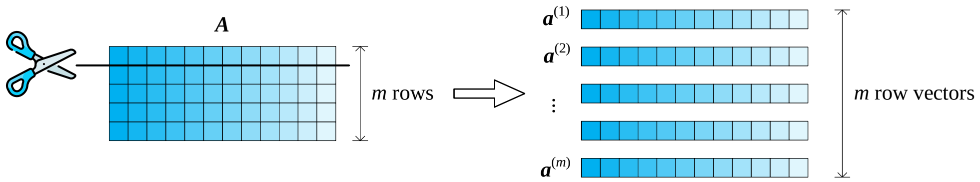


Figure 2. Viewing a matrix as a collection of row vectors

Alternatively, a matrix can be interpreted as a collection of **column vectors** arranged side by side (see Figure 3). We can express $\boldsymbol{A}$ as:

$$\boldsymbol{A}_{m\times p}=\begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} & \cdots & a_{1,p} \\ a_{2,1} & a_{2,2} & a_{2,3} & \cdots & a_{2,p} \\ a_{3,1} & a_{3,2} & a_{3,3} & \cdots & a_{3,p} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{m,1} & a_{m,2} & a_{m,3} & \cdots & a_{m,p} \end{bmatrix}=\left[\begin{bmatrix} a_{1,1} \\ a_{2,1} \\ a_{3,1} \\ \vdots \\ a_{m,1} \end{bmatrix} \begin{bmatrix} a_{1,2} \\ a_{2,2} \\ a_{3,2} \\ \vdots \\ a_{m,2} \end{bmatrix} \begin{bmatrix} a_{1,3} \\ a_{2,3} \\ a_{3,3} \\ \vdots \\ a_{m,3} \end{bmatrix} \cdots \begin{bmatrix} a_{1,p} \\ a_{2,p} \\ a_{3,p} \\ \vdots \\ a_{m,p} \end{bmatrix}\right]=\begin{bmatrix} \boldsymbol{a}_1 & \boldsymbol{a}_2 & \boldsymbol{a}_3 & \cdots & \boldsymbol{a}_p \end{bmatrix} \quad (2)$$

where $\boldsymbol{a}_j$ represents the j-th column vector of $\boldsymbol{A}$.

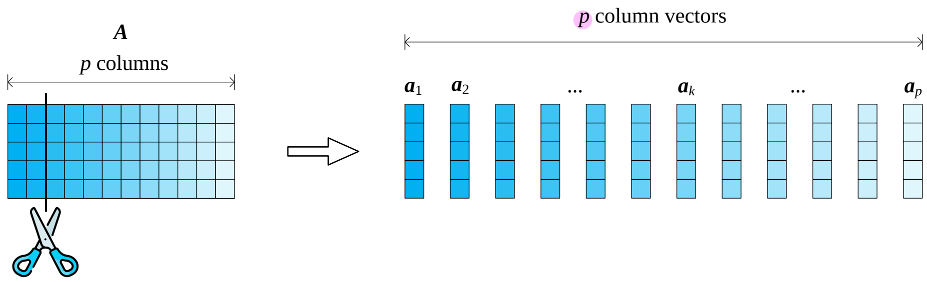


Figure 3. Viewing a matrix as a collection of column vectors

3.1.3 The Importance of Order: Why Position Matters

For a given matrix, the order of its rows and columns is fixed. Unless a mathematical operation explicitly rearranges them, the position of each element within the matrix remains unchanged.

In other words, a matrix is not just a collection of numbers—it is an **ordered structure**, where the arrangement of rows and columns carries important meaning. Each element $a_{i,j}$ is uniquely defined by its position, and changing that order would result in an entirely different matrix. For example, swapping two rows or two columns changes the relationships among data points or features.

This fixed structure is crucial in linear algebra and machine learning because many operations—such as matrix multiplication, transposition, and row transformations—depend strictly on the order of rows and columns. Understanding this helps ensure that when we manipulate matrices, we preserve the intended meaning of the data they represent.

3.2 Diagonals, Transposes, and Reflections

3.2.1 The Main Diagonal: Your Matrix's Guiding Line

Every matrix contains a special set of elements known as the **main diagonal elements** (also called *diagonal entries*). These are the elements whose row and column indices are the same. For example, in Figure 4, the blue elements represent the diagonal entries of matrix $\boldsymbol{A}$: $a_{1,1}$, $a_{2,2}$, $a_{3,3}$, and so on.

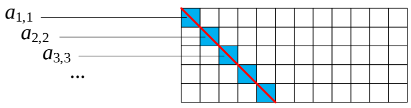


Figure 4. The main diagonal elements of matrix $\boldsymbol{A}$

Connecting these diagonal elements forms a line called the **main diagonal** (also known as the *principal diagonal* or *leading diagonal*). Geometrically, if we imagine each matrix element as a square cell of equal size, the main diagonal is the 45-degree line that starts from the top-left cell and extends down toward the bottom-right corner. This diagonal passes through every cell whose row and column indices are equal—like a neat staircase cutting across the grid of the matrix.

The concept of the main diagonal applies to matrices of any shape, whether tall, wide, or square. Even in special cases such as row vectors or column vectors, a main diagonal can still be identified, though it may contain only a single element. Figure 5 illustrates this idea, showing the red main diagonal across matrices of different dimensions.

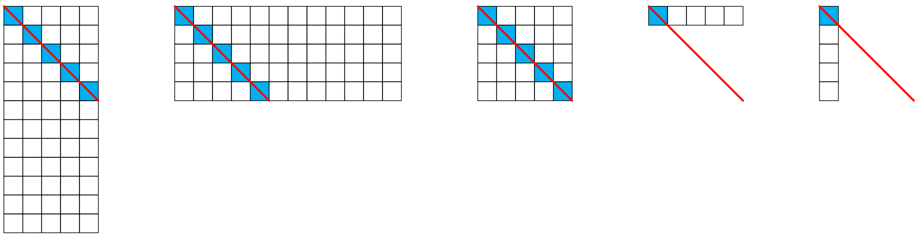


Figure 5. The main diagonals in matrices of various shapes

3.2.2 Flipping Matrices: Understanding the Transpose

The **transpose of a matrix** is an operation that flips the matrix over its main diagonal, effectively turning rows into columns and columns into rows. In mathematical terms, it is a transformation from one matrix to another — a **matrix-to-matrix operation**. The transpose of a matrix $\boldsymbol{A}$ is denoted by $\boldsymbol{A}^{\mathrm{T}}$.

As shown in Figure 6, if matrix $\boldsymbol{A}$ has a shape of $m \times p$, then its transpose $\boldsymbol{A}^{\mathrm{T}}$ will have a shape of $p \times m$.

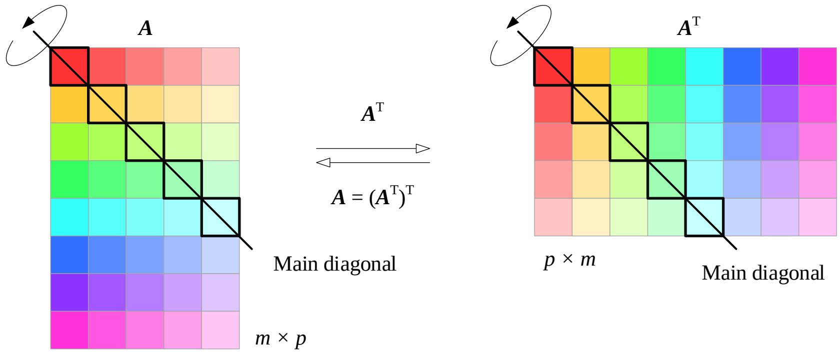


Figure 6. Transposing a matrix as reflection across the main diagonal

Each element of $\boldsymbol{A}$, originally located at row i and column j, moves to row j and column i in the transposed matrix. Symbolically,

$$(\boldsymbol{A}^{\mathrm{T}})_{j,i} = \boldsymbol{A}_{i,j} \tag{3}$$

An important property of this operation is that the elements lying on the **main diagonal** (shown in the black boxes in Figure 6) remain unchanged after transposition. This happens because diagonal elements have identical row and column indices — for example, $a_{1,1}$, $a_{2,2}$, $a_{3,3}$ all satisfy $i = j$, so their positions do not change when rows and columns are swapped.

Visually, we can imagine each element of the matrix as a square cell in a grid. Transposing the matrix is like reflecting that grid across its main diagonal — the top-right and bottom-left parts swap places, while the diagonal itself stays fixed. This geometric interpretation helps us see that the transpose operation preserves the matrix's internal structure but changes its orientation.

3.3 Matrix Shapes and Special Types

3.3.1 Rectangular, Tall, and Wide Matrices

Matrices come in a variety of shapes, each with its own characteristics and applications. A rectangular matrix is one in which the number of rows and columns are not equal. Rectangular matrices can take two main forms: tall and wide. A tall matrix has many more rows than columns, whereas a wide matrix has many more columns than rows. These two types are closely related: transposing a tall matrix produces a wide matrix, and vice versa, as illustrated in Figure 7.

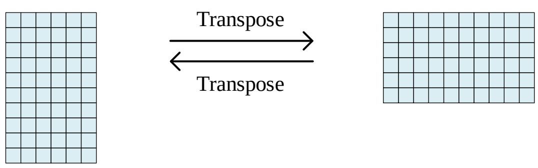


Figure 7. Tall and wide matrices are related through transposition

3.3.2 Square Matrices and Symmetry

A **square matrix** is a special and particularly important form of matrix where the number of rows equals the number of columns, giving it the shape $n \times n$. Square matrices play a central role in linear algebra and machine learning because they support many key operations, such as determinants, eigenvalues, and linear transformations.

If a $n \times n$ square matrix $\boldsymbol{A}$ satisfies

$$\boldsymbol{A} = \boldsymbol{A}^{\mathrm{T}} \tag{4}$$

meaning it is equal to its transpose, it is called a **symmetric matrix**. Symmetric matrices have many useful properties, such as having real eigenvalues and orthogonal eigenvectors, which are important in applications like principal component analysis (PCA).

Later in this book, we will introduce the **covariance matrix**, which is a special type of **real symmetric matrix**. In machine learning, the covariance matrix plays a crucial role in understanding the relationships between different features in a dataset. Each entry of the covariance matrix measures how much two variables vary together, while the diagonal elements represent the variance of individual features.

3.3.3 Diagonal and Identity Matrices

A **diagonal matrix** is another important type of matrix in which all non-zero elements are confined to the main diagonal, and all off-diagonal elements are zero. Diagonal matrices can be either square (square diagonal matrices) or rectangular (rectangular diagonal matrices). The elements on the main diagonal may be zero or nonzero. Figure 8 shows examples of diagonal matrices with different shapes.

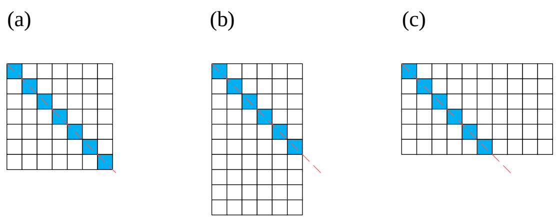


Figure 8. Diagonal matrices in various shapes

A particularly notable square diagonal matrix is the **identity matrix**, denoted as $\boldsymbol{I}$. In an identity matrix, all diagonal elements are 1, and all off-diagonal elements are 0,

$$\boldsymbol{I}_{n\times n} = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix} \tag{5}$$

The identity matrix serves as the multiplicative identity in matrix algebra: multiplying any compatible matrix by III leaves it unchanged.

3.3.4 Triangular Matrices

Another special type of square matrix is the triangular matrix, which comes in two varieties. An upper triangular matrix has all elements below the main diagonal equal to zero, for instance

$$\boldsymbol{U}_{n\times n} = \begin{bmatrix} u_{1,1} & u_{1,2} & \cdots & u_{1,n} \\ 0 & u_{2,2} & \cdots & u_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & u_{n,n} \end{bmatrix} \tag{6}$$

A lower triangular matrix has all elements above the main diagonal equal to zero, such as

$$\boldsymbol{L}_{n\times n} = \begin{bmatrix} l_{1,1} & 0 & \cdots & 0 \\ l_{2,1} & l_{2,2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ l_{n,1} & l_{n,2} & \cdots & l_{n,n} \end{bmatrix} \tag{7}$$

Transposing an upper triangular matrix produces a lower triangular matrix, and vice versa, as illustrated in Figure 9.

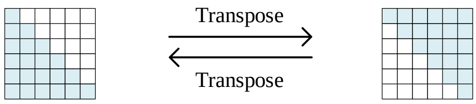


Figure 9. Transposing an upper triangular matrix produces a lower triangular matrix, and vice versa

Understanding these different matrix shapes and their properties is fundamental in linear algebra. They provide the foundation for more advanced operations, such as matrix decomposition, solving linear systems, and transforming datasets in machine learning.

3.4 Playing with Numbers: Basic Matrix Operations

3.4.1 Addition, Subtraction, and Scalar Multiplication

Matrix operations provide the foundation for many machine learning algorithms, so it is important to understand them intuitively. The simplest operation is **matrix addition**. If two matrices, $\boldsymbol{A}$ and $\boldsymbol{B}$, have the same shape, adding them means adding each pair of corresponding elements.

In other words, the element in the i-th row and j-th column of the resulting matrix is simply the sum of the elements in the same position in $\boldsymbol{A}$ and $\boldsymbol{B}$:

$$\boldsymbol{A}_{m\times p} + \boldsymbol{B}_{m\times p} = \begin{bmatrix} a_{1,1}+b_{1,1} & a_{1,2}+b_{1,2} & \cdots & a_{1,p}+b_{1,p} \\ a_{2,1}+b_{2,1} & a_{2,2}+b_{2,2} & \cdots & a_{2,p}+b_{2,p} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m,1}+b_{m,1} & a_{m,2}+b_{m,2} & \cdots & a_{m,p}+b_{m,p} \end{bmatrix}_{m\times p} \tag{8}$$

Similarly, **matrix subtraction** works element-wise: each element of $\boldsymbol{B}$ is subtracted from the corresponding element of $\boldsymbol{A}$:

$$\boldsymbol{A}_{m\times p} - \boldsymbol{B}_{m\times p} = \begin{bmatrix} a_{1,1}-b_{1,1} & a_{1,2}-b_{1,2} & \cdots & a_{1,p}-b_{1,p} \\ a_{2,1}-b_{2,1} & a_{2,2}-b_{2,2} & \cdots & a_{2,p}-b_{2,p} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m,1}-b_{m,1} & a_{m,2}-b_{m,2} & \cdots & a_{m,p}-b_{m,p} \end{bmatrix}_{m\times p} \tag{9}$$

3.4.2 Multiplying Matrices: Rules and Dimensions

Another fundamental operation is **scalar multiplication**, where a single number (a scalar) multiplies every element of a matrix.

If k is a scalar and $\boldsymbol{A}$ is a matrix, the product $k\boldsymbol{A}$ is a new matrix of the same shape as $\boldsymbol{A}$, with each element scaled by k:

$$k\boldsymbol{A} = \begin{bmatrix} k\cdot a_{1,1} & k\cdot a_{1,2} & \cdots & k\cdot a_{1,p} \\ k\cdot a_{2,1} & k\cdot a_{2,2} & \cdots & k\cdot a_{2,p} \\ \vdots & \vdots & \ddots & \vdots \\ k\cdot a_{m,1} & k\cdot a_{m,2} & \cdots & k\cdot a_{m,p} \end{bmatrix} \tag{10}$$

This operation can enlarge or shrink the values in the matrix depending on the value of k.

A special case occurs when $k = 0$, which produces the **zero matrix $\boldsymbol{O}$**, a matrix of the same size as $\boldsymbol{A}$ where every element is zero.

3.5 Matrix Multiplication: Intuitive Perspectives

3.5.1 Fundamentals of Matrix Multiplication

Matrix multiplication is a central operation in machine learning and linear algebra, but it is slightly more involved than addition or scalar multiplication. Suppose we have two matrices, $\boldsymbol{A}$ and $\boldsymbol{B}$.

Let $\boldsymbol{A}$ have shape $m \times p$, meaning it has m rows and p columns, and let $\boldsymbol{B}$ have shape $p \times n$, meaning it has p rows and n columns.

The product of these matrices, denoted

$$\boldsymbol{C} = \boldsymbol{AB} = \boldsymbol{A} @ \boldsymbol{B} \tag{11}$$

results in a new matrix $\boldsymbol{C}$ with shape $m \times n$. In this book, we often use the @ symbol to represent matrix multiplication in order to align with NumPy's syntax and conventions.

From a dimensional perspective, matrix multiplication is only defined when the number of columns in the first matrix $\boldsymbol{A}$ equals the number of rows in the second matrix $\boldsymbol{B}$.

If this condition is not met, the **multiplication cannot be performed**.

The resulting matrix $\boldsymbol{C}$ ($m \times n$) inherits the number of rows (m) from $\boldsymbol{A}$ and the number of columns (n) from $\boldsymbol{B}$, while the shared dimension p is "summed over" and disappears in the final product.

Each element of the product matrix $\boldsymbol{C}$ is computed as the dot product of a row from $\boldsymbol{A}$ and a column from $\boldsymbol{B}$ (see Figure 10) Specifically, $c_{i,j}$, the element in the i-th row and j-th column of $\boldsymbol{C}$ is calculated as

$$c_{i,j} = a_{i,1}b_{1,j} + a_{i,2}b_{2,j} + \cdots + a_{i,k}b_{k,j} + \cdots + a_{i,p}b_{p,j} = \sum_{k=1}^{p} a_{i,k}b_{k,j} \tag{12}$$

In simple terms, $c_{i,j}$ is obtained by multiplying each element of the i-th row of $\boldsymbol{A}$ with the corresponding element of the j-th column of $\boldsymbol{B}$ and then summing all these products. This operation can be interpreted as an **inner product** between the row vector from $\boldsymbol{A}$ and the column vector from $\boldsymbol{B}$.

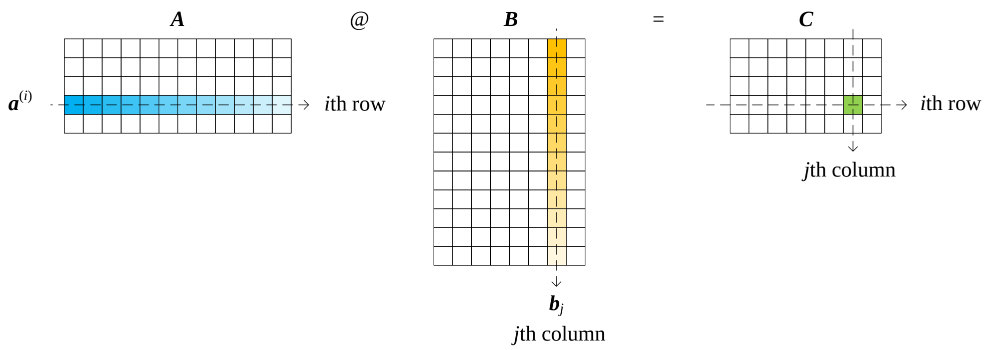


Figure 10. Illustration of how the element of $c_{i,j}$ in $\boldsymbol{C} = \boldsymbol{AB}$ is computed, showing the element-wise multiplication and summation of row and column vectors.

It is important to note that matrix multiplication is **generally not commutative**. This means that even if $\boldsymbol{A}$ @ $\boldsymbol{B}$ is defined, $\boldsymbol{B}$ @ $\boldsymbol{A}$ may not be defined, and even when both are defined, their results are usually different. Understanding this property is crucial for correctly implementing algorithms in machine learning, where the order of matrix operations directly affects outcomes.

3.5.2 Linear Transformations

The core purpose of matrix multiplication is to perform **linear transformations**.

But what exactly is a linear transformation?

Consider the matrix equation $\boldsymbol{Ax} = \boldsymbol{y}$, where $\boldsymbol{A}$ is a 2×2 matrix and $\boldsymbol{x}$ is a 2×1 vector. As shown in Figure 11, $\boldsymbol{x}$ can be interpreted geometrically as a directed line segment in the plane whose tail (starting point) is at the origin and whose head (endpoint) is located at the coordinates given by $\boldsymbol{x}$. In other words, the vector points from (0, 0) to (x_1, x_2).

The result $\boldsymbol{y}$ is also a 2×1 (column) vector on a 2D plane.

In this case, multiplying $\boldsymbol{x}$ by $\boldsymbol{A}$ can be interpreted as applying a transformation to the two-dimensional plane. Depending on the matrix, this transformation might scale, rotate, reflect, shear, project, or combine several of these effects in sequence.

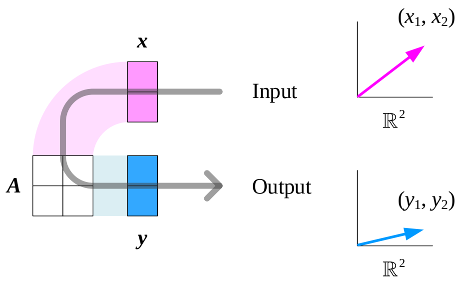


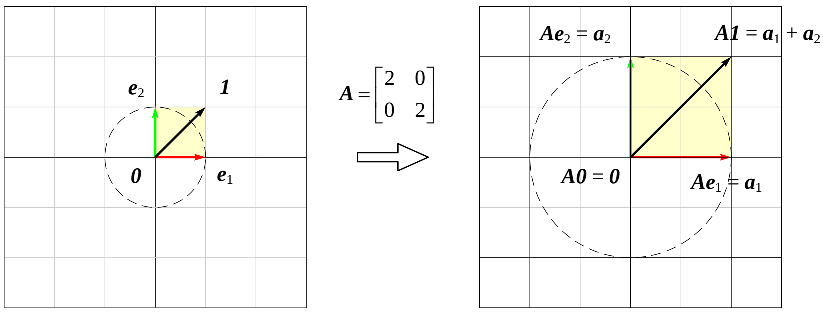
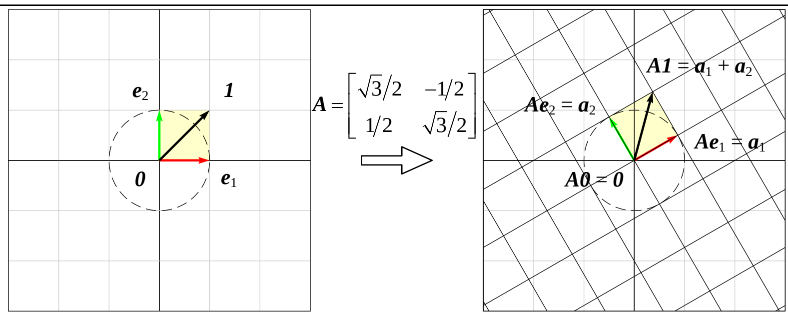
Figure 11. Visualization of a 2 × 2 matrix $\boldsymbol{A}$ acting on the plane

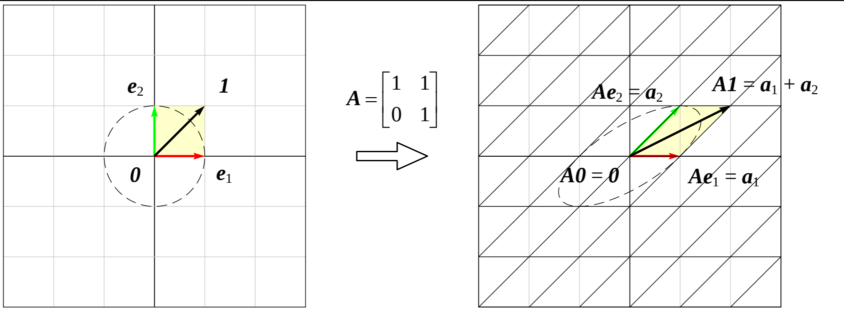
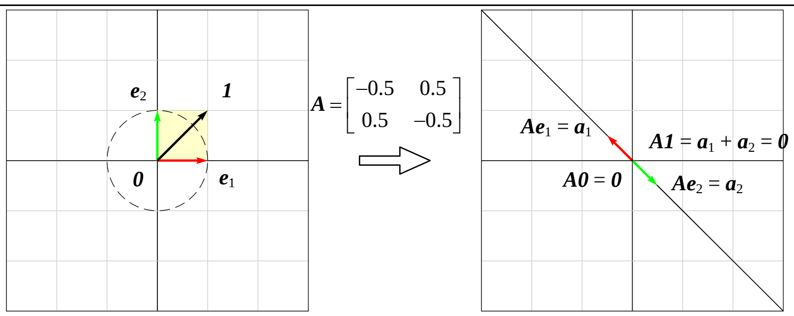
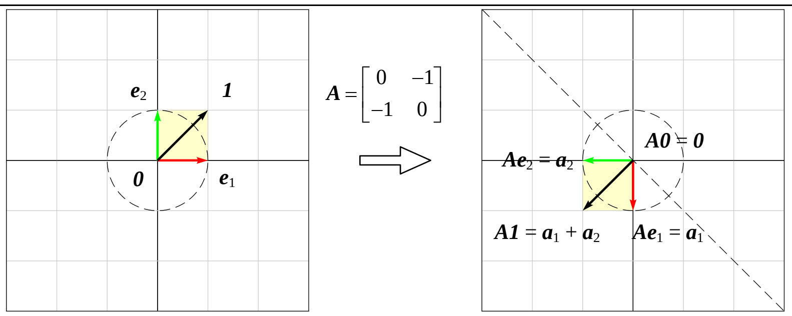
A **linear transformation** is a transformation that preserves both vector addition and scalar multiplication. Geometrically, this means it can stretch, rotate, shear, or flip the space, but it does not bend, curve, or distort it. An important property of all linear transformations is that the **origin remains fixed**, meaning the point (0, 0) always maps to itself.

Table 1 illustrates several simple geometric transformations in the plane, showing how a 2 × 2 matrix $\boldsymbol{A}$ acts on the standard basis vectors $\boldsymbol{e}_1$ and $\boldsymbol{e}_2$. Each transformation corresponds to a familiar geometric effect, such as scaling, rotation, horizontal shear, orthogonal projection, or reflection.

These examples provide an intuitive way to understand how matrices "move" vectors in space, forming the foundation for more advanced concepts in machine learning, computer graphics, and linear algebra.

Table 1: Common simple geometric transformations in the plane.

<table>
<tr><th>Transformation</th><th>Matrix</th><th>Visualization</th></tr>
<tr><td>Scaling</td><td>$\begin{bmatrix} s_1 & 0 \\ 0 & s_2 \end{bmatrix}$</td><td>
</td></tr>
<tr><td>Rotation</td><td>$\begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$
Rotate the object counterclockwise by an angle θ around the origin.</td><td>
</td></tr>
</table>

Horizontal Shear	$\begin{bmatrix} 1 & k \\ 0 & 1 \end{bmatrix}$	
Orthogonal Projection	$\boldsymbol{I} - \frac{\boldsymbol{n} @ \boldsymbol{n}^{\mathrm{T}}}{\boldsymbol{n}^{\mathrm{T}} @ \boldsymbol{n}}$ ***n*** is the normal vector of the line, onto which a vector is orthogonally projected.	
Reflection	$\boldsymbol{I} - 2 \cdot \frac{\boldsymbol{n} @ \boldsymbol{n}^{\mathrm{T}}}{\boldsymbol{n}^{\mathrm{T}} @ \boldsymbol{n}}$ ***n*** is the normal vector of the line across which a vector is reflected.	

3.5.3 Matrix Multiplication: The Inner Product Perspective

Matrix multiplication can be understood from several viewpoints, and one of the most intuitive is the inner product perspective.

As shown in Figure 12, let matrix ***A*** have shape $m \times p$ and matrix ***B*** have shape $p \times n$. When we multiply ***A*** and ***B*** to form ***C*** = ***AB***, it helps to visualize their alignment. Place matrix ***A*** on the left side of the resulting matrix ***C***, aligning their rows, and place matrix ***B*** above ***C***, aligning their columns. This visual arrangement makes the shape rule of matrix multiplication immediately clear.

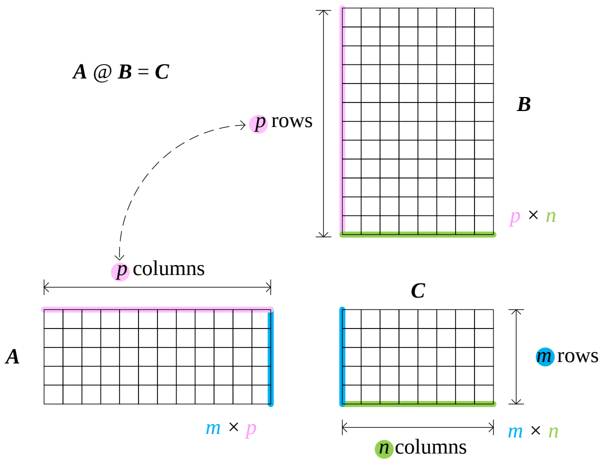


Figure 12. Aligning $\boldsymbol{A}$ on the left (row-aligned) and $\boldsymbol{B}$ on top (column-aligned) of $\boldsymbol{C} = \boldsymbol{AB}$, showing how the dimensions of the result are determined.

The rows of $\boldsymbol{A}$ determine the number of rows in $\boldsymbol{C}$ (that is, m), while the columns of $\boldsymbol{B}$ determine the number of columns in $\boldsymbol{C}$ (that is, n). Therefore, the product $\boldsymbol{C} = \boldsymbol{A}$ @ $\boldsymbol{B}$ always has shape $m \times n$.

This alignment also reveals the fundamental rule of matrix multiplication: the number of columns in $\boldsymbol{A}$ must equal the number of rows in $\boldsymbol{B}$. Without this match, the multiplication cannot be performed because the dimensions fail to align.

As illustrated in Figure 13, each element $c_{i,j}$ in the resulting matrix $\boldsymbol{C}$ is computed by taking the **dot product** (inner product) between the i-th row of $\boldsymbol{A}$ and the j-th column of $\boldsymbol{B}$. In other words, $c_{i,j}$ is obtained by multiplying corresponding elements from that row and column and summing the results.

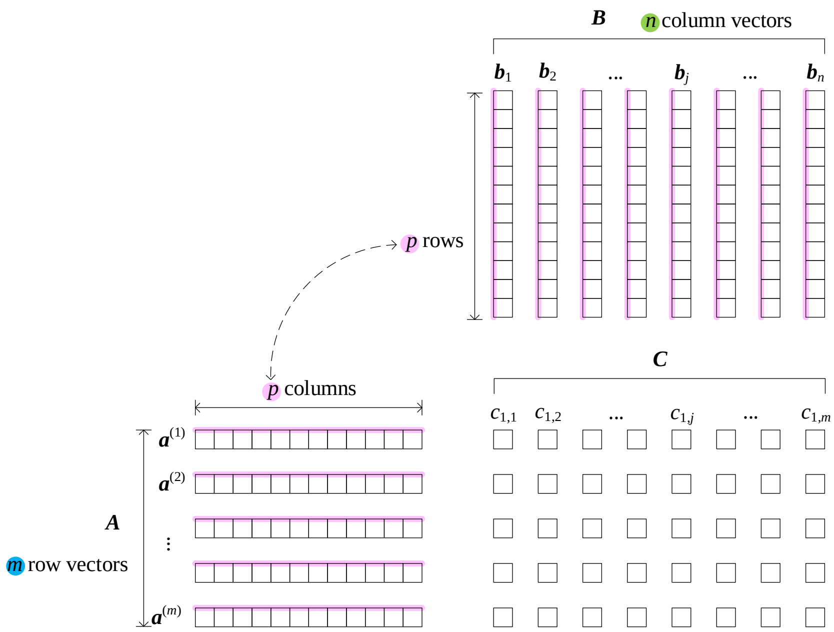


Figure 13. Computing each element $c_{i,j}$ of $\boldsymbol{C} = \boldsymbol{AB}$ as the inner product of the i-th row of $\boldsymbol{A}$and the j-th column of $\boldsymbol{B}$, illustrated using the row-and-column alignment view.

3.5.4 Matrix Multiplication: The Outer Product Perspective

Matrix multiplication can also be understood from another powerful viewpoint — the **outer product perspective**. Consider two matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ such that $\boldsymbol{A}$ has shape $m \times p$ and $\boldsymbol{B}$ has shape $p \times n$.

We can express matrix $\boldsymbol{A}$ as a collection of its **column vectors**:

$$\boldsymbol{A}_{m\times p} = \begin{bmatrix} \boldsymbol{a}_1 & \boldsymbol{a}_2 & \cdots & \boldsymbol{a}_p \end{bmatrix} \tag{13}$$

where each column vector $\boldsymbol{a}_k$ has shape $m \times 1$.

On contrary, we can express matrix $\boldsymbol{B}$ as a collection of its **row vectors**:

$$\boldsymbol{B}_{p\times n} = \begin{bmatrix} \boldsymbol{b}^{(1)} \\ \boldsymbol{b}^{(2)} \\ \vdots \\ \boldsymbol{b}^{(p)} \end{bmatrix} \tag{14}$$

where each row vector $\boldsymbol{b}^{(k)}$ has shape $1 \times n$.

With these representations, the matrix product $\boldsymbol{C} = \boldsymbol{A}$ @ $\boldsymbol{B}$ can be written as the sum of p outer products:

$$\boldsymbol{A}_{m\times p} @ \boldsymbol{B}_{p\times n} = \begin{bmatrix} \boldsymbol{a}_1 & \boldsymbol{a}_2 & \cdots & \boldsymbol{a}_p \end{bmatrix} @ \begin{bmatrix} \boldsymbol{b}^{(1)} \\ \boldsymbol{b}^{(2)} \\ \vdots \\ \boldsymbol{b}^{(p)} \end{bmatrix} = \boldsymbol{a}_1 @ \boldsymbol{b}^{(1)} + \boldsymbol{a}_2 @ \boldsymbol{b}^{(2)} + \cdots + \boldsymbol{a}_p @ \boldsymbol{b}^{(p)} = \sum_{k=1}^{p} \boldsymbol{a}_k @ \boldsymbol{b}^{(k)} \tag{15}$$

Surprisingly, this shows that matrix multiplication can be viewed as the sum of p matrices of the same shape, as illustrated in Figure 14.

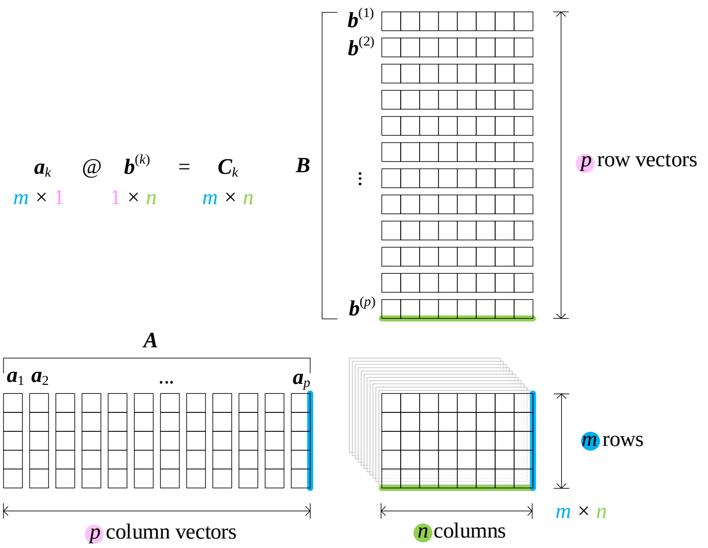


Figure 14. Visualizing matrix multiplication $\boldsymbol{A}$ @ $\boldsymbol{B}$ as the sum of several matrices of identical shape, each formed by an outer product between a column of $\boldsymbol{A}$ and a row of $\boldsymbol{B}$.

Each term $\boldsymbol{a}_k @ \boldsymbol{b}^{(k)}$ represents an **outer product**, and the index k corresponds to the dimension that "disappears" (the shared dimension between $\boldsymbol{A}$ and $\boldsymbol{B}$).

If we define

$$\boldsymbol{C}_k = \boldsymbol{a}_k @ \boldsymbol{b}^{(k)} \tag{16}$$

then $\boldsymbol{C}$ is simply the sum of all $\boldsymbol{C}_k$

$$\boldsymbol{C} = \boldsymbol{C}_1 + \boldsymbol{C}_1 + \cdots + \boldsymbol{C}_p \tag{17}$$

Each $\boldsymbol{C}_k$ and $\boldsymbol{C}$ share the same shape $m \times n$. From a dimensional point of view, the intermediate dimension 1 in $(m \times 1) @ (1 \times n)$ collapses, resulting in an $m \times n$ matrix.

This way of expressing matrix multiplication is known as the **outer product expansion**.

To understand the outer product more concretely, let's focus on two column vectors $\boldsymbol{a}$ and $\boldsymbol{b}$. As shown in Figure 15, the product $\boldsymbol{a} @ \boldsymbol{b}^{\mathrm{T}}$ produces a matrix in which the **rows and columns are scalar multiples** of one another.

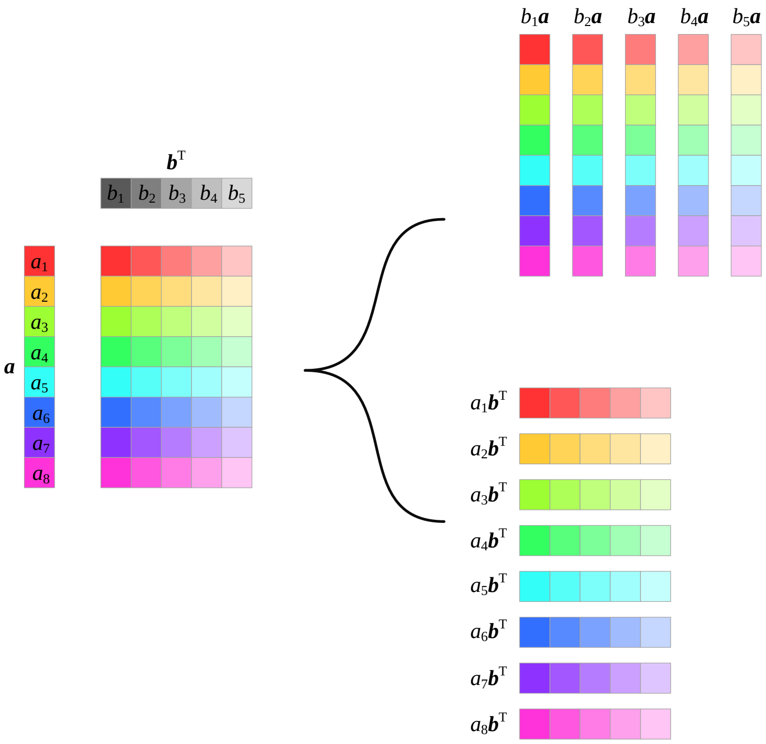


Figure 15. The matrix $\boldsymbol{a} @ \boldsymbol{b}^{\mathrm{T}}$ has rows and columns that are scalar multiples of a single vector, forming a rank-one structure.

Each **column** of $\boldsymbol{a} @ \boldsymbol{b}^{\mathrm{T}}$ can be written as a scalar multiple of $\boldsymbol{a}$:

$$\boldsymbol{a} @ \boldsymbol{b}^{\mathrm{T}} = \boldsymbol{a} @ \begin{bmatrix} b_1 & b_2 & \cdots & b_n \end{bmatrix} = \begin{bmatrix} b_1\boldsymbol{a} & b_2\boldsymbol{a} & \cdots & b_n\boldsymbol{a} \end{bmatrix} \tag{18}$$

and each **row** can be written as a scalar multiple of $\boldsymbol{b}^{\mathrm{T}}$:

$$\boldsymbol{a} @ \boldsymbol{b}^{\mathrm{T}} = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_m \end{bmatrix} \boldsymbol{b}^{\mathrm{T}} = \begin{bmatrix} a_1\boldsymbol{b}^{\mathrm{T}} \\ a_2\boldsymbol{b}^{\mathrm{T}} \\ \vdots \\ a_m\boldsymbol{b}^{\mathrm{T}} \end{bmatrix} \tag{19}$$

Because all rows are multiples of one row vector, and all columns are multiples of one column vector, the resulting matrix has **rank one**. In general, the **rank** of a matrix represents the maximum number of linearly independent rows or columns. For a rank-one matrix, every row and column is linearly dependent on the others.

If either aaa or bbb is the zero vector, then $\boldsymbol{a}$ @ $\boldsymbol{b}^{\mathrm{T}}$ becomes the **zero matrix** $\boldsymbol{O}$, whose rank is 0.

This outer product viewpoint is deeply important in machine learning, especially in **Principal Component Analysis (PCA)**, where data matrices are decomposed into sums of rank-one matrices that capture the main directions of variation.

3.5.5 Matrix Multiplication from a Third Perspective: Linear Combination of Column Vectors

Let us begin with a simple case of matrix–vector multiplication,

$$\boldsymbol{A}_{m\times p}\boldsymbol{b}_{p\times 1} = \boldsymbol{c} \tag{20}$$

where $\boldsymbol{A}$ has shape $m \times p$, $\boldsymbol{b}$ has shape $p \times 1$, and their product $\boldsymbol{c}$ is an $m \times 1$ column vector. Figure 16 illustrates this operation.

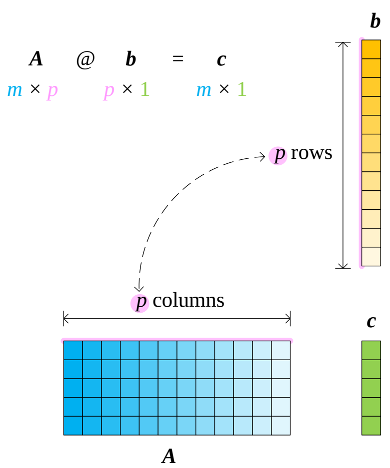


Figure 16. Matrix–vector multiplication $\boldsymbol{Ab} = \boldsymbol{c}$

Now, let's look more closely at what this multiplication really means.

We can express matrix $\boldsymbol{A}$ as a collection of its column vectors:

$$\boldsymbol{A}_{m\times p} = \begin{bmatrix} \boldsymbol{a}_1 & \boldsymbol{a}_2 & \cdots & \boldsymbol{a}_p \end{bmatrix} \tag{21}$$

where each column vector $\boldsymbol{a}_k$ has shape $m \times 1$.

Using this representation, we can rewrite the multiplication $\boldsymbol{A}$ @ $\boldsymbol{b}$:

$$\boldsymbol{A}_{m\times p}\boldsymbol{b}_{p\times 1} = \begin{bmatrix} \boldsymbol{a}_1 & \boldsymbol{a}_2 & \cdots & \boldsymbol{a}_p \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_p \end{bmatrix} = b_1\boldsymbol{a}_1 + b_2\boldsymbol{a}_2 + \cdots + b_p\boldsymbol{a}_p = \boldsymbol{c} \tag{22}$$

In other words, the result ccc is a **linear combination of the column vectors** of $\boldsymbol{A}$, where the coefficients are given by the components of $\boldsymbol{b}$. This is the **third viewpoint of matrix multiplication**—the *column vector linear combination perspective*.

Now consider the more general case of multiplying two matrices $\boldsymbol{C} = \boldsymbol{A} @ \boldsymbol{B}$.

We can again use the same column combination perspective. Let both $\boldsymbol{B}$ and $\boldsymbol{C}$ be written in terms of their column vectors:

$$\boldsymbol{C} = \begin{bmatrix} \boldsymbol{c}_1 & \boldsymbol{c}_2 & \cdots & \boldsymbol{c}_n \end{bmatrix} = \boldsymbol{A} @ \boldsymbol{B} = \boldsymbol{A} @ \begin{bmatrix} \boldsymbol{b}_1 & \boldsymbol{b}_2 & \cdots & \boldsymbol{b}_n \end{bmatrix} = \begin{bmatrix} \boldsymbol{A} @ \boldsymbol{b}_1 & \boldsymbol{A} @ \boldsymbol{b}_2 & \cdots & \boldsymbol{A} @ \boldsymbol{b}_n \end{bmatrix} \tag{23}$$

Then, as shown in Figure 17, the multiplication $\boldsymbol{C} = \boldsymbol{A} @ \boldsymbol{B}$ can be viewed column by column:

$$\boldsymbol{c}_j = \boldsymbol{A} @ \boldsymbol{b}_j \tag{24}$$

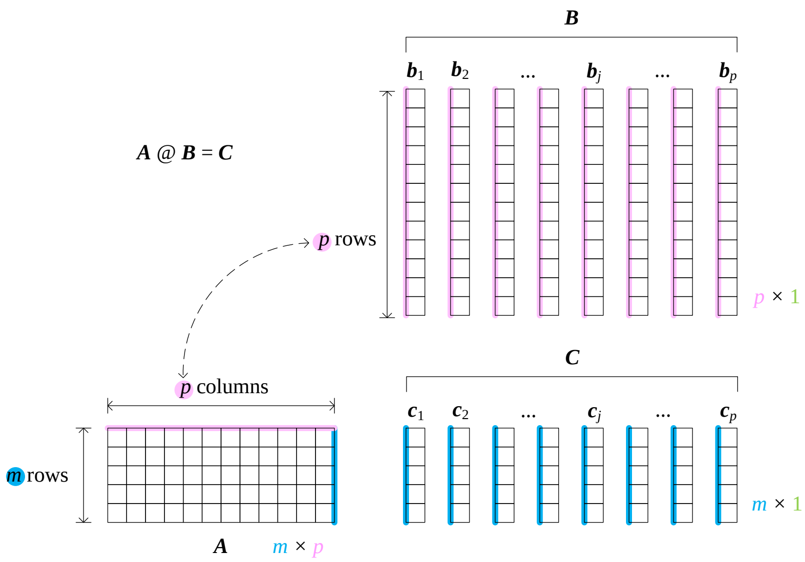


Figure 17. Both $\boldsymbol{B}$ and $\boldsymbol{C}$ can be viewed as collections of column vectors

Finally, after computing all the individual products $\boldsymbol{Ab}_1$, $\boldsymbol{Ab}_2$, $\boldsymbol{Ab}_3$, …, $\boldsymbol{Ab}_n$, we can place them side by side to reconstruct the full product matrix:

$$\begin{bmatrix} \boldsymbol{A} @ \boldsymbol{b}_1 & \boldsymbol{A} @ \boldsymbol{b}_2 & \cdots & \boldsymbol{A} @ \boldsymbol{b}_n \end{bmatrix} = \begin{bmatrix} \boldsymbol{c}_1 & \boldsymbol{c}_2 & \cdots & \boldsymbol{c}_n \end{bmatrix} \tag{25}$$

This perspective helps us see that **matrix multiplication is nothing more than performing multiple vector combinations in parallel**, one for each column of the result matrix.

The three viewpoints we have explored—inner product, outer product expansion, and column combination interpretation—are not separate rules, but rather complementary ways of understanding the same operation. Each highlights a different structural aspect of matrix multiplication and deepens our intuition for later topics such as linear transformations, subspaces, and principal component analysis (PCA).

3.6 Determinants: Measuring Area, Volume, and Orientation

3.6.1 The 2×2 Case: Parallelograms and Area

The **determinant** of a square matrix is a scalar that captures how a linear transformation changes space. Geometrically, it tells us how the transformation defined by a matrix stretches or shrinks areas (in 2D), volumes (in 3D), or higher-dimensional hypervolumes. In essence, the determinant maps a matrix to a single real number that summarizes how that matrix scales and orients space.

The 2 × 2 Case: From Parallelogram to Area

Consider a 2 × 2 matrix

$$\boldsymbol{A} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \tag{26}$$

The determinant of $\boldsymbol{A}$ is given by

$$\det(\boldsymbol{A}) = \begin{vmatrix} a & b \\ c & d \end{vmatrix} = ad - bc \tag{27}$$

We can think of $\boldsymbol{A}$ as composed of two column vectors:

$$\boldsymbol{A} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \quad \Rightarrow \quad \boldsymbol{a}_1 = \begin{bmatrix} a \\ c \end{bmatrix}, \quad \boldsymbol{a}_2 = \begin{bmatrix} b \\ d \end{bmatrix} \tag{28}$$

When we place both column vectors ($\boldsymbol{a}_1$ and $\boldsymbol{a}_2$) at the origin and draw them in the plane, they form a **parallelogram** (Figure 18). The area of this parallelogram, including its sign, is exactly the determinant of $\boldsymbol{A}$.

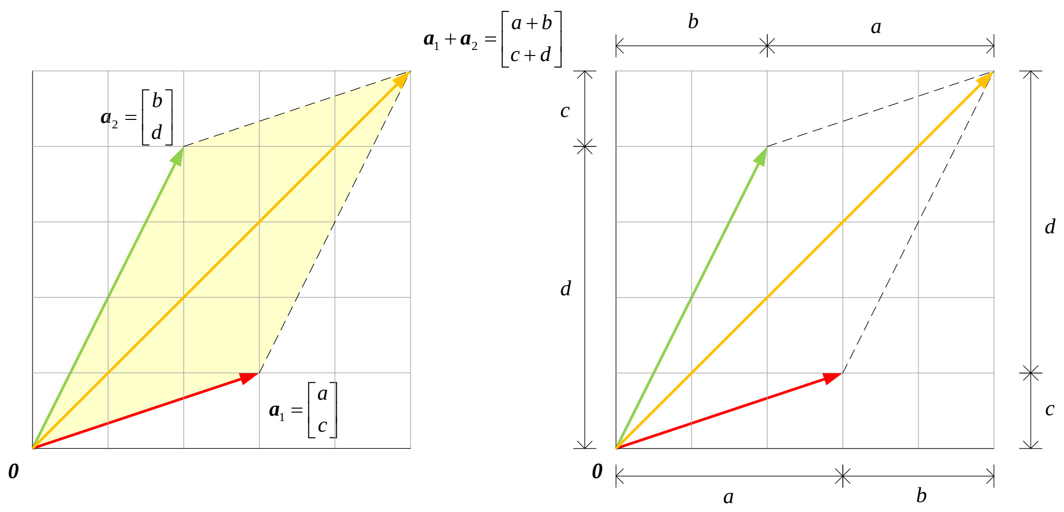


Figure 18. The parallelogram formed by the two column vectors of a 2 × 2 matrix

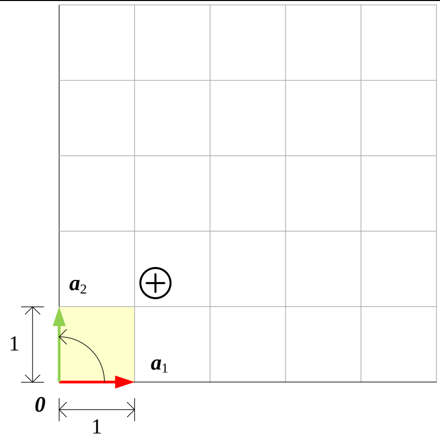
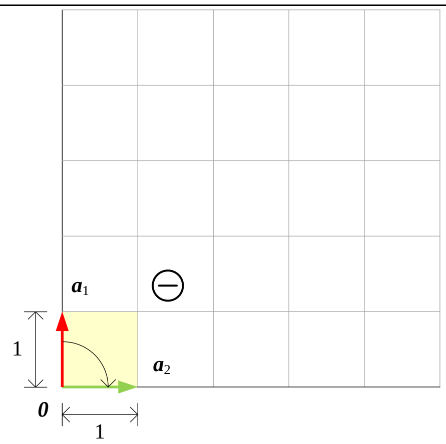
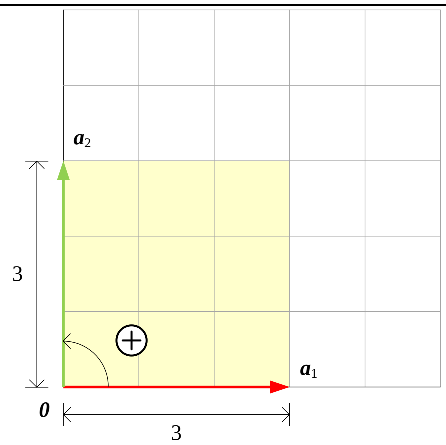
If $|\det(\boldsymbol{A})| > 1$, the linear transformation represented by $\boldsymbol{A}$ expands the area.

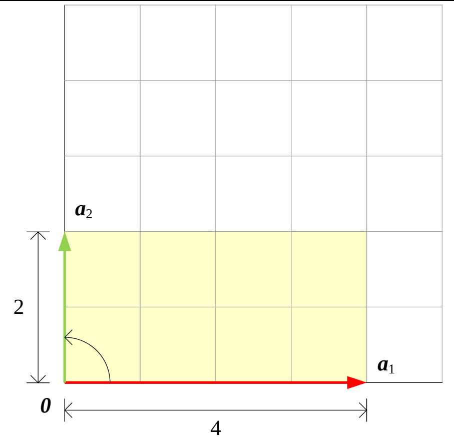
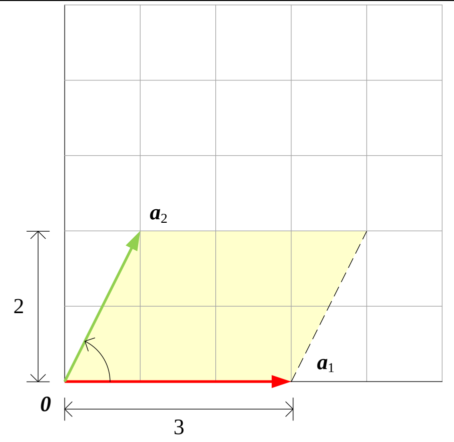
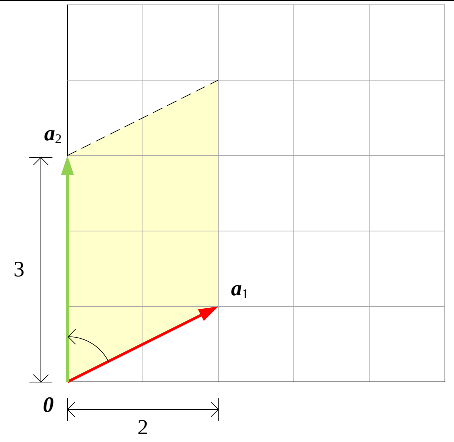
If $0 < |\det(\boldsymbol{A})| < 1$, it compresses the area.

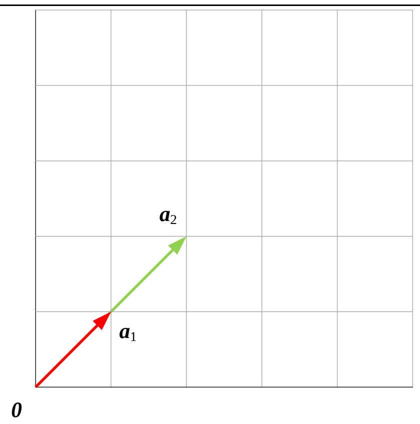
If $|\det(\boldsymbol{A})| = 0$, the transformation collapses the plane into a line — a sign of **dimensional reduction**.

If $\det(\boldsymbol{A})| < 0$, the transformation includes a **reflection** (it flips the orientation).

Table 2. Determinants and geometric interpretations for special 2×2 matrices

Square matrix	Determinant	Geometric effect
$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ Identity matrix	1 Preserves area and orientation	
$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$ Permutation matrix	−1 Same area, but reflected	
$\begin{bmatrix} 3 & 0 \\ 0 & 3 \end{bmatrix}$ Diagonal matrix (same diagonal entries)	9 Uniform scaling	

$\begin{bmatrix} 4 & 0 \\ 0 & 2 \end{bmatrix}$ Diagonal matrix (different entries)	8 Non-uniform scaling	
$\begin{bmatrix} 3 & 1 \\ 0 & 2 \end{bmatrix}$ Upper triangular matrix	6 Scaling + horizontal shear	
$\begin{bmatrix} 2 & 0 \\ 1 & 3 \end{bmatrix}$ Lower triangular matrix	6 Scaling + vertical shear	

$\begin{bmatrix} 1 & 2 \\ 1 & 2 \end{bmatrix}$ Linearly dependent	0 Collapsed area (degenerate)	

3.6.2 The 3×3 Case: Parallelepipeds and Volume

Now consider a 3 × 3 matrix

$$A = \begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,1} & a_{2,2} & a_{2,3} \\ a_{3,1} & a_{3,2} & a_{3,3} \end{bmatrix} \tag{29}$$

If we decompose A into three column vectors,

$$A = \begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,1} & a_{2,2} & a_{2,3} \\ a_{3,1} & a_{3,2} & a_{3,3} \end{bmatrix} \Rightarrow \quad a_1 = \begin{bmatrix} a_{1,1} \\ a_{2,1} \\ a_{3,1} \end{bmatrix}, \quad a_2 = \begin{bmatrix} a_{1,2} \\ a_{2,2} \\ a_{3,2} \end{bmatrix}, \quad a_3 = \begin{bmatrix} a_{1,3} \\ a_{2,3} \\ a_{3,3} \end{bmatrix} \tag{30}$$

these vectors span a **parallelepiped** in 3D space (Figure 19). The absolute value of the determinant gives the **volume** of that parallelepiped, and the sign indicates its **orientation**.

To determine the sign, we can use the **right-hand rule**:

- Curl the fingers of your right hand from the direction of a_1 to a_2.
- If your thumb points in the same direction as a_3, then det(A) > 0 (a right-handed system).
- If your thumb points in the opposite direction, then det(A) < 0.

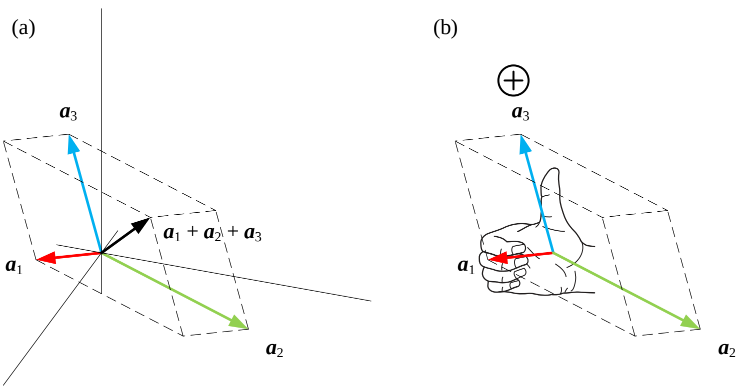
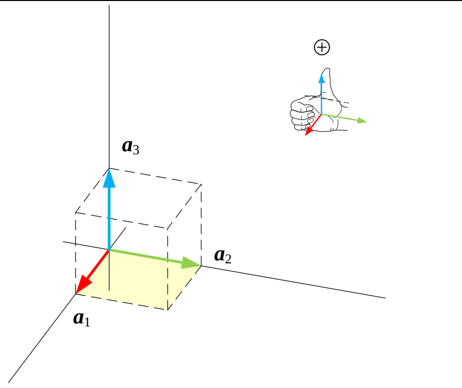
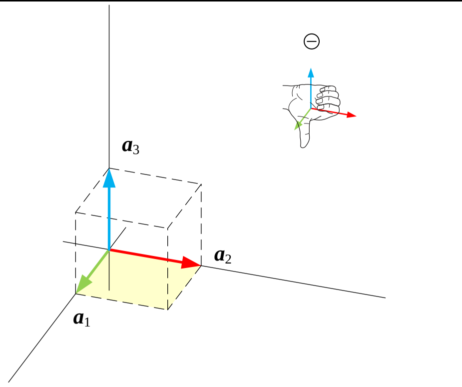
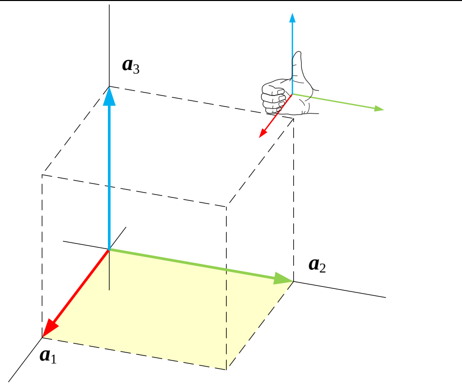
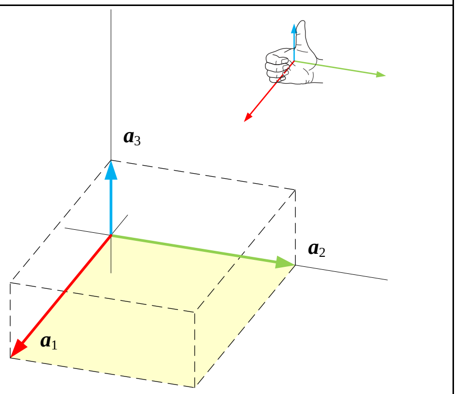
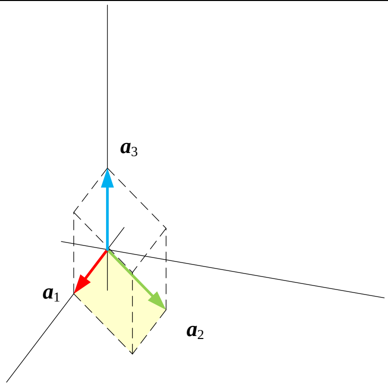
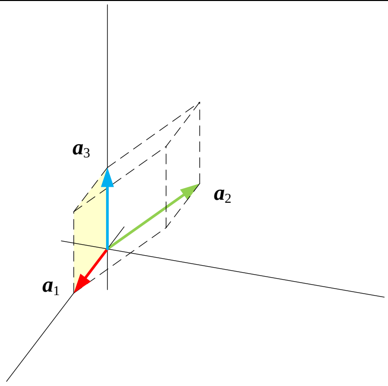
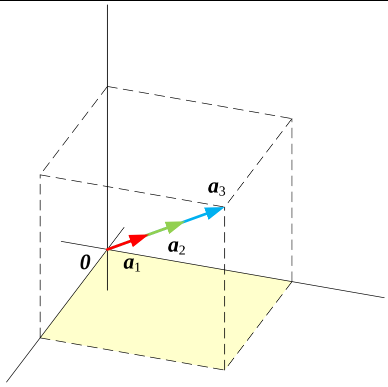
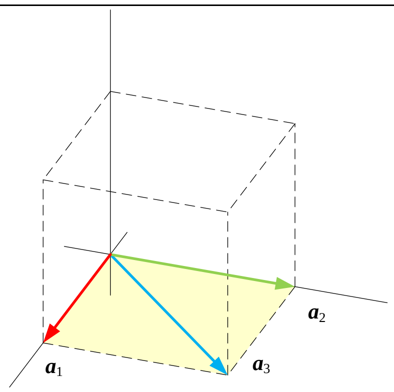


Figure 19. The parallelepiped formed by the three column vectors of a 3×3 matrix

Table 3. Determinants and geometric interpretations for special 3×3 matrices

Square matrix	Determinant	Geometric effect
$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ Identity matrix	1 Preserves volume and orientation	
$\begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ Permutation matrix	−1 Same volume, but reflected	
$\begin{bmatrix} 3 & 0 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & 3 \end{bmatrix}$ Diagonal matrix (same diagonal entries)	27 Uniform 3D scaling	
$\begin{bmatrix} 3 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ Diagonal matrix (different entries)	6 Non-uniform 3D scaling	

$\begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ Upper triangular matrix	1 Shear	
$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix}$ Lower triangular matrix	1 Shear	
$\begin{bmatrix} 1 & 2 & 3 \\ 1 & 2 & 3 \\ 1 & 2 & 3 \end{bmatrix}$ Linearly dependent	0	
$\begin{bmatrix} 3 & 0 & 3 \\ 0 & 3 & 3 \\ 0 & 0 & 0 \end{bmatrix}$ Linearly dependent	0	

3.7 Inverses of Matrices and Undoing Transformations

3.7.1 When a Matrix Can Be Reversed

For an $n \times n$ square matrix $\boldsymbol{A}$, if there exists another matrix $\boldsymbol{A}^{-1}$ such that

$$\boldsymbol{A} @ \boldsymbol{A}^{-1} = \boldsymbol{A}^{-1} @ \boldsymbol{A} = \boldsymbol{I} \tag{31}$$

where $\boldsymbol{I}$ is the $n \times n$ identity matrix, then $\boldsymbol{A}$ is said to be **invertible** or **non-singular**. The matrix $\boldsymbol{A}^{-1}$ is called the **inverse** of $\boldsymbol{A}$.

A square matrix $\boldsymbol{A}$ is invertible **if and only if** its determinant is nonzero: $\det(\boldsymbol{A}) \neq 0$.

Intuitively, this means the linear transformation defined by $\boldsymbol{A}$ does not collapse space—it does not flatten areas, volumes, or higher-dimensional objects. When no information is lost under the transformation, the process can be reversed, and thus the matrix is invertible.

Consider a 2×2 matrix

$$\boldsymbol{A} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \tag{32}$$

Its inverse exists when $ad - bc \neq 0$, and is given by

$$\boldsymbol{A}^{-1} = \frac{1}{ad - bc}\begin{bmatrix} d & -b \\ -c & a \end{bmatrix} \tag{33}$$

If the determinant equals zero, the transformation compresses the plane into a line or a point, making reversal impossible.

From a geometric perspective, multiplying a vector $\boldsymbol{x}$ by $\boldsymbol{A}$ produces a new vector $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{y}$, which represents a linear transformation such as scaling, rotation, shear, or reflection (Figure 11).

Multiplying $\boldsymbol{y}$ by $\boldsymbol{A}^{-1}$ undoes that transformation, mapping $\boldsymbol{y}$ back to the original vector $\boldsymbol{x}$ (Figure 20). The inverse matrix is therefore the mathematical operator that **reverses** the effects of $\boldsymbol{A}$.

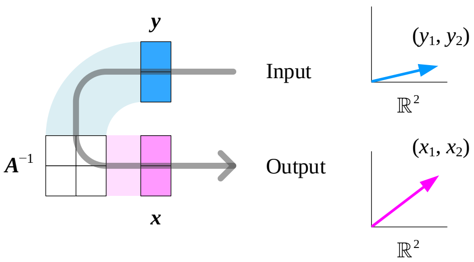


Figure 20. The reverse mapping $\boldsymbol{x} = \boldsymbol{A}^{-1}\boldsymbol{y}$ undoing the transformation

3.7.2 Scaling and Inverse Scaling

Consider a scaling matrix

$$\boldsymbol{S} = \begin{bmatrix} 3/2 & 0 \\ 0 & 2/3 \end{bmatrix} \tag{34}$$

Matrix $\boldsymbol{S}$ scales all points in the plane by a factor of 3/2 along the x_1-axis and 2/3 along the x_2-axis. As a result, shapes are stretched horizontally and compressed vertically.

Interestingly, the determinant of $\boldsymbol{S}$ is det($\boldsymbol{S}$) = 1, meaning the transformation preserves **area**, even though it distorts shape.

The inverse of $\boldsymbol{S}$ reverses this effect:

$$\boldsymbol{S}^{-1} = \begin{bmatrix} 2/3 & 0 \\ 0 & 3/2 \end{bmatrix} \tag{35}$$

This inverse scales the x_1-axis by 2/3 and the x_2-axis by 3/2, perfectly undoing the deformation (Figure 21).

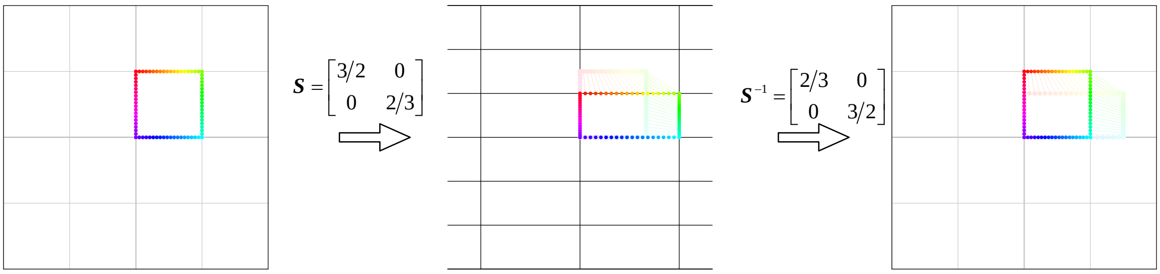


Figure 21. Scaling and its inverse scaling

3.7.3 Rotation and Inverse Rotation

A rotation matrix in two dimensions is defined as

$$\boldsymbol{R} = \begin{bmatrix} -1/2 & -\sqrt{3}/2 \\ \sqrt{3}/2 & -1/2 \end{bmatrix} \tag{36}$$

The matrix $\boldsymbol{R}$ rotates all vectors counterclockwise by 120 degrees around the origin. The determinant of any rotation matrix is 1, indicating that rotation preserves area and orientation.

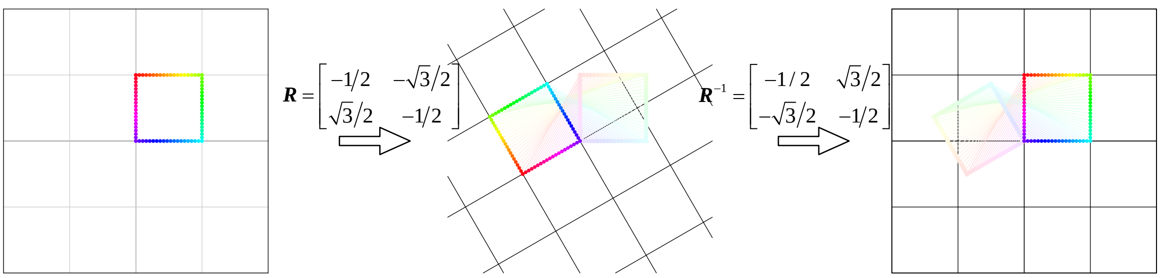


Figure 22. Rotation and its inverse rotation in the plane

The inverse rotation matrix is simply the rotation in the **opposite** direction:

$$\boldsymbol{R}^{-1} = \begin{bmatrix} -1/2 & \sqrt{3}/2 \\ -\sqrt{3}/2 & -1/2 \end{bmatrix} \tag{37}$$

Thus, if $\boldsymbol{R}$ rotates vectors counterclockwise by 120 degrees, $\boldsymbol{R}^{-1}$ rotates them clockwise by 120 degrees, restoring their original positions (Figure 22).

You might have noticed for the rotation matrix $\boldsymbol{R}$, its inverse is equal to it transpose

$$\boldsymbol{R}^{-1} = \boldsymbol{R}^{\mathrm{T}} \tag{38}$$

Thus

$$\boldsymbol{R}^{\mathrm{T}}\boldsymbol{R} = \boldsymbol{R}\boldsymbol{R}^{\mathrm{T}} = \boldsymbol{I} \tag{39}$$

We call $\boldsymbol{R}$ an orthogonal matrix.

Geometrically, an orthogonal matrix represents a linear transformation that **preserves both lengths and angles**. This means that when you multiply any vector $\boldsymbol{x}$ by an orthogonal matrix $\boldsymbol{R}$, the result $\boldsymbol{R}\boldsymbol{x}$ has exactly the same length as $\boldsymbol{x}$. The transformation may rotate or reflect the vector, but it never stretches, compresses, or distorts it. This property makes orthogonal matrices fundamental in many areas of mathematics and machine learning.

3.7.4 Shear and Inverse Shear

A shear transformation skews the coordinate system without changing the overall area. Consider

$$\boldsymbol{K} = \begin{bmatrix} 1 & 0.5 \\ 0 & 1 \end{bmatrix} \tag{40}$$

Matrix $\boldsymbol{K}$ shears points horizontally: each vector's x_1-component increases by half of its x_2-coordinate. For example, (1, 1) becomes (1.5, 1), and (2, 3) becomes (3.5, 3).

The determinant of $\boldsymbol{K}$ equals 1, confirming that shear does not change the area.

The inverse shear removes this distortion:

$$\boldsymbol{K}^{-1} = \begin{bmatrix} 1 & -0.5 \\ 0 & 1 \end{bmatrix} \tag{41}$$

Applying $\boldsymbol{K}^{-1}$, exactly reverses the shear (Figure 23).

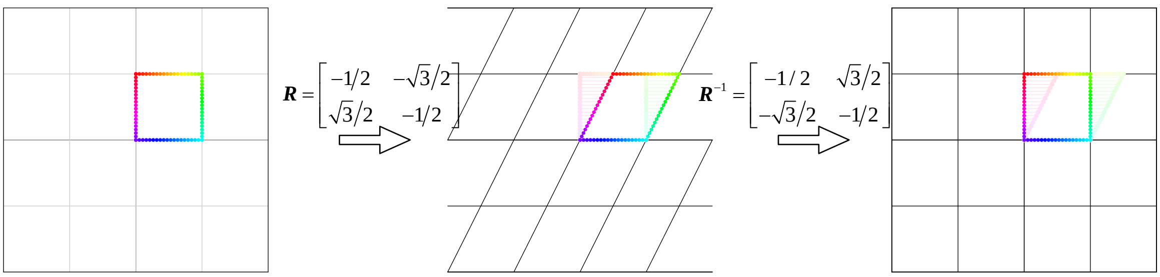


Figure 23. Shear and inverse shear:

3.7.5 Reflection and Inverse Reflection

A reflection (or mirror) matrix flips vectors across an axis. For example,

$$M = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \tag{42}$$

reflects points across the vertical axis (x_2-axis). This means the x_1-coordinate changes sign, while the x_2-coordinate remains the same. The determinant is −1, whose absolute value equals one, meaning the area is preserved—but the negative sign shows that orientation is **reversed**.

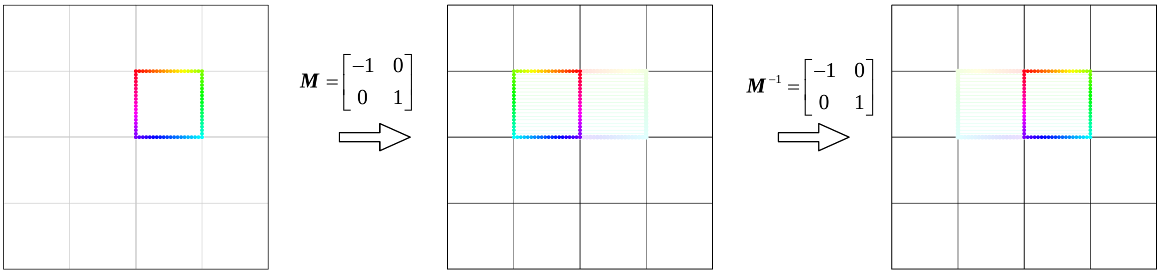


Figure 24. Reflection and its inverse

The inverse of a reflection is simply the same reflection again, since applying it twice restores the original configuration:

$$M^{-1} = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \tag{43}$$

Thus, reflections are self-inverse (Figure 24).

3.8 Conclusion

This chapter introduces the essential concepts of matrices for machine learning, focusing on their structure, types, and operations. A matrix is an ordered array of numbers organized in rows and columns, which can be viewed as collections of row or column vectors.

Key special types include square matrices, diagonal matrices, identity matrices, triangular matrices, and symmetric matrices, each with unique properties that simplify computations and support linear transformations.

Fundamental operations such as addition, scalar multiplication, transposition, and especially matrix multiplication are explained both algebraically and geometrically, highlighting perspectives like inner product, outer product, and linear combination of columns. The determinant is introduced as a measure of scaling and orientation, linking algebraic properties to geometric effects.

Finally, the chapter covers inverses and linear transformations, showing how scaling, rotation, shear, and reflection can be reversed, and emphasizing orthogonal matrices, which preserve lengths and angles, a concept central to many machine learning applications.