05. Diagonal, Identity, Transpose & Orthogonal Matrix

1. Diagonal Matrix

Definition

A **diagonal matrix** is a square matrix where **all off-diagonal elements are zero**. Only the main diagonal may have non-zero values.

$$D = \begin{bmatrix} d_1 & 0 & 0 \\ 0 & d_2 & 0 \\ 0 & 0 & d_3 \end{bmatrix}$$

Example 01 : *If* $A = \begin{bmatrix} -3 & 0 \\ 0 & -3 \end{bmatrix}$ *and* $B = \begin{bmatrix} 8 & 0 \\ 0 & 5 \end{bmatrix}$ *, then* $AB = \begin{bmatrix} -24 & 0 \\ 0 & -15 \end{bmatrix}$

Example 02 *: If* $A = \begin{bmatrix} -5 & 0 \\ 0 & 11 \end{bmatrix}$ *and* $B = \begin{bmatrix} 7 & 0 \\ 0 & -13 \end{bmatrix}$ *are two diagonal matrices, then*

$$A + B = B + A = \begin{bmatrix} 2 & 0 \\ 0 & -2 \end{bmatrix}$$

$$AB = BA = \begin{bmatrix} -35 & 0 \\ 0 & -143 \end{bmatrix}$$

- **Main diagonal:** d_1, d_2, d_3
- Off-diagonal elements = 0

Properties

- Easy to compute **inverse** (if $d_i \neq 0$)
- Determinant = product of diagonal entries: $\det(D) = d_1 d_2 \dots d_n$
- Eigenvalues = diagonal entries

ML Applications

- Covariance matrices in **PCA** can be diagonalized

- Scaling features in **linear models**

2. Identity Matrix

Definition

An **identity matrix** $I_n$is a square matrix with **1s on the diagonal** and **0s elsewhere**:

$$I_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Properties

- Acts like **1** in multiplication: $AI = IA = A$
- Determinant = 1
- Inverse of identity = identity itself

ML Applications

- Used as **neutral element** in linear algebra operations
- **Regularization**: $(X^T X + \lambda I)^{-1} X^T y$in Ridge Regression

Identity vs Diagonal Matrix

The difference between a diagonal matrix and an identity matrix is outlined below:

Identity Matrix	Diagonal Matrix
A special diagonal matrix where all diagonal entries are 1.	A square matrix where all off-diagonal elements are zero.
All diagonal entries are 1.	Can be any numbers (zero, positive, negative, fractions).
Example: $A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	Example: $A = \begin{bmatrix} 4 & 0 & 0 \\ 0 & 5 & 0 \\ 0 & 0 & 6 \end{bmatrix}$
Acts as the "do nothing" matrix: AI = IA = A.	Scales rows/columns by the diagonal entries.

3. Transpose of a Matrix

Definition

The **transpose** of a matrix $A$is denoted A^T, obtained by **flipping rows into columns**:

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \Rightarrow A^T = \begin{bmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{bmatrix}$$

Properties

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

ML Applications

- Dot product: $x \cdot y = x^T y$
- Covariance: $\Sigma = \frac{1}{n} X^T X$
- Backpropagation gradients: $\frac{\partial L}{\partial W} = X^T \frac{\partial L}{\partial Z}$

4. Orthogonal Matrix

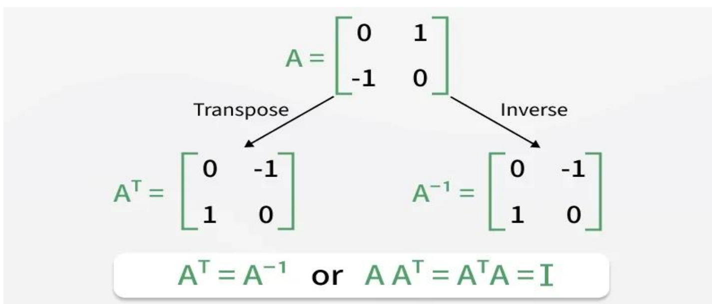
Definition

A square matrix $Q$is **orthogonal** if:

$$Q^T Q = QQ^T = I$$

- Columns (and rows) are **orthonormal vectors**
- Norm = 1, Dot product = 0

$$Q = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, Q^T Q = I$$



Example 1: Prove orthogonal property that multiplies the matrix by transposing results into an identity matrix if A is the given matrix.

Solution:

$$A = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$Thus,\ A^T = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

$$\Rightarrow A \cdot A^T = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Which is an identity matrix.

Thus, A is an Orthogonal Matrix.

Properties

- Inverse = transpose: $Q^{-1} = Q^T$
- Determinant = ±1
- Preserves **lengths and angles** (rotation/reflection)

ML Applications

- PCA: orthogonal **eigenvectors**
- QR decomposition: $A = QR$
- Dimensionality reduction, SVD

5. Python / NumPy Examples

```
import numpy as np

# 1. Diagonal Matrix
D = np.diag([2, 3, 4])
print("1. Diagonal Matrix:\n", D)
print("-"*40)

# 2. Identity Matrix
I = np.eye(3)
print("2. Identity Matrix:\n", I)
print("-"*40)

# 3. Transpose
A = np.array([[1, 2, 3],
              [4, 5, 6]])
print("3. Original Matrix A:\n", A)
print("3. Transpose of A:\n", A.T)
print("-"*40)

# 4. Orthogonal Matrix using QR decomposition
# Generate a random 3x3 matrix
random_matrix = np.random.rand(3,3)

# QR decomposition
Q, R = np.linalg.qr(random_matrix)
print("4. Orthogonal Matrix Q (from QR decomposition):\n", Q)

# Verify Q is orthogonal: Q^T * Q = I
print("Check Q^T @ Q:\n", np.round(Q.T @ Q, 5))
print("-"*40)
```

```
# Optional: Multiplication demo

# Diagonal * Identity
print("Diagonal * Identity:\n", D @ I)

# Transpose * Original
print("A^T * A:\n", A.T @ A)
```

6. Summary Table

Matrix Type	Notation	Key Property	ML Use Case
Diagonal	D	Off-diagonal = 0	Covariance PCA, scaling
Identity	I	I × A = A	Regularization, Neutral element
Transpose	A^T	Flip rows & columns	Dot products, Covariance, Gradients
Orthogonal	Q	Q^T Q = I	PCA, QR decomposition, SVD