LU Decomposition: Solving Large-Scale Linear Systems in ML

Name: Shreyash Lal

University Roll: 13030824012

Paper Name: Linear Algebra

Paper Code: BSCAIML301

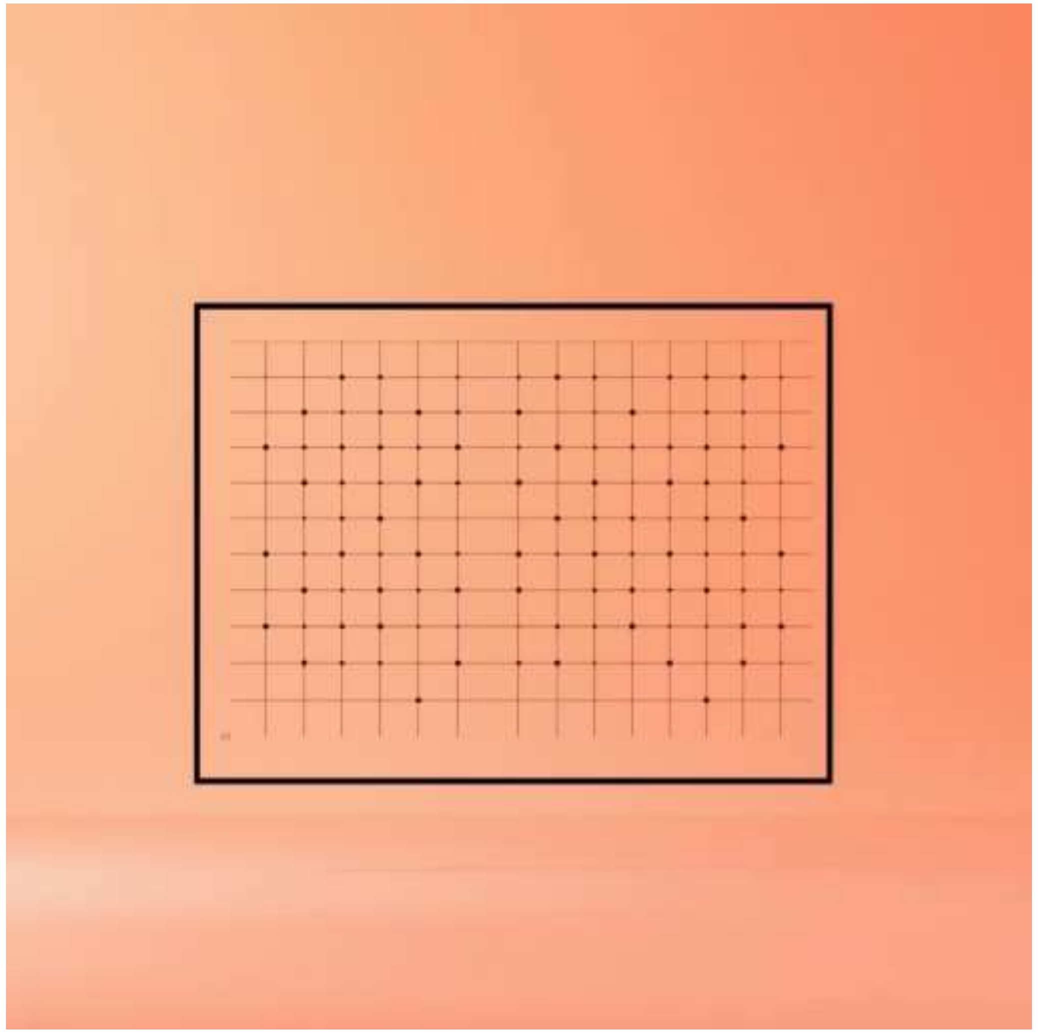
Dept: CSE(AI & ML) - A

Sem: 3rd

What is LU Decomposition?

LU Decomposition is a fundamental matrix factorization technique that breaks down a square matrix A into the product of a lower triangular matrix L and an upper triangular matrix U.

This process simplifies complex linear systems by transforming them into a sequence of easier-to-solve triangular systems. It's essential for efficient computation in numerical linear algebra.



The Mathematical Concept

LU Decomposition expresses a matrix A as:

$$A = LU$$

- A: The original square matrix.
- L: A lower triangular matrix with ones on the main diagonal.
- U: An upper triangular matrix.

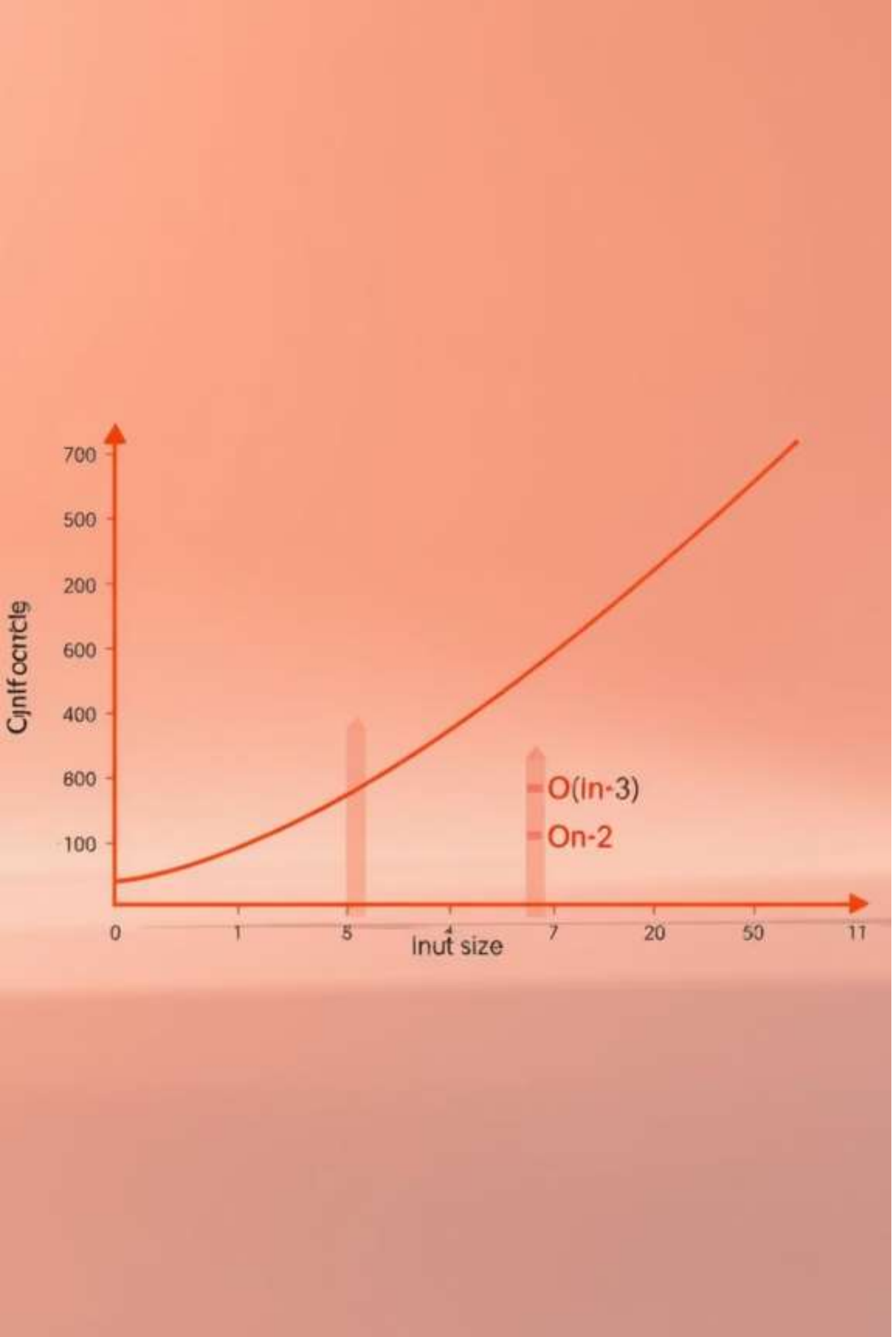
Step 1: Forward Substitution

Solve $Ly = b$ for y.

Step 2: Backward Substitution

Solve $Ux = y$ for x.

This two-step process replaces direct $Ax = b$ solving, which can be computationally intensive.



Computational Advantages

LU Decomposition offers significant efficiency for solving multiple linear systems with the same coefficient matrix.

1

Efficiency

Once A is decomposed, solving Ax=b for different b vectors is much faster, requiring O(n^2) operations instead of O(n^3).

2

Numerical Stability

Pivoting (row exchanges) can be incorporated to enhance numerical stability, especially when dealing with ill-conditioned matrices.

This makes it ideal for iterative algorithms where the system matrix remains constant.

Application in Machine Learning

LU Decomposition is a cornerstone in various ML applications, particularly where large linear systems arise.

Optimization

Used in Newton's method for solving non-linear optimization problems, crucial for training complex models.

Linear Regression

Efficiently solves normal equations for finding optimal regression coefficients in large datasets.

Numerical Stability

Enhances the robustness of algorithms handling large, sparse matrices common in deep learning and NLP.

Model Inversion

Useful for computing inverses of Hessian matrices in second-order optimization algorithms.

Case Study: Solving Ax = b

Consider the system:

$$\begin{bmatrix} 2 & 1 \\ 4 & 3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 5 \\ 11 \end{bmatrix}$$

Decomposition

A is decomposed into:

$$L = \begin{bmatrix} 1 & 0 \\ 2 & 1 \end{bmatrix}, U = \begin{bmatrix} 2 & 1 \\ 0 & 1 \end{bmatrix}$$

Forward Substitution

Solve Ly = b for y:

$$\begin{bmatrix} 1 & 0 \\ 2 & 1 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} 5 \\ 11 \end{bmatrix}$$

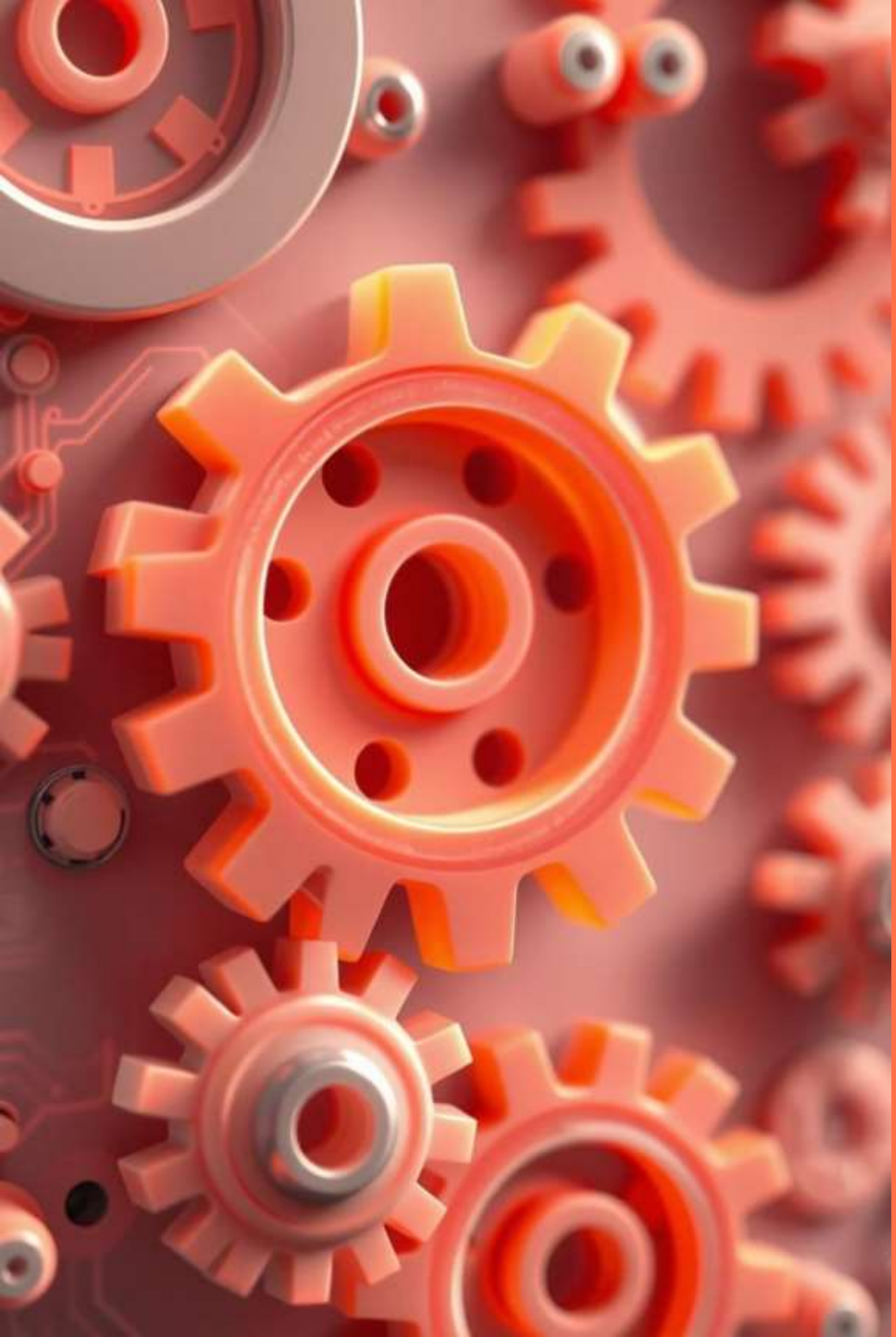
$$y = \begin{bmatrix} 5 \\ 1 \end{bmatrix}$$

Backward Substitution

Solve Ux = y for x:

$$\begin{bmatrix} 2 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 5 \\ 1 \end{bmatrix}$$

$$x = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$$



Conclusion & References

Key Takeaways

- LU decomposition simplifies linear systems.
- Offers significant computational efficiency for large-scale problems.
- Crucial for optimization, regression, and numerical stability in ML.

Further Reading

- Golub, G. H., & Van Loan, C. F. (2013). Matrix Computations (4th ed.). Johns Hopkins University Press.
- Strang, G. (2016). Linear Algebra and Learning from Data. Wellesley-Cambridge Press.

Thank you!