

Applications of Matrices in Artificial Intelligence: A Mathematical Perspective

Rahul A. Parshuramkar and Arpit A. Kharwade*

Assistant Professor, Mahatma Gandhi Arts, Science and Late N. P. Commerce College, Armori Dist-Gadchiroli, MS, India

*Corresponding author Email:

Manuscript Details

Received :13.06.2026

Accepted: 14.06.2026

Published: 30.06.2026

Available online on <https://www.irjse.in>

ISSN: 2322-0015

Cite this article as:

Rahul A. Parshuramkar and Arpit A. Kharwade.
Applications of Matrices in Artificial Intelligence:
A Mathematical Perspective, *Int. Res. Journal of
Science & Engineering*, 2026, Volume 14(3): 163-
169

<https://doi.org/10.5281/zenodo.20751956>



Open Access This article is licensed
under a Creative Commons Attribution

4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this license, visit <http://creativecommons.org/licenses/by/4.0/>

Abstract

Artificial intelligence has emerged as one of the most transformative technologies of the twenty-first century, yet beneath its remarkable capabilities lies a relatively modest mathematical foundation. This article examines the pivotal role of matrices as the fundamental mathematical infrastructure underpinning modern AI systems. We explore how matrices provide a natural and efficient framework for data representation, transformation, and optimization across diverse AI paradigms, including neural networks, deep learning architectures, and advanced models such as transformers and graph neural networks. Through analysis of matrix operations including multiplication, eigenvalue decomposition, and the singular value decomposition, we demonstrate that linear algebra offers both the computational efficiency and mathematical rigor essential for contemporary AI. The article further examines recent innovations including matrix transformations for encrypted learning, low-rank approximations for computational efficiency, and neural network approaches to matrix problem-solving. We conclude that matrices constitute not merely a convenient tool but the indispensable mathematical language through which artificial intelligence is articulated, implemented, and advanced.

Keywords: Matrices, Linear Algebra, Artificial Intelligence, Neural Networks, Matrix Multiplication, Singular Value Decomposition, Deep Learning

1. Introduction

The extraordinary progress in artificial intelligence over the past decade has captured the imagination of researchers, policymakers, and the public alike. From large language models capable of generating human-like text to systems that surpass human

performance in image recognition and strategic games, AI has demonstrated capabilities that seemed implausible only a few years ago. What is perhaps most remarkable about this progress is that it rests upon a mathematical foundation that has been well-understood for centuries: linear algebra and, more specifically, matrices [1].

As researchers have observed, "All AI techniques fundamentally rely on four core components: data, optimization methods, statistical intuition, and linear algebra" [2]. Among these, linear algebra provides the structural framework within which the other components operate. Matrices offer a concise and elegant language for representing the vast quantities of data that modern AI systems process, for describing the transformations that these systems learn, and for implementing the optimization procedures that enable learning itself [3].

The relationship between matrices and artificial intelligence is not merely incidental but foundational. When we examine the internal workings of a neural network, we find that all parameters are stored in matrices, and the computation of outputs from inputs is achieved through matrix multiplication. When we seek to reduce the dimensionality of high-dimensional data, techniques such as principal component analysis and the singular value decomposition—both fundamentally matrix-based—provide the mathematical machinery. When we consider recent advances in efficient AI, we find that innovations such as low-rank approximation and linear attention are rooted in elementary linear algebra concepts.

This article aims to provide a comprehensive mathematical exploration of matrix applications in artificial intelligence. We shall examine how matrices serve as the natural data structures for AI, how matrix operations implement the core computations of neural networks, and how advanced matrix decompositions enable sophisticated AI capabilities. We shall also explore emerging directions including matrix transformations for privacy-preserving AI and neural network approaches to matrix problem-solving. The

article does not claim that matrix algebra alone explains AI's capabilities—optimization, statistics, and algorithmic innovation all play essential roles—but rather that matrices constitute the indispensable mathematical infrastructure upon which these capabilities are built.

2.1 Matrices as Data Structures

At its most basic level, a matrix is a rectangular array of numbers arranged in rows and columns. This simple structure provides an exceptionally versatile framework for representing the data that AI systems process [4].

A matrix A of size $m \times n$ has m rows and n columns, with individual elements denoted a_{ij} , where i represents the row index and j the column index. In AI applications, this structure accommodates various forms of data representation:

- **Feature Vectors:** Each data point can be represented as a vector (a matrix with one row or one column), where each component corresponds to a specific attribute or feature of the data point. This vectorization simplifies mathematical representation and manipulation of features [5].
- **Datasets:** A collection of data points is naturally represented as a matrix, where rows correspond to individual instances and columns correspond to features. This representation enables efficient storage, manipulation, and analysis of entire datasets.
- **Model Parameters:** In machine learning models, particularly linear models, the parameters (weights) are represented as vectors or matrices. The dot product of the feature vector and the weight vector forms the basis of many algorithms, representing the influence of each feature.

2.2 Matrix Operations and Their AI Significance

Several fundamental matrix operations are essential to AI computations:

Matrix Multiplication: This operation is perhaps the most computationally significant in AI. If A is an $m \times n$ matrix and B is an $n \times p$ matrix, their

product AB is an $m \times p$ matrix, with each entry computed as the sum of products of corresponding elements. Matrix multiplication enables the transformation of input data through the layers of a neural network, where each layer applies a linear transformation followed by a nonlinear activation [5].

Matrix Transpose: The transpose of a matrix, denoted A^T , is obtained by swapping rows with columns. This operation is essential for various optimization algorithms and for computing covariance matrices in statistical methods.

Identity Matrix: The identity matrix I , with ones on the main diagonal and zeros elsewhere, satisfies $AI = IA = A$ for any compatible matrix A . In neural networks, the identity matrix serves as the starting point for various optimization procedures.

Determinant: The determinant of a square matrix provides information about the matrix's properties, including whether it is invertible. For a 2×2 matrix $\begin{bmatrix} a & b \\ c & d \end{bmatrix}$, the determinant is $ad - bc$.

2.3 Eigenvalues and Eigenvectors

Eigenvalues and eigenvectors are among the most powerful concepts in matrix algebra, with profound implications for AI.

For a square matrix A , an eigenvector v and its corresponding eigenvalue λ satisfy:

$$Av = \lambda v$$

This equation means that the linear transformation represented by A scales the eigenvector by the eigenvalue without changing its direction. The characteristic equation for finding eigenvalues is:

$$\det(A - \lambda I) = 0$$

In AI, eigenvalues and eigenvectors enable:

- **Dimensionality Reduction:** Principal component analysis (PCA) uses the eigenvectors of the covariance matrix to identify the directions of maximum variance in data

- **Understanding Transformations:** Eigenvalues reveal how a linear transformation stretches or compresses space along its principal axes
- **Stability Analysis:** The eigenvalues of the Jacobian matrix in neural network training provide insights into convergence behavior

3. Matrices in Neural Networks

3.1 The Neural Network Architecture as Matrix Operations

An artificial neural network consists of layers of nodes: an input layer, one or more hidden layers, and an output layer. Each node (artificial neuron) connects to nodes in subsequent layers with associated weights and thresholds. This architecture translates naturally into matrix operations.

Consider a simple two-layer neural network. The input vector X is multiplied by a weight matrix W to produce the output Y :

$$Y = WX$$

For a network with two input nodes and two hidden nodes, the weight matrix captures all connections:

$$\begin{pmatrix} W_{11} & W_{12} \\ W_{21} & W_{22} \end{pmatrix} \cdot \begin{pmatrix} X_1 \\ X_2 \end{pmatrix} = \begin{pmatrix} O_1 \\ O_2 \end{pmatrix}$$

This matrix formulation reveals that all the computations in a neural network can be expressed through matrix multiplication, enabling efficient implementation on parallel hardware such as GPUs.

3.2 Activation Functions and Matrix Operations

The simple matrix multiplication $Y = WX$ is a linear transformation. If we combined multiple linear transformations, the result would itself be a linear transformation, limiting the expressive power of the network. This is where activation functions become essential.

Activation functions introduce nonlinearity into the network, enabling it to learn complex patterns. Common activation functions include:

- **Step Function:** Outputs 0 when the input is less than or equal to 0, and 1 when the input is greater than 0

- **Sigmoid Function:** Maps inputs to values between 0 and 1, approaching 1 for large positive inputs and 0 for large negative inputs
- **ReLU (Rectified Linear Unit):** Outputs 0 for negative inputs and the input value for positive inputs

The activation function is applied after the matrix multiplication, so the forward propagation through a layer can be expressed as:

$$Y = f(WX + b)$$

where f is the activation function applied elementwise, and b is a bias vector.

3.3 Backpropagation and Matrix Calculus

The training of neural networks relies on backpropagation, an algorithm that computes gradients of the loss function with respect to network parameters. These gradients are then used to update parameters through gradient descent. Matrix calculus provides the mathematical framework for backpropagation.

If we have a loss function L and a weight matrix W in a layer, the gradient $\frac{\partial L}{\partial W}$ is a matrix of the same dimensions as W . This gradient indicates how the loss would change with small changes in each weight. The update rule becomes:

$$W_{new} = W_{old} - \eta \frac{\partial L}{\partial W}$$

where η is the learning rate.

The chain rule, expressed through matrix operations, enables efficient propagation of gradients backward through the network. This is why understanding matrix differentiation is essential for understanding neural network training.

4. Advanced Matrix Techniques in Deep Learning

4.1 The Singular Value Decomposition

The singular value decomposition (SVD) is perhaps the most powerful matrix factorization technique, with extensive applications in AI. For any $m \times n$ matrix A , the SVD is:

$$A = U\Sigma V^T$$

where U is an $m \times m$ orthogonal matrix, Σ is an $m \times n$ diagonal matrix with nonnegative singular values, and V^T is the transpose of an $n \times n$ orthogonal matrix.

SVD applications in AI include:

- **Dimensionality Reduction:** By retaining only the largest singular values and corresponding singular vectors, SVD provides the optimal low-rank approximation to a matrix (Eckart-Young theorem)
- **Recommendation Systems:** Matrix factorization techniques used in collaborative filtering are variants of SVD
- **Noise Reduction:** SVD can separate signal from noise in data

4.2 Low-Rank Approximations and Efficient AI

Recent advances in efficient AI have relied heavily on low-rank matrix approximations. The LoRA (Low-Rank Adaptation) technique, for example, "enables practitioners to cut down training and inference costs by orders of magnitude without sacrificing accuracy - but the idea exploits a simple low-rank approximation".

Similarly, the development of "Linear Attention" mechanisms has exploited "a clever idea based on approximating a nonlinear mapping by a linear one, reducing the cost from $O(n^2)$ to $O(n)$ ". These innovations demonstrate that "almost all the important recent contributions to improved AI algorithms are rooted in simple [linear algebra] ideas".

4.3 Transformers and Attention

The transformer architecture, introduced in the landmark paper "Attention is All You Need," has become the foundation of modern large language models. The core concept of attention can be expressed through matrix operations.

In the scaled dot-product attention mechanism, for a query matrix Q , key matrix K , and value matrix V :

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

The product QK^T is a matrix multiplication, and softmax is applied to each row. This matrix

formulation enables efficient computation of attention scores across all pairs of elements in a sequence .

4.4 Graph Neural Networks and Matrix Representations

Graph neural networks (GNNs) represent a significant area of AI research where matrices play a central role. In a GNN, the graph structure is represented by an adjacency matrix A , and node features by a feature matrix X . A graph convolutional layer can be expressed as :

$$H^{(l+1)} = \sigma \left(D^{-\frac{1}{2}} A D^{-\frac{1}{2}} H^{(l)} W^{(l)} \right)$$

where D is the degree matrix, $H^{(l)}$ is the node feature matrix at layer l , $W^{(l)}$ is the weight matrix, and σ is an activation function.

This matrix formulation shows how GNNs propagate information through the graph structure using matrix multiplication, with the adjacency matrix determining which node pairs interact .

5. Matrix Transformations and Computational Efficiency

5.1 Matrix Transformations for Efficient AI

Recent research has explored novel matrix transformation techniques to enhance the efficiency and security of AI systems. One study introduced "a new matrix transformation technique that ensures compatibility with both homomorphic encryption algorithms and AI model weight matrices, significantly improving computational efficiency" .

The transformation approach achieved "a reduction of up to 90% in data storage and a tenfold increase in speed compared to models that do not use the matrix transformation method" . This demonstrates that careful attention to matrix structure can yield substantial improvements in AI system performance.

5.2 Neural Networks for Matrix Problem-Solving

The relationship between matrices and neural networks is bidirectional: neural networks, which rely on matrix

operations, are increasingly used to solve matrix problems. This has led to the development of specialized neural network architectures for matrix computations.

Zeroing neural networks (ZNNs), for example, are specifically designed for real-time matrix problem-solving. Recent research has introduced anti-noise integral zeroing neural networks for matrix factorization that "can tolerate time-variant bounded vanishing and nonvanishing noises when exposed to external noise" . These models achieve "a root mean square error reduction of 92.6249% compared to scenarios without the proposed method" [7].

This integration of neural networks with matrix computations represents a powerful synergy: neural networks leverage matrix algebra for their operations, and in turn, neural networks are used to solve matrix problems more efficiently and robustly.

6. Discussion

6.1 The Primacy of Matrices in AI

The evidence presented in this article supports a strong claim: matrices provide the fundamental mathematical infrastructure for artificial intelligence. This primacy arises from several factors:

Natural Representation: Matrices offer a natural and efficient way to represent the structured data that AI systems process. Feature vectors, datasets, and network parameters all find natural expression as matrices .

Computational Efficiency: Matrix multiplication, the core operation of neural networks, can be efficiently implemented on modern hardware, particularly GPUs designed for parallel linear algebra operations .

Mathematical Rigor: The well-developed theory of matrix algebra provides rigorous foundations for understanding AI systems—their capabilities, limitations, and behavior.

Unifying Framework: Matrix algebra unifies diverse AI approaches – neural networks, SVMs, principal component analysis, and graph-based methods – under a common mathematical framework[8].

6.2 Implications for AI Research and Education

The matrix-centric nature of AI has important implications for research and education:

Numerical Linear Algebra Expertise: The observation that "almost all the important recent contributions to improved AI algorithms are rooted in simple linear algebra ideas" suggests that researchers with expertise in numerical linear algebra are well-positioned to contribute to AI advancement.

Educational Priorities: For students seeking to understand AI at a deep level, a thorough grounding in matrix algebra is essential. This suggests that linear algebra should receive substantial attention in AI curricula.

Interdisciplinary Opportunities: The intersection of numerical linear algebra and AI represents a fertile area for interdisciplinary research, with opportunities for contributions from mathematics, computer science, and engineering.

6.3 Challenges and Limitations

Despite their central role, matrix-based approaches face challenges:

Scalability: The computational demands of matrix operations in large-scale AI systems are enormous, driving research into more efficient algorithms and specialized hardware [8].

Interpretability: While matrix operations are mathematically transparent, the complex transformations learned by deep networks remain difficult to interpret. This creates tension between mathematical rigor and practical explainability.

Limitations of Linear Algebra: Matrix operations are linear, while many AI challenges are fundamentally nonlinear. The reliance on matrix algebra requires

careful integration with nonlinear activation functions and other mechanisms.

7. Conclusion

This article has examined the applications of matrices in artificial intelligence from a mathematical perspective. We have seen that matrices provide the natural data structures for representing features, datasets, and model parameters; that matrix multiplication implements the core computations of neural networks; and that advanced matrix decompositions such as the singular value decomposition enable sophisticated AI capabilities.

The relationship between matrices and AI is both deep and bidirectional. Neural networks rely on matrix operations for their implementation, while neural networks themselves are increasingly used to solve matrix problems more efficiently and robustly. Recent innovations in efficient AI, including low-rank approximations and linear attention mechanisms, demonstrate that progress in AI often comes from new applications of established matrix algebra concepts.

As numerical linear algebra researchers have observed, "the NLA community as a whole has been rather slow in embracing AI research" despite the fact that "almost all the important recent contributions to improved AI algorithms are rooted in simple NLA ideas". This suggests significant opportunities for interdisciplinary contributions at the intersection of matrix algebra and artificial intelligence.

In conclusion, matrices constitute not merely a convenient tool but the indispensable mathematical language through which artificial intelligence is articulated, implemented, and advanced. As AI continues to evolve, the fundamental importance of matrix algebra is likely to increase, not diminish.

Conflicts of interest: The authors stated that no conflicts of interest.

Correspondence and requests for materials should be addressed to Arpit A. Kharwade

Peer review information : IRJSE thanks the anonymous reviewers for their contribution to the peer review of this work. A peer review file is available.

Reprints and permissions information is available at

<https://www.irjse.in/reprints>

References

1. Jin K, Ma Q, Jia H. Applications of Linear Algebra in Artificial Neural Networks. In: Proceedings of the 2024 4th International Conference on Computational Modeling, Simulation and Data Analysis (CMSDA 2024). Hangzhou, China; 2024.
2. Hawkins J. Matrix Algebra. In: Mathematics for Artificial Intelligence. Taylor & Francis; 2026.
3. Deep learning, transformers and graph neural networks: a linear algebra perspective. Numerical Algorithms. 2025;100:2095-2134.
4. Rani K. Linear Algebra as the Mathematical Foundation of Artificial Intelligence: Concepts, Applications, and Future Prospects. International Journal of Engineering Science and Humanities. 2024;14(1).
5. Efficient Artificial Intelligence With Novel Matrix Transformations and Homomorphic Encryption. IEEE Transactions on Artificial Intelligence. 2024.
6. Yang Y, et al. A novel real-time noise-resilient zeroing neural network and its applications to matrix problem solving. Neural Networks. 2025.
7. A Matrix Algebra Approach to Artificial Intelligence. MPG.eBooks; 2024.
8. al-Masri AM, Abothher AD, damak H. Application of linear Algebra in enhancing artificial intelligence techniques. In: Proceedings of the 3rd International Scientific Conference on Science and Technology. Libya; 2025.

© 2026 | Published by IRJSE

Publisher's Note

IJLSCI remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.