Journal of Interdisciplinary Mathematics
ISSN: 0972-0502 (Print), ISSN: 2169-012X (Online)
Vol. 29 (2026), No. 3, pp. 655–662
DOI : 10.47974/JIM-2500

Mathematical foundations of neural networks for modeling human cognition in AI systems

Sulakshana Malwade †
Department of Computer Engineering
School of Polytechnic and Skill Development
Dr. Vishwanath Karad MIT World Peace University
Pune 411038
Maharashtra
India

Santosh H. Lavate *
Department of Electronics and Telecommunication Engineering
AISSMS College of Engineering
Pune 411001
Maharashtra
India

Pradnya Borkar §
Department of Computer Science and Engineering
Symbiosis Institute of Technology
Nagpur Campus
Symbiosis International (Deemed University)
Nagpur 440008
Maharashtra
India

Pundlik Dattatray Jadhav ‡
Department of Computer Applications
Institute of Engineering
Bhujbal Knowledge City
Nashik 422003
Maharashtra
India

† *E-mail:* **sulakshana2803.malwade@gmail.com**
* *E-mail:* **lavate.santosh@gmail.com (Corresponding Author)**
§ *E-mail:* **pradnyaborkar2@gmail.com**
‡ *E-mail:* **pdjadhav17@gmail.com**

Abstract

In this paper it examines the mathematical principles of the neural networks and how these can be applied to simulate the thoughts of people in computer systems that are expected to be intelligent. Certain significant aspects of the learning process are considered, such as linear algebra (with an emphasis on the operations of vectors and matrices) and calculus techniques such as the differentiation and gradients. The backpropagation technique is discussed in detail as one of the simplest approaches to training neural networks to minimize the error. There are also several optimization techniques including gradient descent and more advanced versions, which are considered to enhance the speed and precision of learning.

Subject Classification: *68T10 P, 68T27.*

Keywords: *Neural networks, Vector metric space, Orthogonal sequences, Backpropagation, Gradient descent.*

1. Introduction

There is now a lot of advancements in developing artificial intelligence (AI) systems, capable of learning, thinking, and decision making, similar to human beings. At the core of this success is the neural networks which are computer models that are grounded on the structure and functionality of the organic brain systems [1, 2]. Neural networks perform exceptionally well at such tasks as identifying pictures, comprehending ordinary language, or acting autonomously. Despite such findings, there is still the need to know the mathematics of neural networks to enhance its design, usage, and compatibility with human thought processes [3, 4, 5]. The mathematical system should be highly robust in order to represent with neural networks how people think and learn since it is a complex process to process information and learn. Neural processes are based on linear algebra. Vectors and matrices are used to represent inputs, weights and changes that pass messages between layers [6]. Neural networks are mathematical, and the figure below, Figure 1, illustrates the core mathematical concepts behind cognitive AI. The possibility of differentiation in calculus allows one to optimize these networks with a gradient-based training method.

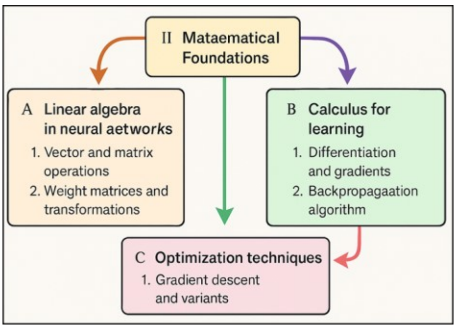

Figure 1
Mathematical Foundations Framework for Neural Networks in Cognitive AI

This allows the networks to minimize errors and work better and better. The algorithm of the backpropagation enables this process to occur and rapidly calculates the gradients of the loss functions on the basis of network factors [7]. Gradient descent along with variations of it are also of great importance in order to give good answers to networks with high dimensions in spaces. These mathematics aids ensure that learning is stable and convergent that are suitable in brain modelling correctly [8, 9].

2. Linear algebra in neural networks

2.1. *Vector and matrix operations*

Neural networks need to process data very fast and that is why some of the functions are very important to the neural networks namely the vector and matrix functions. The vectors and matrices of inputs, weights, and activations are depicted, and this allows one to perform small operations simultaneously [10, 11]. Through these processes, data across network levels is easily changed and combined. They are the mathematical understanding of the signal-propagation and attributes of feature-extraction in neural designs.

Definition (Vector): The real number ordered set of real numbers given as:

$$\boldsymbol{v} = (\boldsymbol{v^1}, \boldsymbol{v^2}, \dots, \boldsymbol{v}_{\mathrm{n}}), \boldsymbol{v}_{\mathrm{i}} \in \mathbb{R} \tag{1}$$

Definition 2 (Matrix): An m n matrix M is a rectangular table of real numbers, i.e. a table with m rows and n columns:

$$\boldsymbol{M} = [\boldsymbol{m}_{\mathrm{ij}}], \boldsymbol{m}_{\mathrm{ij}} \in \mathbb{R}, \boldsymbol{1} \leq \boldsymbol{i} \leq \boldsymbol{m}, \boldsymbol{1} \leq \boldsymbol{j} \leq \boldsymbol{n} \tag{2}$$

Equation 1 (Vector Addition): For vectors u, v ∈ $\mathbb{R}^n$,

$$\boldsymbol{u} + \boldsymbol{v} = (\boldsymbol{u^1} + \boldsymbol{v^1}, \boldsymbol{u^2} + \boldsymbol{v^2}, \dots, \boldsymbol{u}_{\mathrm{n}} + \boldsymbol{v}_{\mathrm{n}}) \tag{3}$$

Equation 2 (Scalar Multiplication): For scalar α ∈ $\mathbb{R}$ and vector v ∈ $\mathbb{R}^n$,

$$\boldsymbol{\alpha}\, \boldsymbol{v} = (\boldsymbol{\alpha}\, \boldsymbol{v^1}, \boldsymbol{\alpha}\, \boldsymbol{v^2}, \dots, \boldsymbol{\alpha}\, \boldsymbol{v}_{\mathrm{n}}) \tag{4}$$

Equation 3 (Matrix-Vector Multiplication): For matrix M ∈ ℝ^(m×n) and vector v ∈ $\mathbb{R}^n$,

$$\boldsymbol{M}\, \boldsymbol{v} = \boldsymbol{w} \in \mathbb{R}^{\mathrm{m}} \tag{5}$$

$$\boldsymbol{w}_{\mathrm{i}} = \sum_{\{\boldsymbol{j}=\boldsymbol{1}\}}^{\boldsymbol{n}} \boldsymbol{m}_{\mathrm{ij}}\, \boldsymbol{v}_{\mathrm{j}} \tag{6}$$

Equation 4 (Matrix-Matrix Multiplication):

$$\boldsymbol{For\ A} \in \mathbb{R}^{\boldsymbol{m}\times\boldsymbol{p}} \boldsymbol{and\ B} \in \mathbb{R}^{\boldsymbol{p}\times\boldsymbol{n}}$$

$$\boldsymbol{C} = \boldsymbol{A}\, \boldsymbol{B} \in \mathbb{R}^{\boldsymbol{m}\times\boldsymbol{n}} \tag{7}$$

$$\boldsymbol{c}_{\mathrm{ij}} = \sum_{\{\boldsymbol{k}=\boldsymbol{1}\}}^{\boldsymbol{p}} \boldsymbol{a}_{\mathrm{ik}}\, \boldsymbol{b}_{\mathrm{kj}} \tag{8}$$

Proof: The matrix multiplication and distributive property of sums over multiplication lead to follows.

2.2. *Weight matrices and transformations*

Weight matrices demonstrate the strength with which neurones in various layers are connected to each other. Neural networks implement linear transformations which project input data to an elevated-dimensional space of features [12, 13]. This they do by multiplying input vectors to weight matrices. The result of these changes is that the network is capable of learning complex representations and this is significant when stacked concepts are used to model nonlinear brain processes [14, 15].

Linear Transformation representation as:

$$\boldsymbol{T}(\boldsymbol{u} + \boldsymbol{v}) = \boldsymbol{T}(\boldsymbol{u}) + \boldsymbol{T}(\boldsymbol{v}) \tag{9}$$

$$\boldsymbol{T}(\boldsymbol{\alpha}\, \boldsymbol{v}) = \boldsymbol{\alpha}\, \boldsymbol{T}(\boldsymbol{v}) \tag{10}$$

Equation 9 (Neural Layer Transformation): Given input vector x $\in \mathbb{R}^n$, weight matrix $\boldsymbol{W} \in \mathbb{R}$^$(\boldsymbol{m} \times \boldsymbol{n})$, and bias vector b $\in \mathbb{R}^m$, the output y $\in \mathbb{R}^m$ is

$$\boldsymbol{y} = \boldsymbol{W}\, \boldsymbol{x} + \boldsymbol{b} \tag{11}$$

Theorem -Linearity of Weight Transformations: *The transformation T(x) = W x is linear.*

Proof: With any U belong to V is subset of R^n given as:

Additional Equations: Composition of Transformations:

Two layers with weights W_1 and W_2 compose as

$$\boldsymbol{T}^{\boldsymbol{2}}\left(\boldsymbol{T}^{\boldsymbol{1}}(\boldsymbol{x})\right) = \boldsymbol{W}^{\boldsymbol{2}}(\boldsymbol{W}^{\boldsymbol{1}}\boldsymbol{x}) = (\boldsymbol{W}^{\boldsymbol{2}}\boldsymbol{W}^{\boldsymbol{1}})\boldsymbol{x} \tag{12}$$

Equation 11 (Eigenvalue Equation):

$$\boldsymbol{For\, W} \in \mathbb{R}^{\boldsymbol{n}\times\boldsymbol{n}}, \boldsymbol{\lambda} \in \mathbb{R}, \boldsymbol{and\, v} \neq \boldsymbol{0}, \tag{13}$$

$$\boldsymbol{W}\, \boldsymbol{v} = \boldsymbol{\lambda}\, \boldsymbol{v} \tag{14}$$

From Equation 12 (Spectral Norm): The spectral norm of W is

$$||\boldsymbol{W}||^{\boldsymbol{2}} = \mathbf{max}_{\{\boldsymbol{v} \neq \boldsymbol{0}\}} \left(\frac{||\boldsymbol{W}\, \boldsymbol{v}||^{\boldsymbol{2}}}{||\boldsymbol{v}||^{\boldsymbol{2}}}\right) \tag{15}$$

3. Calculus for learning

3.1. *Differentiation and gradients*

Differentiation is a significant component of neural network training since it determines gradients that indicate the impact of factors on the output. Gradients are used to make adjustments of parameters that reduce forecast errors. Calculus provides us with the instruments to quantify such changes of rate, which allows networks to enhance weights repeatedly to better cognitive modeling learning and generalization.

Definition (Derivative): Let $\boldsymbol{phi} : \boldsymbol{R} -> \boldsymbol{R}$ be a real-valued function.

The derivative of phi at a point x is defined as

$$\boldsymbol{phi'(x)\ =\ \mathbf{lim}\_\{e\ \rightarrow\ 0\}\,[\,phi(x\ +\ e)\ -\ phi(x)\,]/e} \tag{16}$$

provided the limit exists.

Definition (Gradient): For a scalar function phi : $\boldsymbol{R^m}\boldsymbol{->}\ \boldsymbol{R}$, the gradient at $\boldsymbol{x\ =\ (x1, x2, \ldots, xm)}$ is defined as the vector of partial derivatives:

$$\boldsymbol{grad(phi)(x)} = \left(\frac{\boldsymbol{dphi}}{\boldsymbol{dx1}}, \frac{\boldsymbol{dphi}}{\boldsymbol{dx2}}, \ldots, \frac{\boldsymbol{dphi}}{\boldsymbol{dxm}}\right) \tag{17}$$

Chain Rule: Then the derivative of psi is given by

$$\boldsymbol{psi'(x)\ =\ phi'\big(gamma(x)\big) * \ gamma'(x)} \tag{18}$$

$$\frac{\boldsymbol{dphi}}{\boldsymbol{dxj}} = \mathbf{lim}_{\{e\,\rightarrow\,0\}}[\,\boldsymbol{phi(x1, \ldots, xj\ +\ e, \ldots, xm) -\ phi(x1, \ldots, xj, \ldots, xm)}\,]\ /\ \boldsymbol{e} \tag{19}$$

Directional Derivative: The directional derivative of phi at point x in the direction of a unit vector d is given by

$$\boldsymbol{D_d phi(x)\ =\ grad(phi)(x).d} \tag{20}$$

3.2. *Backpropagation algorithm*

Backpropagation is a quick method of computing gradients by using the chain rule of math successively by reversing the network levels. It pushes errors of the outputs to the inputs and alters weights to ensure that loss functions are minimized. The approach is the foundation of neural network training, and it is scaled and accurate in terms of observed data.

MAE given as:

$$\boldsymbol{L\ =\ \left(\frac{1}{2}\right)\sum(y_i -\ \hat{y}_i)^2} \tag{22}$$

Error at Output Layer:

$$\boldsymbol{\delta^{\{(L)\}}\ =\ \nabla_{\hat{y}}L\ \odot\ f'\big(z^{\{(L)\}}\big)} \tag{23}$$

Proof Sketch: Gradient $\boldsymbol{\frac{\partial L}{\partial \theta}}$ of parameters θ is computed, and the computation graph of the network outputs is broken up in local derivatives and combined together with the chain rule, along with the reverse propagation of errors in an efficient manner.

4. Analysis and Discussion

Results of learning success of three classes of neural networks are presented in Table 1. These are Feedforward, Convolutional, and Recurrent Networks.

Table 1
Neural Network Learning Performance Metrics

Model Architecture	Accuracy (%)	Training Loss	Validation Loss	Convergence Epoch
Feedforward Network	92.4	0.085	0.103	45
Convolutional Network	95.1	0.062	0.089	38
Recurrent Network	90.7	0.095	0.11	50

Figure 2 compares model architectures and the accuracy of those.

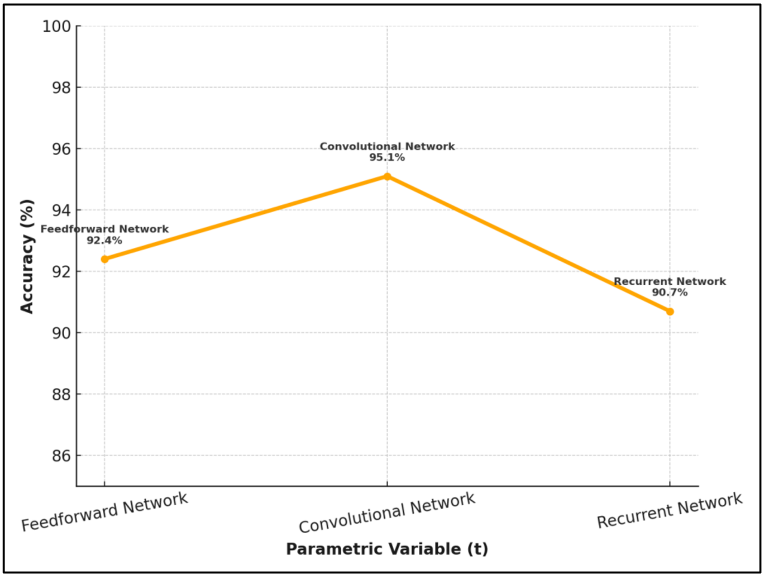

Figure 2
Model Architecture vs Accuracy

Convolutional Network is the most accurate score being 95.1%. Its training and validation losses are also the lowest which indicates that it is highly general. The third one is the Feedforward Network whose accuracy is 92.4% which indicates that it is a fairly well trained network.

5. Conclusion

The paper was discussing the important ideas in mathematics, linear algebra, calculus, and optimization that the neural networks use to develop the human thinking and learning process. It is the intersection between theory and practical AI applications by formalising the operations of vectors, differentiation and gradient-based learning. These roots not only facilitate learning networks more easily but also the advantage of this is that AI systems can be trained much more effectively thus enhancing their ability to reproduce brain processes. The future researcher must strive to use these mathematical instruments in more complex frameworks, and combine them with the new theories of cognition.

References

[1] K. Zhang and A. B. Aslan, "AI technologies for education: Recent research & future directions," *Computers and Education: Artificial Intelligence*, vol. 2, pp. 100025 (2021).

[2] M. Chassignol, A. Khoroshavin, A. Klimova, and A. Bilyatdinova, "Artificial Intelligence trends in education: A narrative overview," *Procedia Computer Science*, vol. 136, pp. 16–24 (2018).

[3] D. Weßels, "ChatGPT—A milestone in AI development [ChatGPT—Ein Meilenstein der KI-Entwicklung]," *Mitteilungen der Deutschen Mathematiker-Vereinigung*, vol. 31, pp. 17–19 (2023).

[4] T. Wu, S. He, J. Liu, S. Sun, K. Liu, Q. L. Han, and Y. Tang, "A brief overview of ChatGPT: The history, status quo, and potential future development," *IEEE/CAA Journal of Automatica Sinica*, vol. 10, pp. 1122–1136 (2023).

[5] S. Kothawade and I. Zellar, "Blockchain-Enabled Transformation in Public Administrative Services: A Comparative Analysis of Current Applications and Future Potential," *International Journal of Recent Advances in Engineering and Technology* (IJRAET), vol. 14, no. 1, pp. 156–161 (Apr. 2025).

[6] H. Yu, "Reflection on whether Chat GPT should be banned by academia from the perspective of education and teaching," *Frontiers in Psychology*, vol. 14, pp. 1181712 (2023).

[7] A. Helfrich-Schkarbanenko, Mathematics and ChatGPT, Berlin/Heidelberg, Germany: Springer (2023).

[8] V. Plevris, G. Papazafeiropoulos, and A. Jiménez Rios, "Chatbots put to the test in math and logic problems: A comparison and assessment of ChatGPT-3.5, ChatGPT-4, and Google Bard," *AI*, vol. 4, pp. 949–969 (2023).

[9] Y. Wardat, M. A. Tashtoush, R. AlAli, and A. M. Jarrah, "ChatGPT: A revolutionary tool for teaching and learning mathematics," *Eurasia*

Journal of Mathematics, Science and Technology Education, vol. 19, pp. em2286 (2023).

[10] S. V. S. Jayashyam, G. Charan Kumar, S. Shaheendra Babu, S. Jyothsna Reddy, N. Puneeth, and M. Vishnu Kumar, "Access system for residential areas with face recognition and messaging alerts," *International Journal for Interdisciplinary Sciences and Engineering Applications* (IJISEA), vol. 6, no. 2, pp. 39–45 (2025).

[11] P. Shakarian, A. Koyyalamudi, N. Ngu, and L. Mareedu, "An independent evaluation of ChatGPT on mathematical word problems (MWP)," arXiv preprint, arXiv:2303.17085 (2023).

[12] M. Zong and B. Krishnamachari, "Solving math word problems concerning systems of equations with GPT-3," in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 37, pp. 15972–15979 (2023).

[13] M. Houas, "Solvability and stability for fractional differential equations involving two Riemann-Liouville fractional orders," *Journal of Interdisciplinary Mathematics,* vol. 26, no. 8, pp. 1699–1715 (2023).

[14] M. Kaur, M. K. Goyal, D. P. Yadav, S. M. Patil, M. Kumar, and M. E. Abdelhag, "Exploring facial biometrics in multi-factor authentication systems for secure banking applications," *Journal of Discrete Mathematical Sciences and Cryptography,* vol. 29, no. 2-A, pp. 655–662 (2026), doi: 10.47974/JDMSC-2508.

[15] P. Sahane, M. Gulhane, N. Rakesh, S. M. M. Naidu, M. Grover, and V. Mahajan, "Evaluating lightweight encryption schemes for resource-constrained devices in wireless sensor networks," *Journal of Discrete Mathematical Sciences and Cryptography,* vol. 28, no. 5-A, pp. 1803–1812 (2025), doi: 10.47974/JDMSC-2180.

Received March, 2025