Kuwait Journal of Engineering Research
ISSN: 2984-8016
Vol. 3 No. 1 (2025)

Advances of AI in Mathematics: Overview of Operation Management

[1]Sanskriti Bhattarai, *[2]Suresh Kumar Sahani, [3]Manisha Jha

[1]Mithila Institute of Technology, Jankpurdham, Nepal

sanskritiofficial72@gmail.com

[2]Faculty of Science, Technology, and Engineering

Rajarshi Janak University, Nepal

sureshsahani@rju.edu.np

[3]Faculty of Management

Rajarshi Janak University, Nepal

Kmanisha2055@gmail.com

Abstract:

Artificial Intelligence, in today's context, is the most used tool in our day-to-day life. The main reason is its versatility and compatibility. When talking about AI, it is built and governed through mathematics. The mathematics behind AI deliberately makes use of general concepts like linear algebra, probability, and calculus. They serve as founding pillars in building AI tools. This research paper explores how mathematics constitutes the foundation of AI and the use of mathematical tools in AI.

Keywords: Artificial Intelligence, Mathematics, Linear Algebra, Vectors, Matrix, Calculus

Introduction

According to **Professor John McCarthy** -an American computer scientist and one of the founders of artificial intelligence, *"AI (Artificial intelligence) is the science and engineering of making intelligent machines, especially intelligent computer programs."* Artificial Intelligence (AI) has transformed modern technology, powering everything from smartphone assistants to self-driving cars. Behind every AI system lies the foundation of mathematical principles that enable machines to learn, recognize patterns, and make decisions. To understand it simply, let's take an example-When we unlock our phones through fingerprint or face recognition or when we use Gemini, Google assistant, Siri, ChatGPT or any such apps for performing simple (surfing information) to complex tasks (decision making), we're interacting with AI and mathematics is involved behind it.

Literature Review

Mathematics forms the foundation of artificial intelligence, with researchers emphasizing that mathematical principles are not merely supportive tools but constitute the essence of AI systems. Recent research shows that AI has emerged as a catalyst for profound advancements, revolutionizing human approaches to complex challenges. This review examines existing research across knowledge representation, natural language processing, and optimization, identifying key contributions and gaps. Research has consistently shown that concepts such as derivatives, gradient descent, vectors, matrices, optimization algorithms, embeddings, and Natural Language Processing (NLP) are central to AI development.

1. Derivatives & Gradient-Based Learning

Early studies by Rumelhart, Hinton, and Williams (1986) demonstrated that **derivatives** allow neural networks to learn by measuring how changes in weights affect model error. Their backpropagation algorithm established derivatives as the foundation of training deep neural networks. Later research emphasized that **gradient descent** uses these derivatives to move in the direction that reduces error most effectively (Bottou, 2010).

2. Vectors & Matrices in AI Computation

AI systems convert data into **vectors**, which allow mathematical operations such as similarity measurement, direction, and magnitude. The classic Vector Space Model developed by Salton et al. (1975) showed how text could be represented numerically. Matrices also play a crucial role in AI; Goodfellow, Bengio, and Courville (2016) explain that neural networks rely heavily on matrix multiplication to process many inputs simultaneously, enabling efficient computation.

3. Optimization Methods in AI

Beyond basic gradient descent, researchers have developed improved optimization techniques. Kingma and Ba (2015) introduced **Adam**, an adaptive optimization algorithm that accelerates learning and improves model stability. Optimization research continues to drive accuracy and efficiency in deep learning.

4. Embeddings & High-Dimensional Representations

A major advancement in AI came from **embeddings**, which convert words or data points into dense numerical vectors. Word2Vec by Mikolov et al. (2013) showed how AI could capture meaning and relationships through mathematics. Later methods like GloVe (Pennington et al., 2014) refined this idea by using matrix factorization to represent global word relationships.

5. NLP & Mathematical Architecture

Modern NLP relies heavily on mathematical operations. The Transformer architecture (Vaswani et al., 2017) introduced attention mechanisms built entirely on **matrix multiplications**, vector projections, and gradient-based optimization. This breakthrough enabled major advances in translation, text understanding, and language modeling.

Summary

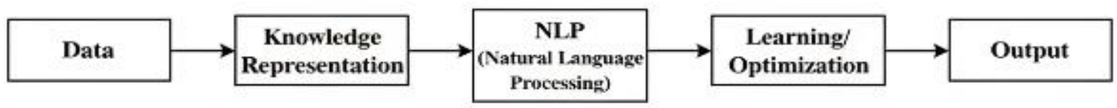
Across all studies, the literature consistently highlights that advancements in AI are fundamentally advancements in mathematics. Derivatives, vectors, matrices, optimization, embeddings, and NLP architectures collectively drive the rapid evolution of intelligent systems.

Mathematics in AI

To understand the complex relationship between mathematics and AI, we need to understand how AI works-step by step and then understand the role of mathematics in each step of AI working.

AI works by **learning from data** and using **mathematical models** to make decisions. The main steps are:

1. Knowledge Representation
2. Natural Language Processing
3. Learning/Optimization



1.Knowledge Representation (KR) – How AI "remembers and reasons."

Knowledge representation (KR) in AI refers to **encoding information about the world into formats that AI systems can utilize to solve complex tasks.** This process enables machines to reason, learn, and make decisions by structuring data in a way that mirrors human understanding.

Simply, Knowledge representation is simply the way AI systems store, organize and manipulate data.

Three common methods of Knowledge Representation are **logic**, **vectors**, and **matrices**.

Let us understand how AI represents information using "**Logic**"?

➔ The AI uses the **general rule** stored in its knowledge and applies it to a **fact**.
Suppose AI has stored a rule and a fact in its memory as:

Rule: Acids have pH less than 7.

Fact: HCl is an acid.

Symbolic Representation:

Let Acid(x) → x is an acid.

Let pH(x) < 7 → x has pH less than 7.

Rule: ∀x (Acid(x) → pH(x) < 7)

[“The symbol ∀ denotes ‘for all’ and is used to represent rules that apply to every element in a set. For example, ∀x (Acid(x) → pH(x)) means all acids have pH less than 7.”]

For example:

Let x equals HCl then it is represented as: -

Fact: Acid(HCl)

Since Acid(HCl) and Acid(x)→pH(x) < 7 ⇒ pH(HCl) < 7

AI combines the fact and the general rule to infer:

HCl has pH less than 7.

Explanation:

➔ This is how **knowledge representation allows AI to reason symbolically** and make logical conclusions.
➔ In this way, AI makes use of propositional logic to represent information.

Let us understand how AI represents information using “**Vectors”?**

➔ AI can represent concepts as **numerical feature vectors** to measure similarity.
Suppose we have two concepts:
Acid $\vec{A} = (2,3)$
HCl $\vec{H} = (3,4)$

Here, the numbers are just a **simple numerical representation** of properties, such as strength, acidity, or other features.

The **dot product** tells us how closely related two concepts are:

$$\vec{A}.\vec{H} = (2\times3) + (3\times4) = 6+12 = 18$$

Higher dot product → more closely related concepts.
Since Acid and HCl have a high dot product, AI understands that **HCl is closely related to the concept of acids**.

Angle between vectors gives a measure of similarity between 0 (completely different) and 1 (the same).

$$\text{Cos}\theta = \frac{\vec{A}.\vec{H}}{|\vec{A}|\,|\vec{H}|}$$

Where $\overrightarrow{|A|} = \sqrt{2^2+3^2} = \sqrt{13} \approx 3.606$

$|\vec{H}| = \sqrt{3^2+4^2} = \sqrt{25} = 5$

i.e. $\text{Cos}\theta = \frac{\vec{A}.\vec{H}}{|\vec{A}|\,|\vec{H}|} = \frac{18}{(3.606)(5)} \approx 0.99885$

Kuwait Journal of Engineering Research
ISSN: 2984-8016
Vol. 3 No. 1 (2025)

i.e. $\theta \approx 0.055$ radians $\approx 3.16^0$

Interpretation:

- ➔ The angle is **very small (≈3°)**, which means the vectors are almost pointing in the same direction.
- ➔ This tells us that **HCl is extremely similar to the general concept of acid** — just like we expect.
- ➔ In this example, AI "knows" that HCl is an acid because the vectors are closely aligned.

Explanation:

- ➔ The AI **represents the concepts numerically** so it can **compute relationships efficiently**.
- ➔ Vectors allow AI to reason about **similarity, categorization, or clustering**.

Let us understand how AI represents information using "**Matrix"?**

- ➔ AI can store relationships between concepts using a matrix, which makes reasoning fast and structured. AI can infer new facts using **matrix operations**.

  Let us create a relationship matrix(M) using the above fact and rule as:

  Matrix (M) :

$$
\begin{array}{cc}
 & \begin{array}{ccc} \text{HCl} & \text{Acid} & \text{pH<7} \end{array} \\
\begin{array}{c} \text{HCl} \\ \text{Acid} \\ \text{pH<7} \end{array} & \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}
\end{array}
$$

In this matrix,

M_{12} : HCl → Acid = 1 (HCl is an acid) [This is our **FACT**]

M_{23} : Acid → pH<7 = 1 (Acids have pH<7) [This is our **RULE**]

All other entries = 0 (no direct relationship)

To find if HCl has pH<7, we multiply the matrix by itself:

Formula: $M^2 = M \times M$

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

Result: Position [HCl, pH<7] = 1

Interpretation:

- ➔ The relationship matrix has a fact and rule stored in matrix form where 1 represents a relationship and 0 represents none.
- ➔ After multiplying matrix M with itself, we get that M_{13}= 1 which shows the relationship between HCl and pH<7.
- ➔ AI infers that since **HCl is an acid** and **acids have pH < 7, HCl must have pH < 7.**

Explanation:

- ➔ The matrix stores **all direct relationships** numerically.
- ➔ Multiplying matrices allows AI to **discover indirect connections**, i.e., deducing new facts.

2.Natural Language Processing (NLP) – How AI "understand and communicate with human language."

Natural language processing (NLP) is a field within computer science and artificial intelligence that enables computers to recognize, understand and generate text and speech in human understandable language by combining the study of linguistics with techniques such as statistical modeling, machine learning, and deep learning.

Simply, NLP is a branch of AI that focuses on making computers and AI capable of understanding, interpreting, and generating human language in a way that is meaningful.

Behind this linguistic ability lies a strong foundation of mathematical concepts, particularly in linear algebra, probability, and calculus.

Components of NLP

There are two components of NLP:

1. Natural Language Generation (NLG)
2. Natural Language Understanding (NLU)

Natural Language understanding (NLU):

Natural language understanding (NLU) is a subfield of Natural language Processing (NLP) that deals with machine reading comprehension. It primarily focuses on the interpretation of input texts into a form that is easily understood and processed by AI models. It includes:

i. *Text preprocessing*: -

Text preprocessing is the initial step in Natural Language Processing (NLP) that prepares raw text for analysis by converting it into a form that machines can easily interpret. It begins with **tokenization**, where text is divided into smaller parts such as words or sentences. Then, **lowercasing** ensures consistency by treating words like "Apple" and "apple" as the same. Common and less meaningful words, known as **stop words** (e.g., "is," "the"), are removed to focus on important content. Processes like **stemming** and **lemmatization** simplify words to their root forms (e.g., "running" -> "run"), helping models understand related word variations. Finally, **text cleaning** eliminates unnecessary elements such as punctuation, symbols, and numbers, resulting in cleaner and more meaningful data for analysis.

ii. *Word embedding*: -

A word embedding is a way to represent words as vectors of numbers. Instead of treating words as separate, unique symbols, embeddings capture meaning and relationships between words mathematically.

Suppose the word BOOK and COPY are expressed in the form of vector as

$\overrightarrow{book}$ = [1,2,3] and $\overrightarrow{copy}$= [1,1,2]

Using **linear algebra**, their similarity can be measured through the dot product or cosine similarity:

$$\text{Cos}\theta = \frac{\overrightarrow{book} \cdot \overrightarrow{copy}}{|\overrightarrow{book}|\ |\overrightarrow{copy}|}$$

If there are too many words then AI represents them as word matrix as:

$$\begin{matrix} 1 & 2 & 4 \\ 2 & 0 & 0 \\ 3 & 5 & 1 \end{matrix}$$

Each row represents a word, while each column represents a linguistic feature. Such matrices are used by neural networks to identify patterns in text data.

Natural Language Generation (NLG):

Natural Language Generation (NLG) is a subfield of NLP that is responsible for generating natural language from a machine representation system. It takes structured or unstructured data as input and produces coherent, meaningful and contextually appropriate text. Simply, it generates human language from data, instructions or some input.

Word prediction (like autocomplete) is a part of NLG, especially in AI language models. NLG uses word prediction iteratively to generate full sentences and paragraphs.

Word Prediction

Suppose we've an incomplete sentence:

I am playing ………….

And let's say our AI knows only 3 possible next words- *study, football, paper*

Just like before every word is represented as numbers as:

S = [2,1]   F = [3,2]   P = [0,1]

The model also converts "I am playing" into a vector, let's say:

C = [2,1]

The model compares **C** with each word vector to see which fits best.

$$C \cdot S = 2 \times 2 + 1 \times 1 = 5$$
$$C \cdot F = 2 \times 3 + 1 \times 2 = 8$$
$$C \cdot P = 2 \times 0 + 1 \times 1 = 1$$

The model turns these dot products into probabilities.

$$P(\text{word}) = \frac{dot\ product}{\sum dot\ products}$$

Now each probability:

$$P(\text{study}) = \frac{5}{5+8+1} \approx 0.36$$

$$P(\text{football}) = \frac{8}{5+8+1} \approx 0.57$$

$$P(\text{paper}) = \frac{1}{5+8+1} \approx 0.07$$

Since the probability of F is the highest, AI predicts:

"I am playing **football**"

3.Learning/Optimization – How AI "improves itself and finds the best solution."

Learning and optimization are core pillars of artificial intelligence (AI), enabling systems to adapt, improve, and solve complex problems. Modern AI leverages a wide array of optimization techniques to train models, tune parameters, and enhance performance across diverse applications, from deep learning to edge computing. During training, model parameters are adjusted to minimize errors and improve predictive performance. Trained models can then generate output or make predictions on new, unseen data.

Mathematics is the backbone of learning and optimization in AI, providing the frameworks for data representation, model training, and efficient problem-solving. Calculus serves as the essential mathematical tool in optimization techniques used to train ML models.

Let us understand how AI learns and optimizes using "**Calculus"?**

When we teach AI to do tasks like recognizing handwriting or predicting prices, it needs to learn from its mistakes and improve. This is where calculus, especially derivatives, becomes important. Let's understand how AI uses basic calculus concepts to learn.

How Does AI Measure Its Mistakes?

Before AI can improve, it needs to know how wrong it is. This is done using something called a **Loss Function** (or Error Function).

Simple Example: Suppose AI is trying to predict house prices:

- Actual price: Rs. 50,00,000
- AI's prediction: Rs. 45,00,000
- Error: Rs. 5,00,000

For a single prediction, we can calculate error as:

Error = (Predicted value - Actual value)²

For multiple predictions, we take the average:

Loss Function: $L = (\mathbf{1/n}) \sum(\mathbf{Predicted} - \mathbf{Actual})^2$

Where n is the number of predictions.

The AI's Goal: Make this loss (error) as small as possible.

Understanding Gradient Descent with Simple Calculus

Gradient Descent is the main mathematical method AI uses to reduce errors in predictions. It comes from differential calculus, which helps us find how fast a function is changing.

Let's say AI is trying to learn a simple linear relationship:

y = mx + c

where:

• m = slope (what AI needs to learn)
• c = intercept (what AI needs to learn)
• x = input
• y = output (actual value)

Example Data

x	y (actual)
1	3
2	5
3	7

Step 1: Make an Initial Guess

Let's start with m = 0 and c = 0.
This means our model predicts: $y_{pred} = 0x + 0 = 0$

Step 2: Find the Prediction Errors

x	y(actual)	y(predicted)	Error = (predicted − actual)
1	3	0	-3
2	5	0	-5
3	7	0	-7

Step 3: Calculate Total Error (Loss)

We use Mean Squared Error (MSE):

$L = \left(\frac{1}{n}\right) \Sigma (y_{pred} - y_{actual})^2$

Substituting the values:

$L = (1/3)[(0-3)^2 + (0-5)^2 + (0-7)^2] = 27.67$

This is the loss function, and AI's goal is to minimize this error.

Step 4: Use Derivatives to Find the Direction of Improvement

To reduce L, AI needs to know:
• How does L change if m changes?
• How does L change if c changes?

That's exactly what derivatives tell us.

Using normal derivatives:

$dL/dm = (2/n)\ \Sigma (y_{pred} - y_{actual}) \times x$

$dL/dc = (2/n)\ \Sigma (y_{pred} - y_{actual})$

Substitute the values:

$dL/dm = (2/3)[(-3)(1) + (-5)(2) + (-7)(3)] = (2/3)(-34) = -22.67$

$dL/dc = (2/3)[(-3) + (-5) + (-7)] = (2/3)(-15) = -10$

Step 5: Update m and c

We adjust m and c in the opposite direction of the derivative (because the derivative shows the direction of increasing error).

The formula:
New value = Old value − (learning rate) × (derivative)

Let learning rate = 0.1

$m_{new} = 0 - 0.1(-22.67) = 2.27$

$c_{new} = 0 - 0.1(-10) = 1.0$

Step 6: Repeat the Process

Now with m = 2.27 and c = 1.0, our predictions become much closer to actual values.
If we keep repeating these steps, the derivatives (slopes) get smaller and smaller until:
$dL/dm = 0$ and $dL/dc = 0$
At that point, the error is minimum, and AI has learned the best values:
$m \approx 2$, $c \approx 1$
So the AI model becomes: $y = 2x + 1$

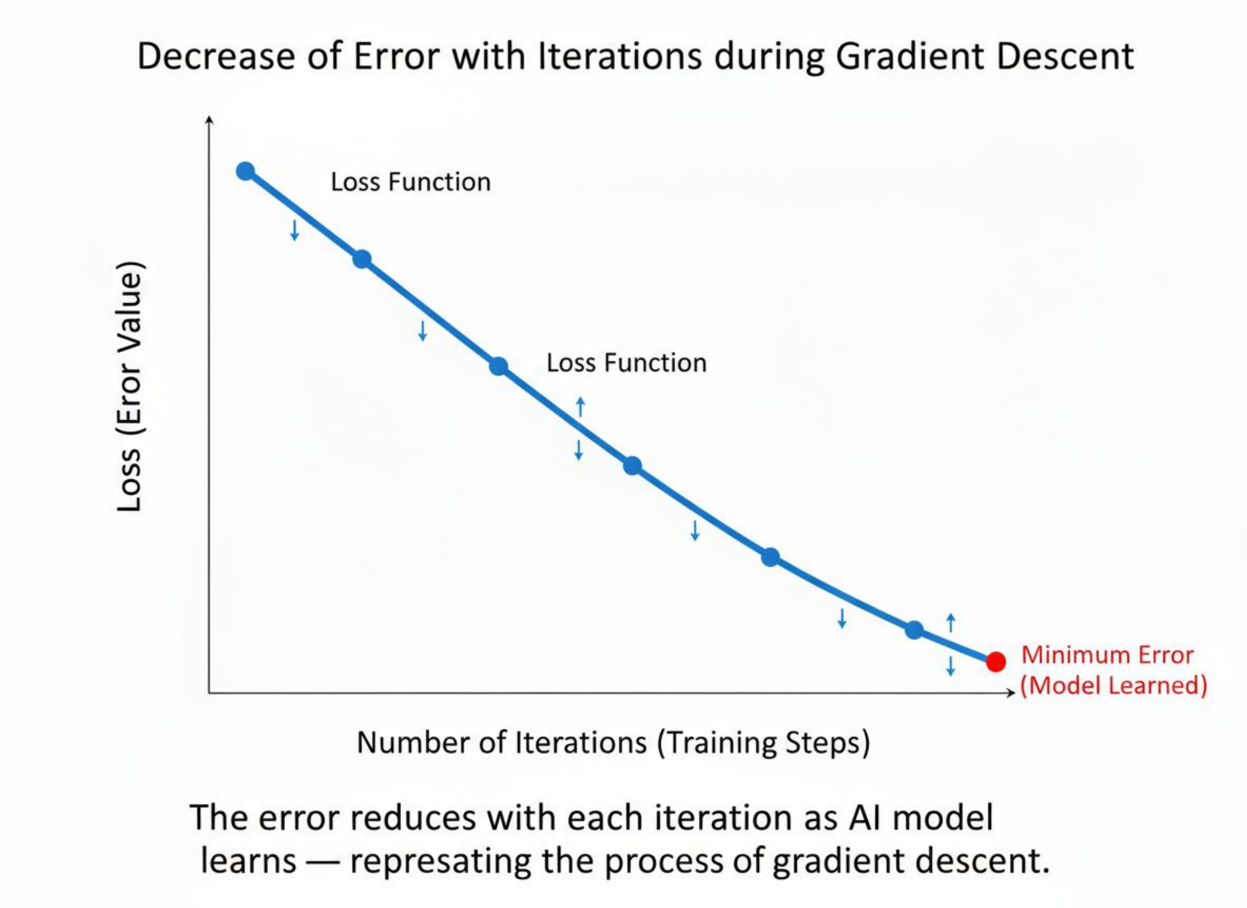
In Simple Words

Gradient Descent helps AI learn from its mistakes. Each time it checks how much it was wrong (error), it uses calculus (derivatives) to see which way to move and slowly adjusts m and c until the line fits the data perfectly.

• A derivative shows how fast something changes.
• When the derivative is positive, increasing the variable increases the error.
• When the derivative is negative, increasing the variable decreases the error.

Kuwait Journal of Engineering Research
ISSN: 2984-8016
Vol. 3 No. 1 (2025)

• So AI moves in the opposite direction of the derivative to reach the lowest point of the loss curve — this is the "descent" in gradient descent.



Practical Implications

The mathematical foundations discussed in the previous sections—logic, linear algebra, probability, and calculus—are not merely theoretical constructs but form the backbone of AI technologies that impact millions of lives daily. This section explores how these mathematical principles are applied across various industries and domains, demonstrating the practical significance of mathematics in modern AI systems.

- Face recognition uses vectors and optimization
- Recommendation systems use matrix operations
- Self-driving cars use calculus-based decision making

1. **Facial Recognition Systems**

Facial recognition technology has become omnipresent in modern security systems, smartphones, and social media platforms. When you unlock your phone using Face ID or when Facebook suggests tagging friends in photos, mathematical operations are working behind the scenes.

Mathematical Foundation - Vectors and Optimization:

Face recognition uses vectors and optimization in the following ways:

Vector Representation:

- Each face is converted into a high-dimensional vector (typically 128 or 512 dimensions) representing unique facial features such as the distance between eyes, nose shape, jawline contour, and cheekbone structure.

- The system stores these feature vectors in a database.

- For example, a face might be represented as: Face_vector = [2.3, 4.1, 1.8, ..., 3.5]

Similarity Calculation:

- When a new face appears, it is converted into a vector and compared with stored vectors using cosine similarity (as explained in the Knowledge Representation section).

- The cosine similarity formula: $\cos\theta = (\vec{A} \cdot \vec{B}) / (|\vec{A}|\,|\vec{B}|)$

- If the similarity score exceeds a threshold (typically > 0.9), the system identifies a match.

Optimization:

- The neural network that generates these face vectors is trained using gradient descent optimization.

- The optimization process adjusts millions of parameters to minimize the error in face recognition.

- The loss function being minimized ensures that:

  * Vectors of the same person are very close (small angle between vectors)

  * Vectors of different people are far apart (large angle between vectors)

<u>Real-World Example</u>:

Apple's Face ID uses over 30,000 infrared dots to map facial features, creating a mathematical model stored as vectors. When you try to unlock your phone:

1. Your face is captured and converted to a vector

2. This vector is compared with the stored vector using cosine similarity

3. If similarity > 0.9, the phone unlocks

4. The system uses optimization to continuously improve accuracy

The optimization algorithms help Face ID adapt to changes in your appearance (glasses, beard, aging) by updating the stored vectors.

2. **Self-Driving Cars**

Autonomous vehicles represent one of the most complex applications of AI, integrating computer vision, prediction, and control systems. Self-driving cars use calculus-based decision making to navigate safely.

Mathematical Foundation - Calculus for Decision Making

Self-driving cars use calculus in multiple critical ways:

Trajectory Prediction Using Derivatives:

- Position of objects is tracked over time

- First derivative (velocity) = rate of change of position: $v = dx/dt$

- Second derivative (acceleration) = rate of change of velocity: $a = dv/dt = d^2x/dt^2$

- These derivatives predict where objects will be in the future

Example: If a pedestrian is at a position $x = 10$m with velocity $v = 2$ m/s, after 3 seconds they will be at:

$x_future = x + v \times t = 10 + 2 \times 3 = 16$m

Path Optimization:

- The car must choose the safest and most efficient path

- This involves minimizing a cost function: $Cost = w_1(collision\_risk) + w_2(time) + w_3(comfort) + w_4(fuel)$

- Calculus (derivatives) finds the minimum of this cost function

- The optimal path has dCost/dpath = 0

Real-World Example:

A Tesla Autopilot system detects a car ahead:

1. Computer vision (using matrix operations) identifies the car

2. Calculus calculates the car's velocity using derivatives: $v = \Delta x/\Delta t$

3. Calculus predicts future position: $x\_future = x_0 + vt + ½at^2$

4. Optimization determines the safest braking profile by minimizing:

   Cost = (risk of collision) + (passenger discomfort)

5. Calculus-based control adjusts brake pressure continuously using derivatives

If the car ahead is 50m away, moving at 20 m/s, and your car is at 30 m/s:

- Relative velocity = 30 - 20 = 10 m/s (closing speed)

- Time to collision = distance/relative velocity = 50/10 = 5 seconds

- Calculus determines optimal deceleration to avoid collision while maintaining comfort

3. **Netflix - Movie Recommendations**

Netflix uses matrix operations to recommend movies and TV shows, accounting for over 80% of content watched on the platform.

Real-World Example:

Netflix User-Item Matrix (simplified):

$$\begin{array}{l|cccc|} & Stranger\ Things & Breaking\ Bad & The\ Crown & Squid\ Game \\ Alice & [\ 5 & 4 & 2 & 0\ ] \\ Bob & [\ 4 & 5 & 1 & 0\ ] \\ Charlie & [\ 2 & 1 & 5 & 4\ ] \\ Diana & [\ 0 & 0 & 4 & 5\ ] \end{array}$$

Matrix Factorization Process:

1. Decompose into User-Feature and Feature-Movie matrices

2. Features might represent: [Action, Drama, International, Teen-appeal]

User-Feature Matrix U:

Alice: [0.8, 0.6, 0.2, 0.9] ← Likes action and teen content

Bob: [0.7, 0.9, 0.1, 0.5] ← Likes drama

Charlie: [0.2, 0.3, 0.9, 0.4] ← Likes international content

Diana: [0.3, 0.2, 0.95, 0.3] ← Strongly prefers international

Feature-Movie Matrix V:

Kuwait Journal of Engineering Research
ISSN: 2984-8016
Vol. 3 No. 1 (2025)
Stranger Things: [0.7, 0.4, 0.1, 0.9] ← Action, teen-appeal

Breaking Bad: [0.6, 0.95, 0.2, 0.3] ← Heavy drama

The Crown: [0.2, 0.6, 0.8, 0.1] ← International, drama

Squid Game: [0.5, 0.5, 0.9, 0.4] ← International, action

Prediction for Alice and Squid Game:

Alice hasn't watched Squid Game (rating = 0 in matrix)

Prediction = U[Alice] × V[Squid Game]^T

= [0.8, 0.6, 0.2, 0.9] · [0.5, 0.5, 0.9, 0.4]

= (0.8×0.5) + (0.6×0.5) + (0.2×0.9) + (0.9×0.4)

= 0.4 + 0.3 + 0.18 + 0.36 = 1.24

Normalized to 5-star scale: ≈ 4.2 stars

Netflix recommends Squid Game to Alice with high confidence.

Matrix Operations for Finding Similar Users:

To find users similar to Alice, calculate cosine similarity using matrix operations:

- Multiply Alice's vector with all other user vectors

- Divide by vector magnitudes

- Bob has highest similarity → recommend what Bob watched

Results And Discussion

This research explored how mathematics forms the foundation of AI through three key areas: Knowledge Representation, Natural Language Processing, and Learning/Optimization.

Key Findings

1. Knowledge Representation- AI uses three mathematical methods to store and process information:

- **Logic:** Uses symbols (∀, →) for reasoning. Example: AI deduced "HCl has pH < 7" from rules and facts.
- **Vectors:** Represents concepts as numbers. The angle between "Acid" and "HCl" vectors was only 3°, showing high similarity.
- **Matrices:** Stores relationships efficiently. Matrix multiplication revealed indirect connections.

2. Natural Language Processing- Mathematics enables AI to understand and generate language:

- **Word Embeddings:** Words represented as vectors allow mathematical comparison
- **Word Prediction:** AI predicted "I am playing football" by calculating dot products and probabilities (football: 57%, study: 36%, paper: 7%)
- These mathematical operations power chatbots and voice assistants

3. Learning and Optimization- Calculus is the core of AI learning:

- **Gradient Descent:** Using derivatives, AI learned y = 2x + 1 from data by reducing error from 27.67 to near zero
- **Real Application:** Handwritten digit recognition improved from 10% to 98% accuracy through calculus-based training

Integration of Mathematics

These mathematical areas work together:

- Vectors store words (Linear Algebra)
- Matrices organize relationships (Linear Algebra)
- Derivatives optimize performance (Calculus)
- Logic ensures consistency (Logic)

For example, in ChatGPT: word embeddings represent language, neural networks process it, and gradient descent continuously improves responses.

Conclusion

This research paper explored the fundamental role of mathematics in Artificial Intelligence by examining Knowledge Representation, Natural Language Processing, and Learning/Optimization.

Mathematics is the Foundation of AI Every aspect of AI relies on mathematical principles:

- Logic enables reasoning
- Linear algebra provides data representation through vectors and matrices
- Calculus enables learning through derivatives and optimization
- Without these foundations, modern AI cannot exist

Integration Creates Intelligence AI's power comes from combining mathematical disciplines. When you use ChatGPT or Google Assistant, you're experiencing the result of logic, linear algebra, and calculus working together seamlessly.

From Theory to Practice The mathematical concepts in this paper directly power real applications:

- Face unlock: comparing facial feature vectors
- Voice assistants: processing words as mathematical embeddings
- ChatGPT: Neural networks trained through gradient descent
- Netflix recommendations: matrix operations finding patterns

Answering the Research Question

"What is the use of mathematics in AI?"

Mathematics serves three essential roles:

1. **Representation:** Encoding information (logic, vectors, matrices)
2. **Processing:** Computing and reasoning (algebraic operations)
3. **Learning:** Improving through experience (calculus-based optimization)

In conclusion: Mathematics is not just used in AI—Mathematics is the foundation of AI. The two are inseparable, and this relationship is key to understanding the technology shaping our future.

References

[1] GeeksforGeeks. (n.d.). *Knowledge representation in AI*. https://www.geeksforgeeks.org/artificial-intelligence/knowledge-representation-in-ai/

[2] IBM. (n.d.). *What is natural language processing (NLP)?* https://www.ibm.com/think/topics/natural-language-processing

[3] McCarthy, J. (n.d.). *What is artificial intelligence?* Stanford University. http://jmc.stanford.edu/artificial-intelligence/what-is-ai/

[4] Rajat. (2024, November 23). *Mathematical optimization in AI: Enhancing algorithms for better performance*. Medium. https://medium.com/@rajat01221/mathematical-optimization-in-ai-enhancing-algorithms-for-better-performance-536551c83ad9

[5] ResearchGate. (2024). *The role of mathematics in the development of artificial intelligence algorithms*. https://www.researchgate.net/publication/394236202_The_Role_of_Mathematics_in_the_Development_of_Artificial_Intelligence_Algoarithms

[6] Singh, K. (2023). The role of mathematics in artificial intelligence and machine learning. *International Journal for Research Publication and Seminar, 14*(5). https://doi.org/10.36676/jrps.v14.i5.1434

[7] Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). *Learning representations by back-propagating errors*. Nature, 323, 533–536. https://www.cs.toronto.edu/~hinton/absps/naturebp.pdf

[8] Bottou, L. (2010). *Large-scale machine learning with stochastic gradient descent*. https://leon.bottou.org/papers/bottou-2010

[9] Salton, G., Wong, A., & Yang, C. S. (1975). *A vector space model for automatic indexing*. Communications of the ACM, 18(11), 613–620. https://doi.org/10.1145/361219.361220

[10] Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press. https://www.deeplearningbook.org

[11] Kingma, D. P., & Ba, J. (2015). *Adam: A Method for Stochastic Optimization*. https://arxiv.org/abs/1412.6980

[12] Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). *Efficient estimation of word representations in vector space*.
https://arxiv.org/abs/1301.3781

[13] Pennington, J., Socher, R., & Manning, C. D. (2014). *GloVe: Global Vectors for Word Representation*. https://nlp.stanford.edu/pubs/glove.pdf

[14] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). *Attention is all you need*. https://arxiv.org/abs/1706.03762

[15] Sahani, S.K. and Sah, D.K. (2023). Hybrid Analytical-Numerical Techniques for Machine Learning Optimization: Integrating Laplace Transform and Runge-Kutta Fourth-Order Methods, Review of Contemporary Philosophy, 22, 1, 6854-6860

[16] Sahani, S.K., et al.(2023), Solutions that Approach the Proposal Generating Difficulty in Parallel Marketplaces for Delivering Material Concessions: A Case Study of Transportation Problem Letters in High Energy Physics, Vol. 2023,315-330.

[17] Sahani, S.K. et al., (2022).A Comprehensive Study on Predicting Numerical Integration Errors using Machine Learning Approaches, Letters in High Energy Physics, 96-103.

[18] Sahani, S.K. (2021). Differential Equation on Astrophysics: A Fundamental Approach to Understanding Cosmic Structures and Their Dynamic Evolution, Letters in High Energy Physics, Vol. 2021, 1-13.

[19] Sahani, S.K. (2024). Big Data Learning Analytics & Optimization: Algorithms, Challenges, and Applications, Analysis and Metaphysics, Vol23 (1), 2024pp. 912–924

[20] Sahani, S.K. (2024). AI-Enhanced Finite Element Method (FEM) for Structural Analysis, J. Electrical Systems 20, 1, 661-676

[21] Sahani, S.K. (2022). A Mathematical Framework for Incorporating Neural Networks into Root-Finding Algorithms, J. Electrical Systems 18, 1, 99-109.

[22] Sahani, S.K. (2023). Application of Numerical Methods in Structural Health Monitoring Using IoT Sensors, J. Electrical Systems 19-1(2023): 194-207.
[23] S.K Sahani et al.(2023). Constructing a Precise Method to Control Non-Linear Systems Employing Special Functions and Machine Learning, Communications on Applied Nonlinear Analysis, Vol.30, No.2, 1-14, 2023.
[24] S. K. Sahani, et al., A case study on analytic geometry and its applications in real life, international Journal of Mechanical Engineering, Vol.4, no.1, June 2019, 151-163,
[25] Sahani, S.K. (2021). Business Demand Forecasting Using Numerical Interpolation and Curve Fitting. The International Journal of Multiphysics, 15(4), 482 - 492.
[26] Sahani, S.K. 2024. Sah, B.K. (2024). Integrating Neural Networks with Numerical Methods for Solving Nonlinear Differential Equations, Computer Fraud and Security, Vol.2024, Issue 1, 25-37
[27] Sahani, S.K., et al. (2025). Case Study on Mechanical and Operational Behavior in Steel Production: Performance and Process Behavior in Steel Manufacturing Plant, Reports in Mechanical Engineering, 6, 1,180-197.