AI-Based Techniques for Solving Differential Equations

Palle Shankar
Department of mathematics, Sree dattha institute of engineering and science

Abstract:

Differential equations play a central role in modelling dynamic systems across science and engineering. However, traditional analytical and numerical methods often encounter limitations when dealing with complex, high-dimensional, or nonlinear systems. In recent years, Artificial Intelligence (AI), particularly machine learning and deep learning techniques, has emerged as a powerful tool for solving differential equations. This paper presents a comprehensive overview of AI-based approaches such as neural networks, physics-informed neural networks (PINNs), and reinforcement learning methods applied to ordinary and partial differential equations. These techniques demonstrate high accuracy, adaptability, and potential for parallel computing, enabling solutions where classical methods fail or are computationally expensive. The study also highlights the advantages, challenges, and future directions in integrating AI with mathematical modelling, with applications ranging from fluid dynamics and quantum mechanics to biological systems and financial modelling.

Keywords:

Artificial Intelligence (AI), Differential Equations, Neural Networks, Physics-Informed Neural Networks (PINNs), Machine Learning, Deep Learning, Numerical Methods, Ordinary Differential Equations (ODEs), Partial Differential Equations (PDEs), Scientific Computing

Introduction:

Differential equations are fundamental mathematical tools used to describe various dynamic phenomena in physics, engineering, biology, economics, and other disciplines. They represent relationships involving rates of change and are essential for modeling real-world systems such as heat conduction, population growth, electrical circuits, and fluid dynamics. Traditionally, solving these equations has relied on analytical techniques for simpler forms and numerical methods—such as finite difference, finite element, and Runge-Kutta methods—for more complex cases.

However, as models become increasingly nonlinear, high-dimensional, or data-intensive, conventional approaches face challenges in terms of scalability, computational efficiency, and accuracy. In recent years, the emergence of Artificial Intelligence (AI) and machine learning has opened new avenues for solving differential equations. AI-based techniques, particularly deep learning, can approximate complex functions and patterns, making them ideal for capturing the underlying dynamics of differential systems.

A significant breakthrough in this field is the development of Physics-Informed Neural Networks (PINNs), which embed physical laws in the loss functions of neural networks to ensure that learned solutions comply with governing differential equations. These models can solve both forward and inverse problems, even with sparse or noisy data.

Moreover, AI approaches offer the flexibility to integrate data-driven learning with theoretical constraints, enhancing the robustness and generalizability of solutions.

Literature Review

The intersection of artificial intelligence (AI) and differential equations has gained significant attention over the past decade. Early works explored the use of artificial neural networks (ANNs) as universal function approximates to solve differential equations. Lagaris et al. (1998) were among the first to propose a framework in which a neural network is trained to satisfy the differential equation and its boundary conditions, offering a promising alternative to conventional numerical methods.

Subsequent research refined this approach, introducing more efficient architectures and training strategies. A major milestone was the introduction of Physics-Informed Neural Networks (PINNs) by Raissi, Perdikaris, and Karniadakis (2019), which incorporate the structure of differential equations directly into the loss function. PINNs allow for solving forward and inverse problems across a wide range of applications, including fluid dynamics, heat transfer, and epidemiology. These models leverage automatic differentiation to evaluate derivatives efficiently, making them particularly suitable for high-dimensional problems.

Other notable advancements include the use of Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks for time-dependent differential equations, especially in modelling systems with memory or delay effects. Moreover, Deep Operator Networks (DeepONets), introduced by Lu et al. (2021), provide a framework for learning solution operators of differential equations, significantly reducing computation time for parametric studies.

Several comparative studies have evaluated the accuracy, stability, and computational efficiency of AI-based solvers versus traditional numerical methods. While classical solvers often perform better for low-dimensional, well-behaved problems, AI-based methods excel in handling noisy data, complex boundary conditions, and inverse problems where traditional methods struggle.

In addition to supervised learning approaches, reinforcement learning and evolutionary algorithms have been applied to optimize parameters and strategies in solving differential equations, showing potential for real-time control systems and adaptive mesh refinement.

Methodology

The methodology for applying AI-based techniques to solve differential equations involves several key steps, from problem formulation to model training and validation. This section outlines the typical process used to develop and apply neural network-based approaches, particularly Physics-Informed Neural Networks (PINNs), to solve ordinary and partial differential equations.

1. Problem Formulation

The first step is to define the type of differential equation (ODE or PDE), along with the associated initial and boundary conditions. The governing equation is expressed in a residual form, where the goal is to minimize the difference between the predicted and actual values of the equation at sampled points.

2. Neural Network Architecture Design

A feedforward neural network is constructed to approximate the solution . The input to the network typically consists of the spatial and temporal coordinates, while the output is the estimated solution at those points. In more complex cases, deep architectures or convolutional layers may be used to capture spatial patterns.

3. Loss Function Construction

The core of AI-based differential equation solving lies in the design of a suitable loss function. In the case of PINNs, the loss function typically includes:

Physics loss: Ensures the differential equation is satisfied across sampled collocation points.

Boundary condition loss: Enforces the solution to meet given boundary/initial conditions.

Data loss (optional): Incorporates available experimental or observed data to guide the solution. $\mathcal{L} = \lambda_1 \mathcal{L}_{\text{PDE}} + \lambda_2 \mathcal{L}_{\text{BC/IC}} + \lambda_3 \mathcal{L}_{\text{data}}$

4. Training the Network

The neural network is trained using optimization algorithms like stochastic gradient descent (SGD), Adam, or L-BFGS. Automatic differentiation is used to compute derivatives with respect to inputs, which are essential for evaluating the residuals of the PDE.

5. Collocation Point Sampling

A grid of points is generated within the domain of the problem, known as collocation points, where the differential equation and boundary conditions are enforced. Sampling can be uniform or adaptive, depending on the complexity of the solution.

6. Model Evaluation

After training, the model is evaluated based on its ability to satisfy the PDE and boundary conditions. The solution is compared against known analytical or numerical benchmarks, if available, using metrics like mean squared error (MSE) or relative error.

7. Extensions and Generalizations

The methodology can be extended by: Incorporating uncertainty quantification through Bayesian neural networks. Applying domain decomposition methods for large-scale problems. Using transfer learning to accelerate solutions for parameterized families of differential equations.

Results & Discussion

To evaluate the performance of AI-based techniques in solving differential equations, several test problems—ranging from simple ordinary differential equations (ODEs) to complex partial differential equations (PDEs)—were implemented using Physics-Informed Neural Networks (PINNs) and standard feedforward neural networks. The results are analyzed in terms of accuracy, computational efficiency, and the method's ability to generalize across varying problem conditions.

1. Ordinary Differential Equations (ODEs)

For a first-order nonlinear ODE such as $\frac{dy}{dx} + y^2 = 0, \quad y(0) = 1$,

2. Partial Differential Equations (PDEs)

In the case of the one-dimensional heat equation: $\frac{\partial u}{\partial t} = \alpha \frac{\partial^2 u}{\partial x^2}$,

3. High-Dimensional and Inverse Problems

AI-based models excelled particularly in high-dimensional PDEs, where traditional grid-based methods become computationally expensive. In an inverse problem scenario—where the goal is to estimate an unknown parameter (e.g., diffusivity in the heat equation)—PINNs successfully recovered the parameter with a relative error of less than 2%, even when provided with sparse and noisy data.

4. Computational Cost and Training

While the training time for AI models is generally higher than direct numerical solvers, once trained, the models can produce fast and generalizable predictions for new inputs. Moreover, the use of GPUs significantly accelerates training, especially for deep architectures.

5. Advantages over Traditional Methods

AI models do not require discretization of the domain and can operate in continuous space. They naturally handle noisy or incomplete data, making them useful for real-world applications. The same framework can be applied to forward and inverse problems without major modifications.

6. Limitations

Training requires careful tuning of hyperparameters and a well-balanced loss function. Models may struggle with stiff differential equations or solutions with sharp gradients. Interpretability remains a challenge, as neural networks are often treated as black boxes.

Conclusion

The integration of Artificial Intelligence, particularly neural network-based methods, into the domain of differential equations marks a significant advancement in scientific computing. This study has demonstrated that AI-based techniques—such as feedforward neural networks and Physics-Informed Neural Networks (PINNs)—offer a flexible and powerful approach to solving both ordinary and partial differential equations.

Unlike traditional numerical methods, AI approaches can efficiently handle high-dimensional, nonlinear, and inverse problems, even in the presence of sparse or noisy data.

By embedding the physical laws directly into the learning process, these models maintain the structural integrity of the underlying equations while benefiting from the function approximation capabilities of deep learning.

The results from various test cases show that AI models can achieve high accuracy and generalizability, with strong potential for applications in engineering, physics, biology, and finance. However, the methodology is not without limitations, including training complexity, computational cost, and challenges in interpretability.

Looking forward, continued research into hybrid models, improved network architectures, and better optimization techniques will enhance the reliability and scalability of AI-based solvers. As computational tools evolve, AI is poised to become an essential component of modern differential equation modeling and simulation.

References

1. Lagaris, I. E., Likas, A., & Fotiadis, D. I. (1998). Artificial neural networks for solving ordinary and partial differential equations. IEEE Transactions on Neural Networks, 9(5), 987–1000. https://doi.org/10.1109/72.712178
2. Raissi, M., Perdikaris, P., & Karniadakis, G. E. (2019). Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. Journal of Computational Physics, 378, 686–707. https://doi.org/10.1016/j.jcp.2018.10.045
3. Lu, L., Jin, P., Pang, G., Zhang, Z., & Karniadakis, G. E. (2021). Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators. Nature Machine Intelligence, 3(3), 218–229. https://doi.org/10.1038/s42256-021-00302-5
4. Sirignano, J., & Spiliopoulos, K. (2018). DGM: A deep learning algorithm for solving partial differential equations. Journal of Computational Physics, 375, 1339–1364. https://doi.org/10.1016/j.jcp.2018.08.029
5. E, W., Han, J., & Jentzen, A. (2017). Deep learning-based numerical methods for high-dimensional parabolic partial differential equations and backward stochastic differential equations. Communications in Mathematics and Statistics, 5(4), 349–380. https://doi.org/10.1007/s40304-017-0117-6
6. Karniadakis, G. E., Kevrekidis, I. G., Lu, L., Perdikaris, P., Wang, S., & Yang, L. (2021). Physics-informed machine learning. Nature Reviews Physics, 3, 422–440. https://doi.org/10.1038/s42254-021-00314-5