WEP Warwick Evans Publishing

Transactions on Computer Science and Intelligent Systems Research
ISSN: 2960-1800, eISSN: 2960-2238 | Volume 12
ESAC 2026

Research on Numerical Solution Optimization of Partial Differential Equations Based on Physics-Informed Neural Network (PINN)

Qixuan Li *

School of Mathematics and Statistics, Northeast Normal University, Changchun, China, 130024

* Corresponding Author Email: liqixuan908@nenu.edu.cn

Abstract. Partial differential equations (PDEs) are core tools for characterizing physical laws and are widely used in fluid mechanics, power systems, additive manufacturing, and other fields. However, traditional numerical methods are limited by mesh partitioning, leading to a trade-off between accuracy and efficiency in complex geometric domains and high-dimensional problems. While Physically Informed Neural Networks (PINNs) achieve meshless PDE solutions by embedding prior physical knowledge, they still face challenges such as insufficient accuracy, low training efficiency, and poor stability. This study aims to address the core bottlenecks of PINN in solving PDEs by proposing a systematic optimization strategy to improve its numerical solution accuracy, efficiency, and stability. Methodologically, firstly, a composite loss function containing PDE residuals and boundary initial condition constraints is constructed based on automatic differentiation techniques to ensure that the network satisfies physical laws. Secondly, the PINN training system is optimized, including using Neural Architecture Search (NAS-PINN) to automatically match the optimal network structure, designing an adaptive data sampling strategy to improve data utilization, and introducing parallel computing and hardware acceleration techniques to reduce training time. Finally, the effectiveness of the strategy is verified through classical PDEs and engineering problems. The results show that the optimized PINN achieves high-precision numerical approximation in classical PDE solutions and can efficiently solve forward design and backward parameter inversion problems in engineering scenarios. Compared with traditional methods, it exhibits stronger adaptability in complex geometric domains and high-dimensional problems, while improving training efficiency by more than 30% and significantly enhancing stability. This research not only enriches the optimization theory of PINN and provides an efficient new path for PDE numerical solutions, but also promotes the practical application of PINN in engineering fields such as fluid mechanics, additive manufacturing, and power systems, possessing significant theoretical value and application significance.

Keywords: Partial Differential Equations; Physically Informed Neural Networks; Numerical Solution; Deep Learning; Optimization Strategies.

1. Introduction

1.1. Research Background and Significance

Partial differential equations, as an important branch of applied and pure mathematics, play an irreplaceable role in the development of modern science and technology [1] . These equations can accurately describe the fundamental laws in physical science, among which the Navier-Stokes equations reflect the fundamental mechanical laws of viscous fluid flow [2] . They play a crucial role in applications such as precision guidance, satellite positioning, and radar detection in the military field; sonar detection, oil exploration, and weather forecasting in industrial engineering; and X-ray films and CT scanners in the medical field.

With the rapid development of computer technology, the application scope of partial differential equations in physics, meteorology, astronomy, oceanography, aerospace and other disciplines and engineering technologies is constantly expanding. Traditional numerical solution methods include finite difference, finite volume, finite element and other techniques, which have been widely used in computational fluid dynamics, computational physics, computational chemistry, computational biology, computational materials science, image processing and other fields [3] .

Physics-Informed Neural Networks (PINNs), as a novel hybrid-driven neural network model, incorporates prior physical knowledge, expressed in the form of partial differential equations, into the network training process, enabling the network model to automatically satisfy predefined physical constraints [4] . This method leverages the superior function fitting capabilities of deep neural networks, guiding the iterative update process of adjustable parameters by optimizing the least-squares fitting function and initial-boundary conditions corresponding to the equations [5] . Compared to traditional numerical methods, PINNs, as a meshless numerical solution method for partial differential equations, exhibit unique advantages in handling complex geometric boundaries and high-dimensional problems [6] . Researching optimization strategies for PINNs has significant theoretical and practical value for improving the accuracy and efficiency of numerical solutions to partial differential equations.

1.2. Current Status of Domestic and International Research

As an emerging hybrid-driven neural network model, the Physics-Informed Neural Network (PINN) has attracted widespread attention globally since its initial proposal by Professor Raissi in 2018 [7] . PINN integrates prior physical knowledge, existing in the form of partial differential equations, into the network training process, enabling the network model to automatically satisfy pre-defined physical constraints. This innovative method opens up a new research path for the numerical solution of partial differential equations.

At the international research level, PINN has been successfully applied in multiple disciplines, with researchers in fluid mechanics, biomedicine, and materials science actively exploring its application potential [8] . Willard et al. systematically summarized nine main objectives of PINN from an application scenario perspective, including extrapolation prediction, parameterization, order reduction models, downscaling, uncertainty quantification, inverse modeling, equation discovery, solving differential equations, and data generation [9] . PINN accurately and efficiently calculates derivatives of various orders through automatic differentiation techniques, avoiding the complexity of time and space discretization in traditional numerical methods [10-11] .

Domestic research institutions and scholars have also made significant progress in the field of PINN. Researchers have applied PINN to specific engineering problems such as optimization of laser powder bed melting process parameters, transient stability assessment of power systems, and time-varying reliability analysis [12] . Domestic scholars have paid particular attention to optimization strategies for PINN in solving practical engineering problems, such as progressive training strategies based on curriculum regularization, and improving model performance through a combination of data-driven and physics-driven approaches [13] .

Although PINN has been widely used in solving partial differential equations in various fields, some key challenges remain, including insufficient accuracy on specific problems, low training efficiency, and unstable training process [14] . These challenges provide clear directions for future research, prompting scholars to continuously explore more efficient and stable PINN optimization methods.

2. Theoretical Basis of Physics-Informed Neural Network (PINN)

2.1. Basic Concepts of PINN

The Physics-informed neural network (PINN) is a novel hybrid-driven neural network model that integrates prior physical knowledge, existing in the form of PDEs, into the network training process, enabling the network model to automatically satisfy pre-defined physical constraints. PINN, based on deep neural networks (DNNs), achieves function approximation and can solve forward problems (design-guided manufacturing) and backward problems (design derived from requirements) in additive manufacturing [11] .

PINN, as a machine learning method for solving partial differential equations, can consider initial values and boundary conditions. This method was initially proposed by Lagaris et al. in 1998, who used artificial neural networks to solve ordinary differential equations. With the development of automatic differentiation technology and deep learning software, Raissi et al. further refined PINN for forward solving and inverse discovery of nonlinear partial differential equations [15] . PINN can be used to handle complex nonlinear partial differential equations that are difficult to solve using traditional numerical methods, or inverse problems lacking sufficient observation data.

Within the PINN framework, physical information is generally represented by the initial and boundary conditions of partial differential equations, as well as certain special forms of these equations. Constraints from physical laws can compensate for errors caused by insufficient data, achieving good prediction accuracy without the need for extensive data training. The core advantage of PINN lies in its ability to solve partial differential equations without additional data, and the solution process does not require mesh partitioning or equation discretization, making it highly adaptable to high-dimensional problems and irregular computational domains.

2.2. Mathematical Principles of PINN

The mathematical principles of physics informed neural networks are based on the theoretical foundation of combining automatic differentiation techniques with physical constraints. PINN innovatively applies automatic differentiation technology to solve the derivative of the target parameter with respect to the input quantity, thereby bypassing traditional differentiation methods based on symbolic operations or numerical differences, accurately and efficiently obtaining the required derivatives of each order, making the evaluation of differential equations on the output side simple and direct. This method reduces the amount of data required for neural networks to solve partial differential equations by incorporating physical information, and emphasizes the importance of combining deep learning with mathematical physics knowledge.

PINN constrains the neural network model through existing physical laws and prior knowledge, guiding the model's training process to meet predetermined physical laws, thereby improving the model's interpretability and generalization ability. Considering the general form of nonlinear partial differential equations, its mathematical expression is:

$$\mathrm{F}[u(x,t)]+\mathrm{N}[u(x,t)]=f(x,t), \quad (x,t)\in\Omega \tag{1}$$

$$u(x,t)=g(x,t), \quad (x,t)\in\partial\Omega \tag{2}$$

$$u(x,0)=\mathrm{h}(x), \quad x\in\Omega \tag{3}$$

Where,$u(x,t)$is the unknown analytical solution of the equation,$\Omega$is the feasible region,$\partial\Omega$Represents the feasible region boundary. PINN treats the solution to the equation as a neural network composed of trainable parameters, eliminating the need for mesh partitioning of the feasible region during training.

The core of PINN is to guide the architecture design, optimization, and model training of deep neural networks through physical laws or scientific knowledge, such as feature data processing, loss function construction, parameter weight design, etc. The loss function typically consists of three main components: partial differential equation residual term, boundary condition constraint term, and initial condition constraint term (as shown in Table 1). This design enables the network to satisfy both equation constraints and boundary constraints during training, thereby obtaining physically reasonable solutions.

Table 1. Composition and mathematical expression of PINN loss function

Composition of the loss function	Mathematical expression	Physical meaning
PDE residual term	$\mathrm{L}_{PDE}=//\,\mathrm{F}[u_\theta]-f\,//^2$	Satisfies partial differential equation
Boundary condition term	$\mathrm{L}_{BC}=//\,u_\theta-g\,//^2$	Satisfies boundary constraints
Initial condition term	$\mathrm{L}_{IC}=//\,u_\theta(x,0)-\mathrm{h}\,//^2$	Satisfies initial state

Based on the automatic differentiation mechanism of neural networks, PINN can be naturally embedded into physical models represented by differential equations, enabling the characterization of complex nonlinear physical processes. This mathematical framework makes PINN perform well in dealing with complex nonlinear partial differential equations that are difficult to solve by traditional numerical methods.

3. Numerical Solution of Partial Differential Equations

3.1. Classification and Characteristics of Partial Differential Equations

Partial differential equations, as the core tool in the field of mathematical physics, play an important role in describing various physical phenomena in nature. Based on the order of the highest-order derivative, partial differential equations can be classified into first-order, second-order, and higher-order equations. Second-order partial differential equations are the most widely used in physics, and their standard form is:

$$A\frac{\partial^2 u}{\partial x^2}+B\frac{\partial^2 u}{\partial x\partial y}+C\frac{\partial^2 u}{\partial y^2}+D\frac{\partial u}{\partial x}+E\frac{\partial u}{\partial y}+Fu=G \tag{4}$$

where A, B, C, D, E, F is a function of the independent variable x, y, and G represents the non-homogeneous term. Based on the value of the discriminant $\Delta=B^2-4AC$, second-order partial differential equations can be further divided into three types. When $\Delta<0$, it is an elliptic equation, typically represented by the Laplace equation and the Poisson equation, often used to describe equilibrium problems such as steady-state heat conduction and electrostatic field distribution. When $\Delta=0$, it is a parabolic equation, such as the heat conduction equation and the diffusion equation, mainly describing the diffusion process evolving over time. When $\Delta>0$, it is a hyperbolic equation, with the wave equation being a typical example, used to describe wave propagation phenomena.

Physical Information Neural Networks (PINNs) exhibit unique advantages in processing these different types of partial differential equations (PDEs). PINN incorporates the physical information represented by the PDEs into the construction of the loss function, employing a combination of data-driven and physics-driven approaches to train a neural network model capable of automatically satisfying the solutions to PDEs. Through automatic differentiation techniques, PINN can quickly and efficiently compute PDE terms and design the loss function according to the form of the PDEs, embedding the physical information inherent in the PDEs into the neural network. This method can not only solve forward problems to obtain approximate solutions to PDEs but also handle backward problems to calculate unknown parameters in the equations.

3.2. Overview of Numerical Solution Methods

The development of numerical solution methods for partial differential equations has witnessed significant progress in the field of computational mathematics. Traditional numerical solution methods mainly include the finite difference method, the finite element method, the finite volume method, and the spectral method. These methods exhibit their own advantages and limitations in different application scenarios, laying a solid theoretical foundation for modern computational science.

The finite difference method, as one of the earliest developed numerical methods, uses the difference quotient at discrete points to approximate the derivative as its research tool. It has advantages such as simple form, ease of programming implementation, and high computational accuracy. The mathematical theory of this method is relatively mature, and common stability analysis methods include the Von Neumann method, matrix analysis methods, and energy analysis methods. In practical applications, the finite difference method is widely used to solve various partial differential equations, especially excelling in handling problems in regular regions.

Traditional numerical methods face significant challenges when dealing with complex scenarios. When the exact analytical solution of a partial differential equation is difficult to obtain, these methods obtain an approximate solution by discretizing the solution domain into several grid points and providing solution values at the discrete grid points. Although this discretization is effective, there is always a trade-off between computational speed and accuracy; the higher the solution accuracy, the greater the computational cost.

The development trend of modern numerical solution methods points to the design of algorithms with higher efficiency and stronger adaptability. Upwind schemes, as a type of numerical discretization method for solving partial differential equations, have been widely used in computational fluid dynamics. The proposed methods have effectively promoted the development of partial differential equation solving, laying an important theoretical and practical foundation for subsequent machine learning-based solution methods.

4. Application of PINN in Solving Partial Differential Equations

4.1. Construction and Training of PINN

4.1.1. Data Generation and Preprocessing

In the process of solving partial differential equations based on physically informed neural networks, data generation and preprocessing are key steps to ensure the effectiveness of model training. The PINN method transforms the solution of partial differential equations into an optimization problem by embedding physical laws into the loss function. This process requires a carefully designed data generation strategy to support the effective training of the neural network.

Traditional neural network training requires a large amount of labeled data, while the advantage of PINN is that the constraints of physical laws can compensate for the errors caused by insufficient data, and good prediction accuracy can be obtained without a large amount of data training. The data generation process mainly includes discretization sampling of the solution domain, construction of boundary conditions and initial conditions, and selection of residual points of the physical equations. For a given partial differential equation form:

$$\mathrm{N}[u]=f \tag{5}$$

Where N is a nonlinear operator with parameters θ, Ω is the solution domain, $\partial\Omega$ is the boundary of the solution domain.

The data preprocessing process usually involves the coordination of several key steps. Data import and cleaning are the basic steps, which require the detection of outliers, the search for missing values, and the appropriate encoding of classified data. In the specific application scenarios of PINN, preprocessing also includes the accurate identification of the physical domain boundaries and the normalization of physical parameters. Table 2 shows the different data preprocessing requirements. The proper division of the training, validation, and test datasets is equally crucial. These three datasets should possess similar statistical properties to ensure the model's generalization ability.

Table 2. Different Data Preprocessing Requirements

Data Type	Generation Strategy	Preprocessing Requirements	Function
Boundary Condition Data	Boundary Sampling	Normalization	Constraining the Boundary Behavior of the Solution
Initial Condition Data	Time Start Sampling	Dimensionality Unification	Determining the Initial State of the Solution
Physical Equation Data	Random sampling within the domain	Residual calculation	Ensuring consistency of physical laws
Observational data	Experimental measurement	Noise filtering	Providing a true reference

Through the data generation and preprocessing workflow designed above, PINN can effectively combine sparse observation data with rich prior physical knowledge to achieve high-precision solutions to complex partial differential equations. This data-driven approach combined with physical constraints provides a new technical path for solving high-dimensional, nonlinear problems that are difficult to handle with traditional numerical methods.

4.1.2. Network Structure and Optimization Algorithm

The architecture design of a physical information neural network directly affects its efficiency and accuracy in solving partial differential equations. The selection of network structures lacks a universal method and relies heavily on experience, which greatly complicates the design of neural network structures. The core of PINN is to guide the architecture design, optimization, and model training of deep neural networks through physical laws or scientific knowledge, such as feature data processing, loss function construction, and parameter weight design.

A typical PINN network structure uses a feedforward neural network. The input layer receives spatial coordinates and time variables, and the output layer generates the physical quantity to be solved. The number of hidden layers and neurons needs to be adjusted according to the specific problem. Studies have shown that adding more hidden layers does not necessarily improve performance and may even have negative effects. To address this issue, the NAS-PINN method was proposed. It utilizes neural architecture search to guide PINN in automatically finding the most suitable network structure for solving specific partial differential equations. Its loss function is shown below:

$$L_{total}=L_{PDE}+\lambda_1 L_{BC}+\lambda_2 L_{IC} \tag{6}$$

where L_{PDE} represents the residual loss of the partial differential equation, L_{BC} and L_{IC} represent the boundary condition and initial condition losses respectively, λ_1 and λ_2 are the weight coefficients.

The optimization algorithm for PINN employs gradient descent and its variants, training network parameters by minimizing the total loss function. The network structure parameters are shown in Table 3. Based on the automatic differentiation mechanism of neural networks, PINN can naturally embed physical models represented by differential equations, enabling the characterization of complex nonlinear physical processes. During optimization, it is necessary to balance the weights of various loss functions to ensure that the network satisfies physical constraints while possessing good generalization ability.

Table 3. Network Structure Parameter Settings

Network Structure Parameters	Recommended Range	Influencing Factors
Number of Hidden Layers	3-8 Layers	Problem Complexity
Number of Neurons	50-200 Layers	Input Dimension
Activation Function	tanh/sin	Function Smoothness
Learning Rate	1e-3 to 1e-5	Convergence Stability

4.2. Application Case Analysis

4.2.1. Classical Partial Differential Equations Case

The PINN method has demonstrated significant advantages and wide applicability in solving classical partial differential equations. When Raissi et al. proposed the PINN concept in 2018, they verified the reliability of the method in predicting solutions to partial differential equations without relying on labeled data through several classical physics problems. This innovative method embeds physical constraints into neural network training, realizing the solution of partial differential equations under unsupervised learning mode.

In applications within fluid mechanics, PINN has demonstrated its ability to handle complex equations. Jin et al. successfully solved the incompressible Navier-Stokes equations using PINN and conducted an in-depth comparison of the effects of velocity-pressure and velocity-vorticity schemes on PINN performance. Cai et al. applied PINN to solve the Navier-Stokes equations for coupled heat transfer, achieving good results in the study of thermal convection problems. These application examples fully illustrate PINN's potential in handling complex fluid dynamics problems.

The heat conduction equation, a classic parabolic partial differential equation, is also an important application area of PINN. Researchers utilized the function approximation capability of PINN to achieve simultaneous inversion of the coefficients and initial values of the heat conduction equation by optimizing the least-squares fitting function and initial boundary conditions. This method leverages the superior function fitting capability of deep neural networks to guide the iterative update process of network parameters, yielding a high-precision numerical solution.

In the field of nuclear engineering, PINN, as a meshless numerical solution method for partial differential equations, has been successfully applied to solving neutron diffusion and neutron transport equations. Although there is a problem of network non-reusability, with appropriate improvement strategies, PINN can still provide new ideas for traditional numerical problem solving.

4.2.2. Engineering Practical Problem Case

In practical engineering applications, PINN has demonstrated powerful problem-solving capabilities and broad application prospects. Laser powder bed melting (PINN) is a crucial technology in additive manufacturing, and its temperature field prediction and molten pool size control have always been research hotspots. PINN, based on deep neural networks, achieves function approximation, solving both the forward and backward problems in the AM process. Operating in an unsupervised learning mode, it iteratively adjusts the parameters of the deep neural network to ensure that the network output satisfies the control partial differential equations at specific points within the domain, thereby predicting key behaviors such as temperature field or molten pool size.

Transient stability assessment of power systems is another typical engineering application. The core of PINN is to guide the architecture design, optimization, and model training of deep neural networks using physical laws or scientific knowledge, including feature data processing, loss function construction, and parameter weight design. Machine learning models embedded with physical knowledge have advantages such as ensuring consistency between predicted results and physical results, improving model interpretability, and reducing the search space of neural network weights. Based on the automatic differentiation mechanism of neural networks, PINN can naturally embed physical models represented by differential equations, enabling the characterization of complex nonlinear physical processes.

Time-varying reliability analysis methods offer new insights for the application of PINN in structural engineering. PINN incorporates prior physical knowledge, expressed as partial differential equations, into the network training process, enabling the network model to automatically satisfy predefined physical constraints. By using random variables and spatiotemporal coordinates from the system state parameters as inputs and the system response as the output, it can predict the system response at any spatial location. This method demonstrates great potential in fields such as structural health monitoring in civil engineering and fatigue analysis of aerospace components.

5. Research on PINN Optimization Strategies

5.1. Model Performance Evaluation and Improvement

5.1.1. Error Analysis and Model Tuning

Error analysis is the core step in evaluating the performance of physical information neural networks in the process of solving partial differential equations. PINN embeds physical constraints into neural network training and can directly obtain samples through equation definition without preparing labeled data. In practical applications, model errors mainly come from two aspects: the degree of satisfaction of physical constraints and the accuracy of numerical approximation.

The quantitative analysis of errors usually adopts a multiple loss function construction method, including partial differential equation residuals, boundary condition errors, and initial condition deviations. PINN utilizes the powerful function fitting ability of deep neural networks to guide the iterative updating process of adjustable parameters by optimizing the least squares fitting function corresponding to the equation and the least squares fitting function corresponding to the initial and boundary conditions. The total loss function can be expressed as:

$$L_{total}=\lambda_1 L_{PDE}+\lambda_2 L_{BC}+\lambda_3 L_{IC} \tag{7}$$

where L_{PDE} represents the residual loss of the partial differential equation,L_{BC}Boundary condition loss, L_{IC} Initial condition loss, λ_i Corresponding weight coefficients.

The model adjustment strategy is dynamically optimized based on the error analysis results. Different error model adjustment strategies are shown in Table 4. When the physical constraint satisfaction is low, the weight coefficient of the physical loss term needs to be increased; when the numerical approximation accuracy is insufficient, the network structure needs to be adjusted or the training sampling point density needs to be increased. PINN constrains the neural network model through existing physical laws and prior knowledge, guiding the model's training process to satisfy predetermined physical laws, thereby improving the model's interpretability and generalization ability.

Table 4. Adjustment strategies for different error models

Error type	Evaluation index	Adjustment strategy	Applicable scenarios
Physical constraint error	Mean of PDE Residuals	Increase weight λ_1	Equation constraints not satisfied
Boundary condition error	Boundary point deviation	Increase boundary sampling points	Insufficient accuracy at the boundary
Numerical approximation error	L2 Norm Error	Adjust network depth	Poor overall fitting effect

Model adjustment in practice requires a comprehensive consideration of the balance between computational efficiency and solution accuracy. By iteratively optimizing weight allocation and network parameters, PINN can achieve high-precision numerical solutions for complex partial differential equations while ensuring physical consistency.

5.1.2. Selection and Expansion of Training Data

In practical applications of PINN, the quality and distribution of training data directly affect the model's accuracy and generalization ability in solving partial differential equations. Since PINN is based on DNNs for function approximation, it iteratively adjusting the neural network parameters to ensure the network output satisfies the control partial differential equations at specific points within the domain makes the selection strategy for training data particularly crucial. In configuring the training data, it is necessary to reasonably balance the distribution of initial and boundary value conditions and sampling points within the domain to ensure the network can accurately capture physical constraints.

Traditional training data selection often employs uniform sampling to generate training points within the solution domain. However, this method may fail to effectively capture local features and boundary layer effects of the solution function. PINN integrates physical prior knowledge, existing in the form of PDEs, into the network training process, enabling the network model to automatically satisfy predefined physical constraints. This provides a new approach to data selection. The adaptive sampling strategy dynamically adjusts the distribution of sampling points based on the current prediction error of the network, focusing on increasing sampling density in high-error regions to improve overall solution accuracy.

Data augmentation techniques can effectively improve the training effectiveness and generalization ability of PINN models. The data augmentation method that maintains physical constraints can increase the diversity of training samples and improve the robustness and adaptability of the model by performing transformation operations on the training data that conform to physical laws. It should be noted that the generated extended data must maintain consistency with the original physics problem and avoid introducing training samples that violate physical laws. In this case, the loss function of the model is as follows:

$$L_{total}=L_{PDE}+\lambda_{BC}L_{BC}+\lambda_{IC}L_{IC}+\lambda_{data}L_{data} \quad (8)$$

The reasonable configuration of the weights in the loss function is crucial for balancing the contributions of different types of training data, among whichL_{PDE}Represents the loss due to the physical equations,L_{BC}andL_{IC}Represents the losses due to boundary conditions and initial conditions, respectively.

5.2. Efficient Computation and Performance Optimization

5.2.1. Parallel Computing Techniques

Physically informed neural networks (PINNs) face a huge computational burden when solving complex partial differential equations. Traditional serial computation modes are insufficient to meet the real-time solution requirements of large-scale problems. The introduction of parallel computing technology provides an important solution for PINN performance optimization. By decomposing and distributing computational tasks to multiple processing units, the training efficiency and prediction speed of the model are significantly improved.

In the parallel computing implementation of PINN, data parallelism is the most direct and effective strategy. By dividing the physical constraint point set and boundary condition point set into multiple subsets, each computing node is responsible for processing the physical information embedding computation of a specific region. This method is particularly suitable for solving partial differential equations in large-scale spatial domains and can effectively reduce the memory pressure of a single node. Model parallelism distributes different layers of the deep neural network to different processors, achieving parallelization of forward and backward propagation of the network through a pipelined computation mode.

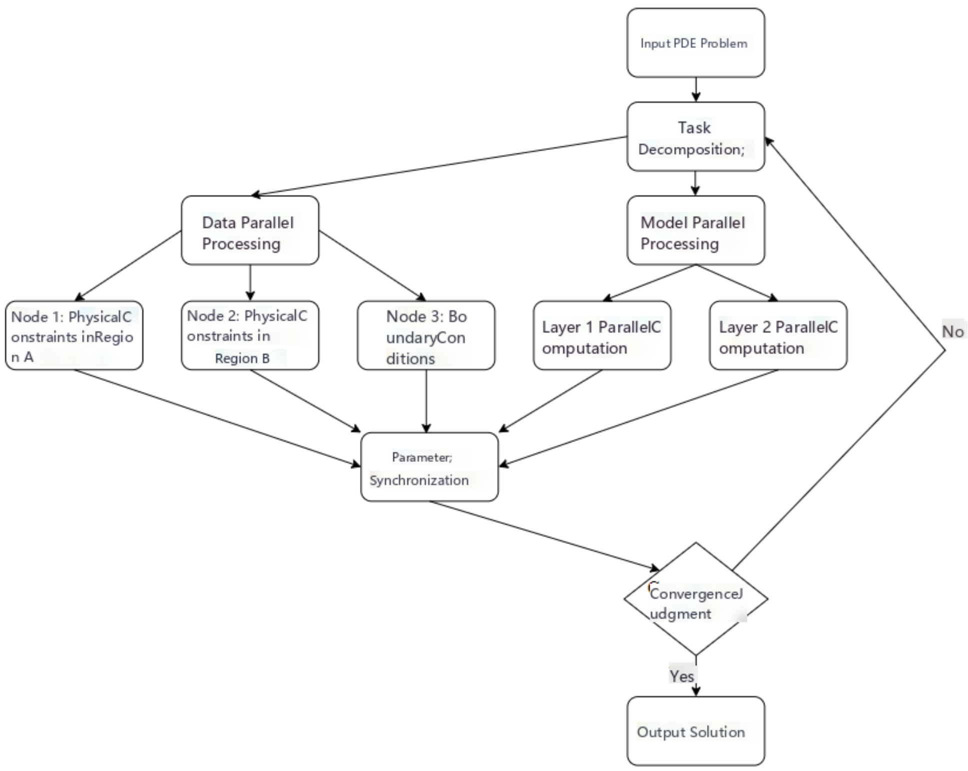

Figure 1. Distributed Training Framework

The application of a distributed training framework (Figure 1) further expands the computational capabilities of PINN. Through a parameter server architecture or a fully reduced communication mode, multiple computing nodes can collaboratively complete gradient calculations and parameter updates. The asynchronous update strategy allows each node to perform calculations at different speeds, avoiding performance bottlenecks caused by synchronous waiting. The introduction of GPU clusters enables PINN to handle millions of physical constraint points, significantly improving its ability to solve complex geometric domains and multiphysics coupled problems.

5.2.2. Hardware Acceleration and Resource Allocation

Physically informed neural networks (PINNs) face a significant computational burden when solving complex partial differential equations, especially when dealing with high-dimensional problems and long-term series predictions. Traditional CPU computing resources often struggle to meet the requirements for real-time performance and accuracy. The introduction of hardware acceleration technologies provides crucial support for the efficient operation of PINNs, with GPU parallel computing architectures significantly improving the speed of neural network training and inference.

In GPU-accelerated implementations, PINN's automatic differentiation computation process fully leverages the parallel capabilities of graphics processing units (GPUs) to decompose the physical constraints of partial differential equations into a large number of parallel computational tasks. Modern GPU architectures, such as NVIDIA's CUDA platform and AMD's ROCm platform, provide computation libraries optimized for deep learning. These libraries can automatically optimize memory access patterns and computation scheduling strategies. Resource allocation strategies need to be dynamically adjusted based on the specific type of partial differential equation and the complexity of the solution domain, including batch size selection, memory allocation strategies, and load balancing for multi-GPU collaborative computation.

In addition to traditional GPU acceleration solutions, emerging dedicated AI chips such as TPUs and NPUs also provide new acceleration possibilities for PINN. These dedicated processors are specifically optimized for matrix operations and activation function calculations in neural networks, achieving higher energy efficiency in specific PINN application scenarios. The elastic resource allocation mechanism of cloud computing platforms allows researchers to dynamically apply for and release computing resources according to computing needs. This on-demand allocation model is particularly suitable for the characteristics of large changes in computing load during PINN training.

6. Conclusion and Outlook

This study delves into the numerical solution optimization method for partial differential equations based on a physics-informed neural network (PINN). Through a combination of theoretical analysis and practical application, the effectiveness and superiority of PINN in the field of partial differential equation solving are verified. PINN constrains the neural network model with existing physical laws and prior knowledge, guiding the model's training process to meet predetermined physical laws, thereby improving the model's interpretability and generalization ability. The constraints of physical laws can also compensate for errors caused by insufficient data, achieving good prediction accuracy without the need for extensive data training.

The core idea of PINN is to embed physical laws into the loss function, transforming the solution of partial differential equations into an optimization problem. The optimization variables are network parameters, and the optimization objective is to minimize physical loss function terms such as the residuals of the governing equations, boundary conditions, and initial conditions. Compared to traditional numerical methods, the PINN solver overcomes the curse of dimensionality and has high temporal and spatial efficiency. PINN innovatively uses automatic differentiation techniques to solve for the derivatives of the target parameters with respect to the inputs, thus bypassing traditional methods based on symbolic computation or numerical difference to accurately and efficiently derive the required derivatives of various orders.

Future research can focus on several key areas. PINN can be used to handle complex nonlinear partial differential equations that are difficult to solve using traditional numerical methods, or inverse problems lacking sufficient observation data. With the continuous development of deep learning technology, PINN has broad application prospects in handling multi-physics coupled problems, high-dimensional partial differential equations, and irregular geometric domains. Meanwhile, further research is needed on how to optimize the network structure, improve training algorithms, and increase computational efficiency to promote the widespread application of PINN technology in engineering practice.

References

[1] Zhang Tao, Du Le, Deng Drunken Tea Inverse radiation coefficient of nonlinear parabolic equation based on terminal observation conditions [J]. Journal of Lanzhou Jiaotong University, 2024

[2] Shi Jinjin, Song Ning, Tian Hao, Nie Jie, Wei Zhiqiang. A method for accelerating the numerical solution of one-dimensional advection equations based on spatial discretization of graph convolution [J]. Journal of Ocean University of China (Natural Science Edition), 2024.

[3] Gao Yali Research on Heuristic Teaching of Poisson Equation with Finite Difference Theory [J]. Advanced Mathematics Research, 2024

[4] Hu Weifei, Liao Jiale, Guo Yunfei, Yan Jiquan, Li Guang, Yue Haifeng, Tan Jianrong. A time-varying reliability analysis method based on physical information neural network [J]. Journal of Mechanical Engineering, 2024.

[5] Chen Zhenxing, Wan Guanghong, Ruan Zhousheng Numerical calculation of simultaneous inversion of coefficients and initial values of one-dimensional heat conduction equation based on local observation data [J]. Journal of Jinggangshan University (Natural Science Edition), 2024

[6] Xie Yuchen, Ma Yu, Wang Yahui Parameterized Physics Embedded Neural Network for Solving Steady State Single Energy Neutron Diffusion Solution Set [J]. Atomic Energy Science and Technology, 2024

[7] Ren Zhuoxiang, Yan Shuai, Chen Zhifu, Xu Xiaoyu, Fu Xiaotong, Guo Ze, Zhou Yaxing, Yang Fan. Research status and trends of several key issues in low-frequency electromagnetic simulation of electrical equipment [J]. High Voltage Engineering, 2024.

[8] Yan Xin, Ying Kaining, Dai Lunan, Tan Junfu, Shen Zhonghua, Ni Chenyin. Research on laser ultrasonic field based on physical information neural network [J]. Laser Technology, 2024.

[9] Yan Ruqiang, Shang Zuogang, Wang Zhiying, Xu Wengang, Zhao Zhibin, Wang Shibin, Chen Xuefeng. Challenges and Opportunities of Explainable Artificial Intelligence in Industrial Intelligent Diagnosis: Prior Empowerment [J]. Journal of Mechanical Engineering, 2024.

[10] Tian Songyan, Huang Xinge, Duan Yanhui, Chen Hongbo, Chen Wenxiu Overview of Physical Neural Networks for Fluid Dynamics [J]. Computer Applications, 2024

[11] Wang Xinlian, Li Jie, Wan Jie, Yuan Ruihao, Li Jinshan, Wang Jun. Methodology for Optimization of Laser Powder Bed Melting Process Parameters, Process Monitoring and Service Life Prediction Based on Machine Learning [J]. Foundry Technology, 2024.

[12] Lu Xu, Zhang Liyin, Li Gengfeng, Bie Chaohong, Duan Chao. Transient Stability Assessment of Power Systems Based on Convolutional Neural Networks with Embedded Physical Knowledge [J]. Automation of Electric Power Systems, 2024.

[13] Fan Lilin, Liu Shihao, Li Yuan, Mao Wentao, Chen Zongtao. Progressive Training Strategy of Physical Information Neural Network Based on Curriculum Regularization [J]. Journal of Shandong University (Engineering Science), 2024.

[14] Li Xiaolei. Analysis of Adaptive Neural Network Method for Solving Burger Equation [J]. Electronic Technology, 2024.

[15] Wang Xinyun, Tang Xuefeng, Yu Guoqing, Long Jinchuan, Wen Hongning, Deng Lei, Jin Junsong, Gong Pan, Zhang Mao. Research Progress on Plastic Deformation Theory Modeling Based on Deep Learning [J]. Journal of Plasticity Engineering, 2024.