IJARSCT

ISSN (Online) 2581-9429

International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)

International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal

Impact Factor: 7.301

Volume 3, Issue 3, December 2023

Deep Learning Approaches for High-Dimensional Partial Differential Equations Solution

N. Srinivas[1] and Dr. Manjeet Jakhar[2]

Research Scholar, Department of Mathematics[1]
Research Guide, Department of Mathematics[2]
NIILM University, Kaithal, India

Abstract: *This abstract explores the application of deep learning techniques for solving high-dimensional partial differential equations (PDEs). High-dimensional PDEs pose significant challenges in traditional numerical methods due to the curse of dimensionality, making them computationally expensive and often infeasible. Deep learning, specifically neural networks, has shown promise in efficiently approximating complex functions and handling high-dimensional data. This paper reviews various deep learning approaches, such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs), applied to the numerical solution of high-dimensional PDEs. The study discusses the advantages and limitations of these techniques, highlighting their potential to enhance accuracy and computational efficiency in comparison to classical methods. Additionally, it addresses the incorporation of domain knowledge and the exploration of hybrid methodologies to further improve the robustness and generalization of deep learning models in tackling challenging high-dimensional PDE problems.*

Keywords: Numerical Methods, Scientific Computing, Machine Learning

I. INTRODUCTION

Deep learning approaches for high-dimensional partial differential equation (PDE) solutions have emerged as powerful tools in the realm of scientific computing and computational mathematics. These methods leverage neural networks to approximate complex solutions to PDEs in high-dimensional spaces, offering a promising alternative to traditional numerical methods. By encoding the underlying dynamics of the system into the network architecture, deep learning models can efficiently learn and generalize intricate patterns, making them particularly effective for problems with large parameter spaces or intricate geometries. These approaches have shown great success in a variety of fields, including fluid dynamics, quantum mechanics, and materials science, paving the way for more accurate and computationally efficient solutions to challenging high-dimensional PDE problems.

Deep Neural Networks

Deep neural networks (DNNs) have emerged as powerful tools for approximating high-dimensional solutions to partial differential equations (PDEs) in the realm of deep learning. Traditional numerical methods often face challenges when dealing with complex, high-dimensional PDEs due to the curse of dimensionality. DNNs offer a promising alternative by learning intricate mappings from input to output spaces without requiring explicit knowledge of the underlying PDE. These networks can capture complex patterns and dependencies, allowing for efficient and accurate approximations of PDE solutions. A common approach involves training DNNs to minimize the discrepancy between predicted and true solutions, often expressed as a loss function, which incorporates the PDE constraints. Mathematically, this can be formulated as:

$$\text{minimize } \mathcal{L}(\theta) = \frac{1}{N}\sum_{i=1}^{N} \|\mathcal{F}(u_i, x_i, \theta)\|^2$$

where L(θ) is the loss function, θ represents the DNN parameters, F denotes the PDE operator, $u_i$is the true solution, and $x_i$are the input parameters. By training DNNs on a dataset of PDE solutions, these networks can generalize well to unseen scenarios, providing a versatile and efficient approach for high-dimensional PDE problems.

Deep Residual Learning

Deep residual learning has emerged as a powerful paradigm in deep learning, particularly for addressing challenges associated with high-dimensional partial differential equation (PDE) solutions. The essence of residual learning lies in the introduction of shortcut connections, enabling the network to learn residual functions rather than explicitly modeling the desired mapping. In the context of high-dimensional PDE solutions, where traditional approaches may struggle due to the curse of dimensionality, deep residual networks (Res Nets) offer advantages in capturing complex dependencies and improving convergence. The key mathematical formulation of a residual block is given by the equation $H(x) = F(x) + x$ where $H(x)$ represents the desired mapping, $F(x)$ denotes the residual function to be learned, and x is the input. This architecture facilitates the training of deep networks by mitigating vanishing or exploding gradient issues, making it well-suited for efficiently approximating high-dimensional PDE solutions in various scientific and engineering domains.

Deep Learning for Random PDEs: A Strong form

Deep learning approaches have gained significant traction in addressing the challenges posed by high-dimensional partial differential equations (PDEs), offering a promising avenue for efficiently approximating solutions to complex problems. One notable approach involves formulating a strong form representation for random PDEs, where the solution is sought directly through a neural network. The strong form can be expressed as follows:

$$F(u(x,t), \nabla u(x,t), \nabla^2 u(x,t), \ldots, \nabla^n u(x,t); x, t, \omega) = 0$$

where)$u(x,t)$ denotes the solution of the PDE at spatial coordinate x, temporal coordinate t, and randomness source ω. The function F encapsulates the PDE itself, incorporating the various partial derivatives up to the nth order. Leveraging deep neural networks, such as convolutional neural networks (CNNs) or recurrent neural networks (RNNs), to represent the solution u allows for the approximation of high-dimensional PDE solutions in a data-driven manner. Training the neural network involves minimizing the loss function associated with the strong form, effectively optimizing the network parameters to yield accurate solutions. This paradigm shift toward data-driven methodologies for solving high-dimensional PDEs not only addresses the computational challenges but also opens up new possibilities for handling random PDEs, where traditional methods may fall short. The integration of deep learning techniques in this context showcases their versatility and efficacy in tackling complex problems with inherent uncertainty.

Deep learning for random PDEs: A variational form and a lower order loss function

Deep Learning (DL) has emerged as a powerful tool for approximating solutions to high-dimensional Partial Differential Equations (PDEs), particularly in the context of random PDEs. One promising avenue in this domain involves the formulation of a variational form coupled with a lower-order loss function. The variational approach leverages the flexibility of neural networks to approximate the solution by optimizing over a function space, enabling the model to capture complex and high-dimensional dependencies inherent in random PDEs. This formulation often takes the form of minimizing a functional involving the PDE operator and the neural network's output. Mathematically, this can be expressed as:

$$\underset{u}{\text{minimize}}\mathcal{L}(u,\theta) = \mathbb{E}_{\omega}[\mathcal{F}(u(\omega), \nabla u(\omega), \theta)]$$

Here, $u(\omega)$ represents the PDE solution dependent on the random parameter ω, $\nabla u(\omega)$ denotes the gradient of the solution, and θ represents the parameters of the neural network. The functional F encapsulates the PDE operator and additional terms that ensure the solution satisfies the PDE constraints. The expectation is taken over the random parameter space, reflecting the stochastic nature of the PDE.

To address the challenges associated with high-dimensional problems, a lower-order loss function is often introduced. This auxiliary loss function regularizes the training process and encourages the model to learn low-dimensional features that capture the essence of the solution. It can be expressed as:

$$\mathcal{L}_{\text{low-order}}(\theta) = \lambda \cdot \left\| \nabla_{\omega} u(\omega) - \nabla_{\omega} \hat{u}(\omega,\theta) \right\|^2$$

Here, $u\wedge(\omega,\theta)$ is the neural network's output, and $\nabla\omega u\wedge(\omega,\theta)$ is its gradient with respect to the random parameter. The regularization term penalizes deviations in the gradient space, promoting smoother solutions and aiding generalization to unseen scenarios. The hyperparameter λ controls the strength of this regularization.

In summary, the integration of a variational form and a lower-order loss function provides a robust framework for leveraging deep learning in the context of high-dimensional random PDEs. This approach not only enables the model to efficiently learn the underlying structure of the solution space but also addresses the challenges posed by the curse of dimensionality. The mathematical expressions presented capture the essence of the proposed methodology, emphasizing the optimization of a functional over the solution space and the introduction of a regularization term to enhance the model's ability to handle high-dimensional scenarios. This combination holds promise for advancing the state-of-the-art in the numerical approximation of solutions to random PDEs through deep learning techniques.

Numerical Examples

Deep learning approaches have shown remarkable success in solving high-dimensional partial differential equations (PDEs), offering a powerful alternative to traditional numerical methods. Consider the heat equation as an example:

$$\frac{\partial u}{\partial t} = \alpha \nabla^2 u$$

where u is the temperature distribution over space x and time t, and α is the thermal diffusivity. Traditional finite difference methods struggle with high-dimensional problems, but deep learning models, such as neural networks, can efficiently approximate complex solutions. For instance, a convolutional neural network (CNN) can be trained on a dataset of spatiotemporal temperature profiles, learning the underlying dynamics and providing accurate predictions for unseen scenarios. This not only enhances computational efficiency but also allows for generalization to previously unexplored regions of the high-dimensional solution space, making deep learning an attractive and versatile tool for solving complex PDEs in various scientific and engineering applications.

Transient diffusion problem with random coefficient

Deep learning approaches have shown promising capabilities in tackling transient diffusion problems with random coefficients, particularly in the context of high-dimensional partial differential equation (PDE) solutions. These methods leverage neural networks to learn and approximate the complex spatiotemporal patterns inherent in the diffusion process. The governing PDE for transient diffusion with random coefficients can be expressed as:

$$\frac{\partial u}{\partial t} = \nabla \cdot (D(\mathbf{x}, t) \nabla u) + f(\mathbf{x}, t)$$

where u represents the unknown solution, t is time, ∇∇ denotes the gradient operator,$D(\mathbf{x},t)$ is the random diffusion coefficient, and $f(\mathbf{x},t)$ is a source term. Deep learning models, such as neural networks, are employed to approximate the solution u by learning the underlying dynamics from a dataset. This enables the efficient computation of high-dimensional solutions, offering a data-driven approach for solving transient diffusion problems with random coefficients.

A smooth random field for the diffusion coefficient

Deep Learning Approaches have emerged as powerful tools for tackling high-dimensional Partial Differential Equation (PDE) solutions. In the context of diffusion processes, modeling the diffusion coefficient as a smooth random field is a novel and promising approach. The diffusion coefficient, denoted by $D(\mathbf{x})$, is treated as a spatially varying function defined over the domain $\Omega \subseteq \mathrm{R}n$, where $\mathbf{x}$ represents the spatial coordinates. Leveraging deep neural networks, one can efficiently learn and parameterize this field, capturing intricate dependencies in high-dimensional PDE systems. Mathematically, this can be expressed as:

$$D(\mathbf{x}) = f_\theta(\mathbf{x})$$

Here, $f\theta(\mathbf{x})$ represents a deep neural network with parameters θ, and it is trained to approximate the diffusion coefficient at any spatial location $\mathbf{x}$. Such an approach enables the effective incorporation of complex spatial variations in diffusion, improving the accuracy and efficiency of high-dimensional PDE solutions through the expressive power of deep learning architectures.

A non-smooth random field for the diffusion coefficient

In the realm of solving high-dimensional partial differential equations (PDEs) using Deep Learning (DL) approaches, the incorporation of non-smooth random fields for the diffusion coefficient represents a significant advancement. Traditionally, PDEs involve smooth deterministic coefficients, but real-world applications often exhibit complex and irregular behavior. Introducing non-smoothness in the diffusion coefficient allows for a more accurate representation of the underlying physical processes. Deep Learning models, such as neural networks, have shown promise in capturing and learning from such intricate patterns. Mathematically, this can be expressed by considering a PDE with a non-smooth diffusion coefficient $D(x)$, where x denotes the spatial coordinates:

$$-\nabla \cdot (D(x)\nabla u(x)) + f(x,u) = 0$$

Here, $u(x)$ represents the solution to the PDE, and $f(x,u)$ is a source term. Leveraging DL techniques, these high-dimensional PDEs with non-smooth coefficients can be efficiently and accurately solved, paving the way for enhanced understanding and simulation of complex physical phenomena.

Steady heat equation with random thermal conductivity

Deep Learning Approaches for High-Dimensional PDE Solutions have demonstrated promising capabilities in addressing challenges posed by the steady heat equation with random thermal conductivity. The conventional numerical methods encounter difficulties in efficiently handling high-dimensional spaces and complex parameter variations. Leveraging deep neural networks, these approaches have shown efficacy in capturing intricate patterns and relationships within the high-dimensional solution space. By employing techniques such as neural networks with architectures tailored for function approximation, the steady heat equation with random thermal conductivity can be efficiently solved, enabling accurate predictions and insights into the system's behavior. The key innovation lies in the network's ability to learn and generalize from the available data, adapting to variations in the thermal conductivity. The fundamental equation governing the steady heat conduction with random thermal conductivity (κ) can be expressed as:

$$-\nabla \cdot (\kappa \nabla u) = 0$$

where u represents the temperature distribution and $\nabla\nabla$ denotes the gradient operator. Deep learning techniques, such as neural network-based solvers, offer a promising avenue to tackle the computational challenges associated with high-dimensional parameter spaces in solving such PDEs.

Steady heat equation with random thermal conductivity on a domain with a hole

Deep Learning Approaches have proven effective in tackling high-dimensional Partial Differential Equation (PDE) problems, such as the steady heat equation with random thermal conductivity on a domain featuring a hole. The PDE governing the steady heat distribution in this scenario can be expressed as follows:

$$\nabla \cdot (k(\mathbf{x})\nabla u(\mathbf{x})) = 0$$

where $u(\mathbf{x})$ is the temperature distribution, $k(\mathbf{x})$ represents the spatially varying thermal conductivity with randomness introduced in the presence of a hole in the domain. Deep learning techniques, particularly neural networks, can be employed to approximate the solution $u(\mathbf{x})$ by learning the complex patterns inherent in the high-dimensional and stochastic nature of the problem. This involves training the neural network on a dataset generated from simulations or observations, allowing it to generalize and provide accurate predictions for temperature distributions in the presence of random thermal conductivity and domain irregularities. These approaches provide a powerful and versatile framework for solving challenging PDEs in diverse physical scenarios.

II. DISCUSSION

Deep learning approaches have shown great promise in solving high-dimensional partial differential equations (PDEs), offering a powerful alternative to traditional numerical methods. One notable technique is the use of neural networks, particularly deep neural networks (DNNs), to approximate complex solutions of PDEs. The ability of DNNs to capture intricate patterns and relationships within data makes them well-suited for tackling high-dimensional problems. One common approach involves training neural networks to learn the mapping between the input parameters of a PDE and its corresponding solution. Mathematically, this can be expressed as $u = F(x;\theta)$ where u is the solution, x represents the input parameters, and θ denotes the parameters of the neural network. The training process involves optimizing θ to minimize the difference between the predicted and actual solutions, effectively enabling the network to generalize and provide accurate solutions for a wide range of input scenarios. Despite the challenges associated with high-dimensional PDEs, deep learning methodologies offer a promising avenue for efficient and accurate solutions in various scientific and engineering applications.

III. CONCLUSION

Deep learning approaches have demonstrated remarkable potential for addressing high-dimensional partial differential equation (PDE) solutions. The ability of neural networks to capture complex patterns and relationships in large datasets makes them particularly well-suited for handling the intricacies of high-dimensional problems. These approaches leverage the expressive power of deep learning architectures, such as convolutional neural networks (CNNs) or recurrent neural networks (RNNs), to efficiently approximate solutions to PDEs with numerous variables. Their success lies in their capacity to learn and generalize from vast amounts of data, enabling the modeling of intricate and nonlinear relationships within high-dimensional spaces. While challenges such as interpretability and computational demands persist, the promising results achieved thus far suggest that deep learning approaches hold significant potential in advancing the numerical solution of high-dimensional PDEs, opening up new avenues for tackling complex problems in diverse fields, including physics, finance, and engineering.

REFERENCES

[1] G. Fishman, Monte Carlo: Concepts, Algorithms, and Applications, Springer Science & Business Media, 2013.

[2] W.K. Liu, T. Belytschko, A. Mani, Probabilistic finite elements for nonlinear structural dynamics, Comput. Methods Appl. Mech. Engrg. 56 (1) (1986) 61–81.

[3] G. Deodatis, Weighted integral method. I: stochastic stiffness matrix, J. Eng. Mech. 117 (8) (1991) 1851–1864.

[4] F. Yamazaki, M. Shinozuka, G. Dasgupta, Neumann expansion for stochastic finite element analysis, J. Eng. Mech. 114 (8) (1988) 1335–1354.

[5] D. Zhang, Stochastic Methods for Flow in Porous Media: Coping with Uncertainties, Elsevier, 2001.

[6] R.G. Ghanem, P.D. Spanos, Stochastic Finite Elements: A Spectral Approach, Courier Corporation, 2003.

[7] D. Xiu, Numerical Methods for Stochastic Computations: A Spectral Method Approach, Princeton university press, 2010.

[8] D. Xiu, G.E. Karniadakis, The wiener–askey polynomial chaos for stochastic differential equations, SIAM J. Sci. Comput. 24 (2) (2002) 619–644.

[9] D. Xiu, J.S. Hesthaven, High-order collocation methods for differential equations with random inputs, SIAM J. Sci. Comput. 27 (3) (2005) 1118–1139.

[10] A. Doostan, H. Owhadi, A non-adapted sparse approximation of PDEs with stochastic inputs, J. Comput. Phys. 230 (8) (2011) 3015–3034.

[11] G. Blatman, B. Sudret, An adaptive algorithm to build up sparse polynomial chaos expansions for stochastic finite element analysis, Probab. Eng. Mech. 25 (2) (2010) 183–197.

[12] H. Lee, I.S. Kang, Neural algorithm for solving differential equations, J. Comput. Phys. 91 (1) (1990) 110–131.

[13] L. Wang, J.M. Mendel, Structured trainable networks for matrix algebra, in: Neural Networks, 1990., 1990 IJCNN International Joint Conference on, IEEE, 1990, pp. 125–132.

[14] D. Gobovic, M.E. Zaghloul, Analog cellular neural network with application to partial differential equations with variable mesh-size, in: Circuits and Systems, 1994. ISCAS'94., 1994 IEEE International Symposium on, Vol. 6, IEEE, 1994, pp. 359–362.

[15] R. Yentis, M.E. Zaghloul, VLSI Implementation of locally connected neural network for solving partial differential equations, IEEE Trans. Circuits Syst. I 43 (8) (1996) 687–690.

[16] A.J. Meade, A.A. Fernandez, Solution of nonlinear ordinary differential equations by feedforward neural networks, Math. Comput. Modelling 20 (9) (1994) 19–44.

[17] A.J. Meade, A.A. Fernandez, The numerical solution of linear ordinary differential equations by feedforward neural networks, Math. Comput. Modelling 19 (12) (1994) 1–25.

[18] I.E. Lagaris, A. Likas, D.I. Fotiadis, Artificial neural networks for solving ordinary and partial differential equations, IEEE Trans. Neural Netw. 9 (5) (1998) 987–1000.

[19] J. Tompson, K. Schlachter, P. Sprechmann, K. Perlin, Accelerating Eulerian Fluid Simulation With Convolutional Networks, arXiv preprint arXiv:1607.03597.

[20] Y. Suzuki, Neural network-based discretization of nonlinear differential equations, Neural Comput. Appl. (2017) 1–16.

[21] J. Sirignano, K. Spiliopoulos, DGM: A deep learning algorithm for solving partial differential equations, arXiv preprint arXiv:1708.07469.

[22] A. Malek, R.S. Beidokhti, Numerical solution for high order differential equations using a hybrid neural network—optimization method, Appl. Math. Comput. 183 (1) (2006) 260–271.

[23] K. Rudd, Solving Partial Differential Equations Using Artificial Neural Networks (Ph.D. thesis), Duke University, 2013.

[24] E. Weinan, J. Han, A. Jentzen, Deep learning-based numerical methods for high-dimensional parabolic partial differential equations and backward stochastic differential equations, Commun. Math. Stat. (2017) 1–32.

[25] J. Han, A. Jentzen, et al., Overcoming the curse of dimensionality: Solving highdimensional partial differential equations using deep learning, arXiv preprint arXiv:1707.02568.

[26] M. Raissi, P. Perdikaris, G.E. Karniadakis, Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, J. Comput. Phys. 378 (2019) 686–707.

[27] P.B. Bochev, M.D. Gunzburger, Least-Squares Finite Element Methods, Springer, 2006.

[28] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (7553) (2015) 436–444.

[29] I. Goodfellow, Y. Bengio, A. Courville, Deep Learning, MIT press, 2016.

[30] Y. LeCun, I. Kanter, S.A. Solla, Second order properties of error surfaces: Learning time and generalization, in: Advances in Neural Information Processing Systems, 1991, pp. 918–924.

[31] Y.A. LeCun, L. Bottou, G.B. Orr, K.-R. Müller, Efficient backprop, in: Neural Networks: Tricks of the Trade, Springer, 2012, pp. 9–48.

[32] L. Bottou, Stochastic gradient descent tricks, in: Neural Networks: Tricks of the Trade, Springer, 2012, pp. 421–436.

[33] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, arXiv preprint arXiv:1412.6980.

[34] J. Duchi, E. Hazan, Y. Singer, Adaptive subgradient methods for online learning and stochastic optimization, J. Mach. Learn. Res. 12 (Jul) (2011) 2121–2159.

[35] M.D. Zeiler, ADADELTA: an adaptive learning rate method, arXiv preprint arXiv: 1212.5701.

[36] I. Sutskever, J. Martens, G. Dahl, G. Hinton, On the importance of initialization and momentum in deep learning, in: International Conference on Machine Learning, 2013, pp. 1139–1147.

[37] A.G. Baydin, B.A. Pearlmutter, A.A. Radul, J.M. Siskind, Automatic differentiation in machine learning: a survey, J. Mach. Learn. Res. 18 (2018) 1–43.

[38] J. Ba, R. Caruana, Do deep nets really need to be deep?, in: Advances in Neural Information Processing Systems, 2014, pp. 2654–2662.

[39] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, arXiv preprint arXiv:1409.1556.

[40] R.K. Srivastava, K. Greff, J. Schmidhuber, Training very deep networks, in: Advances in Neural Information Processing Systems, 2015, pp. 2377–2385.

[41] R.K. Srivastava, K. Greff, J. Schmidhuber, Highway networks, arXiv preprint arXiv:1505.00387.

[42] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, et al., Going dEeper with Convolutions, Cvpr, 2015.

[43] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.

[44] K. He, J. Sun, Convolutional neural networks at constrained time cost, in: Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on, IEEE, 2015, pp. 5353–5360.

[45] P. Márquez-Neila, M. Salzmann, P. Fua, Imposing Hard Constraints on Deep Networks: Promises and Limitations, arXiv preprint arXiv:1706.02025.

[46] S. Ruder, An overview of gradient descent optimization algorithms, arXiv preprint arXiv:1609.04747.

[47] L. Bottou, Large-scale machine learning with stochastic gradient descent, in: Proceedings of COMPSTAT'2010, Springer, 2010, pp. 177–186.

[48] S. Saarinen, R. Bramley, G. Cybenko, Ill-conditioning in neural network training problems, SIAM J. Sci. Comput. 14 (3) (1993) 693–714.

[49] D. Boffi, F. Brezzi, M. Fortin, Variational formulations and finite element methods, in: Mixed Finite Element Methods and Applications, Springer, 2013, pp. 1–46.

[50] S. Ioffe, C. Szegedy, Batch normalization: Accelerating deep network training by reducing internal covariate shift, arXiv preprint arXiv:1502.03167.