Research Proposal

祝 鏘 ZHU QIANG

Research theme： Improved Physics-Informed Neural Networks

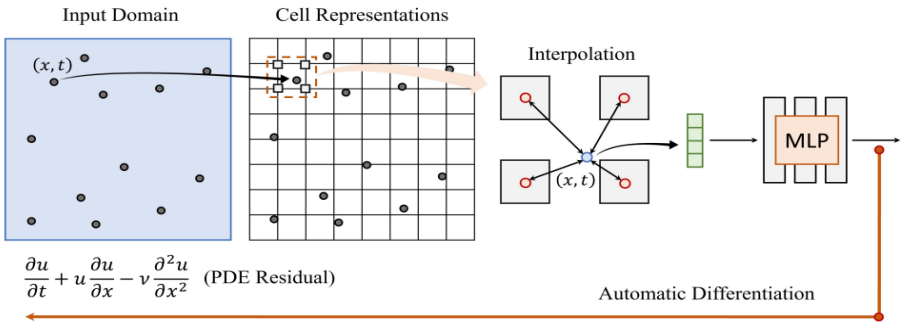
Background

Most physical models are described in the form of partial differential equations. If you choose to solve these models directly, you will face problems such as high complexity and large errors. Therefore, Physics-Informed Neural Networks (PINNs) were proposed. In PINN, we incorporate the differential form constraints in the differential equation into the loss function design of the neural network, thereby obtaining a neural network with physical model constraints. However, Recent studies have shown that spectral bias[1] indeed exists in PINN models[2] and this tendency towards smooth function approximation often leads to failure to accurately capture high-frequency components or singular behaviors in solution functions. So I want to improve its generalization ability for solving other different PDEs.

Previous Research

In my undergraduate thesis, I discussed and explored a grid-based PINN method that improves generalization by combining the advantages of traditional numerical methods and neural networks.

The structure of my method is shown in the figure on the right. The difference from the traditional PINN is that we define the features at the grid points. When calculating the function value information of a certain collocation point, we first perform bilinear interpolation based on the information of the surrounding grid points. In order to support high-order partial derivatives, we introduce kernel functions in bilinear interpolation, such as $k(x) = (1 - \cos(\pi x))/2$, because this function is obviously infinitely differentiable and is universal in almost all partial differential equations.



Let's take a two-dimensional grid as an example. H and W represent the size of the grid in the spatial and temporal dimensions, respectively. x and t represent the normalized coordinates of each collocation point, that is, $x \in [1, H]$, $t \in [1, W]$. We regard the grid as a feature extractor, defined as $\phi(x, t, f)$, where $f \in R^{c \times H \times W}$ represents a tensor consisting of all parameters defined at the grid points, $f_{ij} \in R^c$ represents the trainable parameter vector at a specific grid point (i, j), and c is the number of channels.

Therefore, we define the feature extractor on the right.

$$\phi(x, t, f) = \sum_{i=1}^{H} k(\max(0, 1 - |x - i|)) \sum_{j=1}^{W} k(\max(0, 1 - |t - j|)) f_{ij}$$

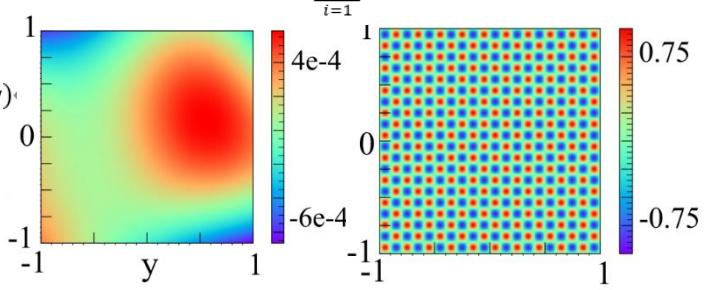
As the grid resolution increases, the accuracy of the solution increases, but the number of collocation points required increases exponentially and is prone to overfitting. Thus, we stack multiple sparse grids, offset a given collocation point in each different grid, so that the input coordinates can be located at different positions in each grid, and calculate each collocation point by adding all grids, which is called multigrid.

we define the feature extractor of the multigrid on the right.

$$\phi_{\text{multi}}(x, t, f) = \sum_{i=1}^{M} \phi\left(x + \frac{(i-1)}{M}, t + \frac{(i-1)}{M}, f^i\right)$$

As an experimental result, we take the two-dimensional Helmholtz equation as an example.

$$\triangle u(x, y) + u(x, y) = q(x, y)$$
$$q(x, y) = k^2 \sin(10\pi x)\sin(10\pi y) - 200\pi^2 \sin(10\pi x)\sin(10\pi y)$$
$$u(x, y) = 0, (x, y) \in \partial[-1, 1]^2$$



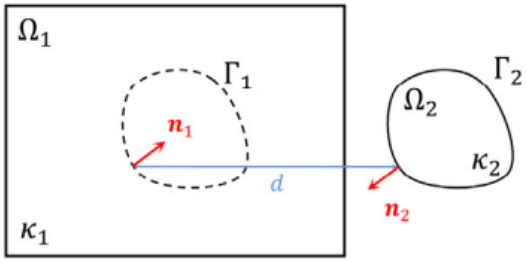
Experimental results show that grid-based PINN method has obvious advantages in dealing with high-frequency PDE problems.

Future Research Content and Methods

In the master's program, I wanted to improve the method I proposed in my undergraduate thesis. Because it cannot handle complex boundary problems. Since my method is based on grid, and PDEs with complex domains have no grid representation, so the boundary conditions of the PDE must be defined on the boundary of the multidimensional cuboid, that is, $(x_1 \ldots x_n) \in \partial([a_1, b_1] \times \cdots \times [a_n, b_n])$

Firstly, inspired by domain separation strategy[3], I considered extending the domain of the PDE to a wider multidimensional cuboid and define the solution to be 0 outside the domain. For example, a diffusion problem

$$-\nabla \cdot \kappa(X)\nabla u = Q(X), X \in \Omega,$$
$$u(X) = g(X), X \in \partial\Omega,$$
$$\kappa(X) = \kappa_i(X), \text{ for } X \in \Omega_i, \ i = 1, 2,$$



with complex boundary conditions can be transformed into a multi-material diffusion problem with simple boundary conditions after the extension.

The structure of this technique is shown in the figure on the left.

The key point of this technique is to take the continuity at the interface of two materials into account in the loss function and use domain separation to capture the fluctuations of the solution function on the interface. [3] proved that for diffusion problems, the solutions of the two domains have the same structure when moving the sampling points, which is the theoretical basis of domain separation.

But this conclusion is not universal. I will theoretically prove similar conclusions in other PDEs, combine this technique with my method, use grid representation in the extended PDE, and use pytorch to build related models.

Secondly, the biggest issue of my method is that there are too many parameters, in fact, each grid point has a c-dimensional parameter vector attached to it (c is the number of channels), which leads to higher computational cost and training time. The core reason is that this method combines traditional numerical methods, so I thought of putting PINN into a pure AI framework.

Inspired by [4], PINN can be used as a tool to combine reinforcement learning and deep learning, and this is still a very new research direction with almost no papers. The advantage of grid-based PINN is that the physical domain is divided into numerous grids to capture the high-frequency components in the solution function, but I guess reinforcement learning may also be able to achieve the same effect.

I think we can use RL to adaptively select training points. By defining a policy, the agent can dynamically choose new sampling points based on the current loss distribution and error information to better cover the physical domain and improve training efficiency.

As for the implementation, inspired by grid-based PINN, we may divide the physical domain into different regions, sample training points in each region, and adjust the sampling strategy based on training errors (rewards). Most importantly, we may no longer need a large number of parameters.

Besides, we may use RL to optimize the strategy for handling complex boundary conditions. We may define different methods for handling boundary conditions as actions, use simulation error as the reward, and employ RL algorithms to optimize the strategy.

Thirdly, inspired by Bayesian physics-informed neural networks[5], Generalization ability is not the only aspect of traditional PINNs that deserves improvement. PINNs need to provide reliable predictions in practical applications, and also need to estimate the uncertainty of the predictions.

By introducing Bayesian neural networks or other uncertainty quantification methods[6,7], PINNs can not only provide prediction values, but also give the uncertainty interval of the prediction, thereby improving the credibility of the model in practical applications.

However, although BNNs bring the advantage of uncertainty quantification, they also increase computational complexity. I consider exploring potentially more efficient Bayesian inference methods, combining variational inference (VI) and Markov chain Monte Carlo (MCMC) methods, to balance accuracy and computational cost.

VI approximates the posterior distribution through optimization, making it faster and more suitable for large-scale data compared to MCMC，but VI often uses simple distributions to approximate complex posterior distributions, which can lead to lower approximation quality. MCMC can precisely approximate posterior distributions, making it suitable for complex distributions, but MCMC methods usually require a large number of samples, leading to high computational overhead, especially in high-dimensional spaces.

I consider using VI to quickly obtain an initial approximate posterior distribution, and then use MCMC to sample from and refine this distribution. This approach maybe maintains computational efficiency while improving approximation quality.

Research Significance

Improving PINNs enhances prediction accuracy, uncertainty quantification, and computational efficiency, broadening their applicability in solving complex scientific and engineering problems.

参考文献

[1] Rahaman, N.; Baratin, A.; Arpit, D.; Draxler, F.; Lin, M.; Hamprecht, F.; Bengio, Y.; and Courville, A. 2019. On the spectral bias of neural networks. In International Conference on Machine Learning, 5301–5310. PMLR.G. Strang: Linear Algebra and Its Applications, 4th ed., Cengage Learning, 2006.

[2] Moseley, B.; Markham, A.; and Nissen-Meyer, T. 2021. Finite Basis Physics-Informed Neural Networks (FBPINNs): a scalable domain decomposition approach for solving differential equations. arXiv:2107.07871.

[3] YAO Y, GUO J, GU T. A deep learning method for multi-material diffusion problems based on physics-informed neural networks[J]. Computer Methods in Applied Mechanics and Engineering, 2023, 417: 116395.

[4] Salvatore Cuomo, Vincenzo Schiano Di Cola, Fabio Giampaolo, Gianluigi Rozza, Maziar Raissi & Francesco Piccialli. Scientific Machine Learning Through Physics–Informed Neural Networks: Where we are and What's Next, 2022

[5] Psaros A F, Meng X, Zou Z, et al. Uncertainty Quantification in Scientific Machine Learning: Methods, Metrics, and Comparisons[J]. arXiv preprint arXiv:2201.07766, 2022

[6] Zhang D, Lu L, Guo L, et al. Quantifying total uncertainty in physics-informed neural networks for solving forward and inverse stochastic problems[J]. Journal of Computational Physics, 2019, 397: 108850.

[7] Yang L, Meng X, Karniadakis G E. B-PINNs: Bayesian physics-informed neural networks for forward and inverse PDE problems with noisy data[J]. Journal of Computational Physics, 2021, 425: 109913.