An integration of weak solution with adversarial networks helps solving high-dimensional partial differential equations

Undergraduate Research Fellowship (URFP) 2021-2022 for Science Students
Poster No.: A6
UID: 3035534452
Major: Decision Analytics

Zhang Jiahao (supervised by Dr. Zhiwen Zhang)
Department of Mathematics, The University of Hong Kong

Abstract

Solving high-dimensional partial differential equations has always been a challenging problem in the field of scientific computing. There are many widely used traditional numerical methods for PDEs like finite difference method, finite element method, and spectral method. However, traditional numerical methods all face the problem of "Curse of dimensionality" when solving high-dimensional PDEs, where they suffer from slow computation, instability, and inaccuracy. In this research, we mainly investigate a group of newly proposed numerical methods based on deep neural networks that can overcome the "Curse of dimensionality". This group of methods for non-parametric PDEs includes Physical-informed Neural Networks (PINNs), Deep Ritz Method (DRM), Deep Galerkin Method (DGM), and Weak Adversarial Network (WAN). Among these deep learning based numerical methods, we make a thorough investigation on WAN which is based on the Generative Adversarial Network. And with a lightning-fast python code implementation, we successfully establish the Weak Adversarial Network and did a lot of numerical experiments on both linear high-dimensional PDEs and nonlinear ones. We also compare the performance of our weak adversarial network model with deep ritz method and physical-informed neural network and find that our model outperforms these two models in terms of running time. Apart from solving the high-dimensional PDEs, we also explore some improvements on the algorithms of solving min-max problem (i.e., the saddle point problem). The success of this research not only construct a complete python code for Weak Adversarial Network, but also accumulates essential data to advance the understanding of the performance and algorithms of weak adversarial network.

Introduction

Solving general high-dimensional partial differential equations (PDEs) has been a long-standing challenge in numerical analysis and computation. To instantiate the derivation of the weak adversarial network method, we first consider the following second-order elliptic PDE with either Dirichlet's or Neumann's boundary conditions on arbitrary domain , $\Omega \subset \mathbb{R}^d$

$$-\sum_{i=1}^{d} \partial_i\left(\sum_{j=1}^{d} a_{ij}\partial_j u\right) + \sum_{i=1}^{d} b_i \partial_i u + cu - f = 0 \qquad (*)$$

in Ω with the boundary condition of:

$$u(x) - g(x) = 0 \quad (Dirichlet) \quad or \quad \left(\frac{\partial u}{\partial \vec{n}}\right)(x) - g(x) = 0 \quad (Neumann), \quad on\ \partial\Omega$$

For the example like above equation, we will see that the method developed in this research can be directly applied to general high-dimensional PDEs, including both linear and nonlinear ones.

PDEs are prevalent and have extensive applications in science, engineering, economics, and finance. The most popular standard approaches to calculate numerical solutions of PDEs include finite difference and finite element methods (FEM). These methods discretize the time interval [0, T] and the domain Ω using mesh grids or triangulations, create simple basis functions on the mesh, convert a continuous PDE into its discrete counterpart, and finally solve the resulting system of basis coefficients to obtain numerical approximations of the true solution.

Although these methods have been significantly advanced in the past decades and are able to handle rather complicated and highly oscillating problems, they suffer the so-called "curse of dimensionality" since the number of mesh points increases exponentially fast with respect to the problem dimension d. Hence, they quickly become computationally intractable for high dimensional problem in practice. As a consequence, these numerical methods are rarely useful for general high-dimensional PDEs, e.g., d ≥ 4, especially when a sufficiently high-resolution solution is needed and/or the domain Ω is irregular.

Facing the challenge, our goal is to provide a computational feasible alternative approach to solve general high-dimensional PDEs defined on arbitrarily shaped domains. More specifically, using the weak formulation of PDEs, we parameterize the weak solution and the test function as the primal and adversarial neural networks respectively, and train them in an unsupervised form where only the evaluations of these networks (and their gradients) on some sampled collocation points in the interior and boundary of the domain are needed. Our approach retains the continuum nature of PDEs for which partial derivatives can be carried out directly without any spatial discretization and is fast and stable in solving general high-dimensional PDEs. Moreover, our method is completely mesh-free and can be applied to PDEs defined on arbitrarily shaped domains, without suffering the issue of the curse of dimensionality.

Methodology

To solve the equation in the form of *, we multiply both side with test function phi(x) and integration by part to get:

$$\begin{cases} \langle \mathcal{A}[u], \varphi \rangle \triangleq \int_\Omega \left(\sum_{j=1}^d \sum_{i=1}^d a_{ij}\partial_j u \partial_i \varphi + \sum_{i=1}^d b_i \varphi \partial_i u + cu\varphi - f\varphi \right) \mathrm{d}x = 0 \\ \mathcal{B}[u] = 0, \quad \text{on } \partial\Omega \end{cases}$$

Observe that the operator norm for A[u] is defined as:

$$\|\mathcal{A}[u]\|_{op} \triangleq \max\left\{ \langle \mathcal{A}[u], \varphi \rangle / \|\varphi\|_2 \mid \varphi \in H_0^1, \varphi \neq 0 \right\}$$

and we can further write as:

$$\min_{u \in H^1} \|\mathcal{A}[u]\|_{op}^2 \Longleftrightarrow \min_{u \in H^1} \max_{\varphi \in H_0^1} |\langle \mathcal{A}[u], \varphi \rangle|^2 / \|\varphi\|_2^2$$

Here we can clearly see that the problem of getting the weak solution of the equation has become a min-max problem (saddle point problem), thus we consider using Generative Adversarial Network to solve this problem. Note that the inside point loss function after taking log will be:

$$L_{\text{int}}(\theta, \eta) \triangleq \log |\langle \mathcal{A}[u_\theta], \varphi_\eta \rangle|^2 - \log \|\varphi_\eta\|_2^2$$

And the boundary point loss function will be:

$$L_{\text{bdry}}(\theta) \triangleq (1/N_b) \cdot \sum_{j=1}^{N_b} \left| u_\theta\left(x_b^{(j)}\right) - g\left(x_b^{(j)}\right) \right|^2$$

Thus, we can write the total loss function for our problem:

$$\min_\theta \max_\eta L(\theta, \eta), \quad \text{where } L(\theta, \eta) \triangleq L_{\text{int}}(\theta, \eta) + \alpha L_{\text{bdry}}(\theta),$$

where alpha is a parameter we can choose.

Finally, we can write out the algorithm for solving the high-dimensional PDEs according to the equation show above. And during the training, we update theta and phi alternatively, which is the same trick we use in the training of GAN.

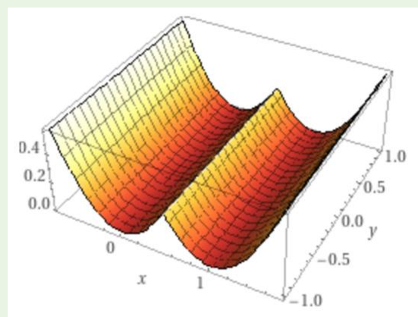
Experiments

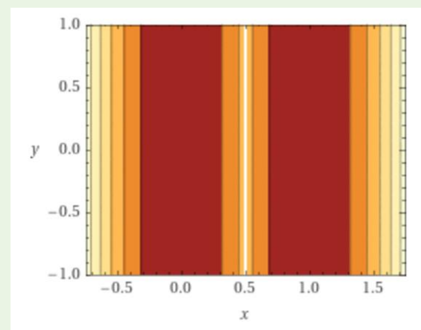
Consider the problem,

$$\begin{cases} -\nabla u = f \quad in \quad \Omega = [0,1] \times [0,1] \\ u = g \quad on \quad \partial\Omega \end{cases}$$

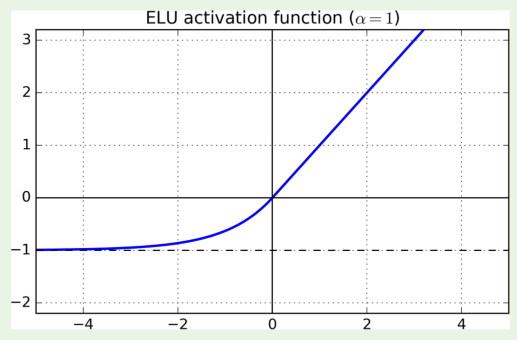
where we have

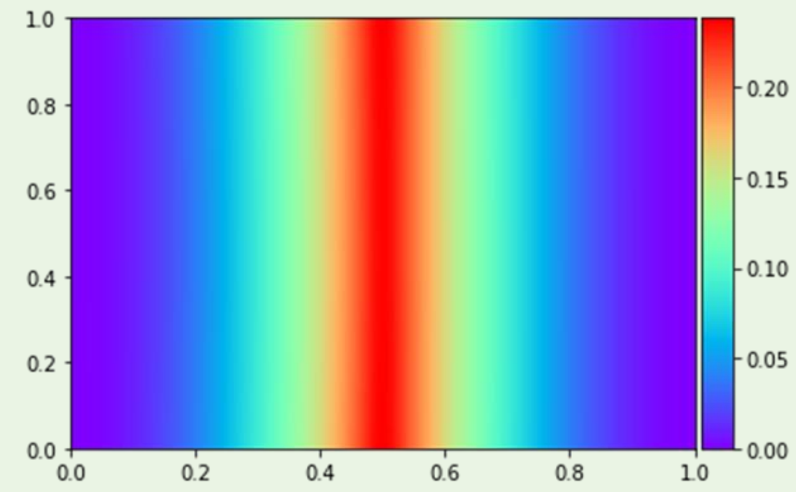
$$u(x, y) = min\{x^2, (1-x)^2\}$$



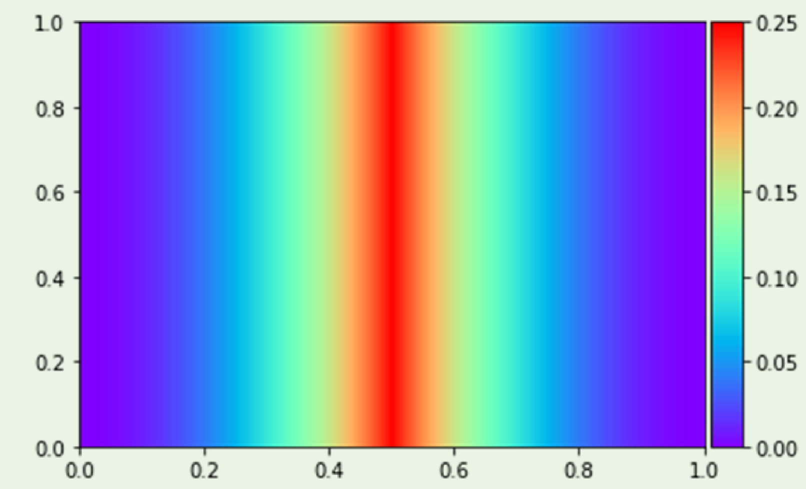


Activation function ELU

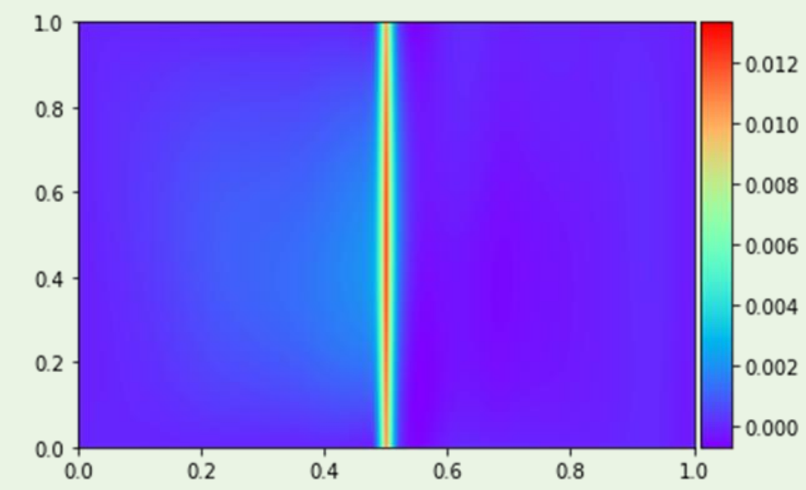




Exact Value



Predict Value



Exact - Predict

Conclusion

To conclude, in this research we have developed a complete python code to achieve the goal of combining weak solution with adversarial networks to solve high-dimensional partial differential equations. After observing the similarity of the loss function of the operator norm of the weak solution form and the loss function of the Generative Adversarial Network, we successfully use the training technique for GAN to get the solution to our target PDE. And we also further investigate some connections between the saddle point problem and the PDE solving problem. Some improvement on the saddle point problem algorithms can also advance our PDE solving model. And this need some future work to explore.

Reference List

Chen, J., Du, R., & Wu, K. (2020). A comparison study of deep Galerkin method and deep Ritz method for elliptic problems with different boundary conditions. arXiv preprint arXiv:2005.04554.
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.
Sirignano, J., & Spiliopoulos, K. (2018). DGM: A deep learning algorithm for solving partial differential equations. Journal of computational physics, 375, 1339-1364.
Yu, B. (2018). The deep Ritz method: a deep learning-based numerical algorithm for solving variational problems. Communications in Mathematics and Statistics, 6(1), 1-12.
Zang, Y., Bao, G., Ye, X., & Zhou, H. (2020). Weak adversarial networks for high-dimensional partial differential equations. Journal of Computational Physics, 411, 109409.