Numer. Math. Theor. Meth. Appl.
doi: 10.4208/nmtma.OA-2020-0062

Vol. **14**, No. 2, pp. 377-404
May 2021

Quasi-Monte Carlo Sampling for Solving Partial Differential Equations by Deep Neural Networks

Jingrun Chen[1], Rui Du[1,*], Panchi Li[2] and Liyao Lyu[3]

[1] *School of Mathematical Sciences and Mathematical Center for Interdisciplinary Research, Soochow University, Suzhou 215006, China*
[2] *School of Mathematical Sciences, Soochow University, Suzhou 215006, China*
[3] *School of Mathematical Sciences and CW Chu College, Soochow University, Suzhou 215006, China*

Received 14 April 2020; Accepted (in revised version) 7 September 2020

Abstract. Solving partial differential equations in high dimensions by deep neural networks has brought significant attentions in recent years. In many scenarios, the loss function is defined as an integral over a high-dimensional domain. Monte-Carlo method, together with a deep neural network, is used to overcome the curse of dimensionality, while classical methods fail. Often, a neural network outperforms classical numerical methods in terms of both accuracy and efficiency. In this paper, we propose to use quasi-Monte Carlo sampling, instead of Monte-Carlo method to approximate the loss function. To demonstrate the idea, we conduct numerical experiments in the framework of deep Ritz method. For the same accuracy requirement, it is observed that quasi-Monte Carlo sampling reduces the size of training data set by more than two orders of magnitude compared to that of Monte-Carlo method. Under some assumptions, we can prove that quasi-Monte Carlo sampling together with the deep neural network generates a convergent series with rate proportional to the approximation accuracy of quasi-Monte Carlo method for numerical integration. Numerically the fitted convergence rate is a bit smaller, but the proposed approach always outperforms Monte Carlo method.

AMS subject classifications: 11K50, 35J20, 65N99

Key words: Quasi-Monte Carlo sampling, deep Ritz method, loss function, convergence analysis.

1. Introduction

Deep neural networks (DNNs) have had great success in text classification, computer vision, natural language processing and other data-driven applications [13, 16,

[*]Corresponding author. *Email addresses:* jingrunchen@suda.edu.cn (J. Chen), durui@suda.edu.cn (R. Du), LiPanchi1994@163.com (P. Li), lylv@stu.suda.edu.cn (L. Lyu)

$21, 30, 35$]. Recently, DNNs have been applied to the field of numerical analysis and scientific computing, with the emphasis on solving high-dimensional partial differential equations (PDEs) [11,12,17,32], which are widely used in physics and finance. Notable examples include Schrödinger equation in the quantum many-body problem [15, 18], Hamilton-Jacobi-Bellman equation in stochastic optimal control [10,17], and nonlinear Black-Scholes equation for pricing financial derivatives [1, 7].

Classical numerical methods, such as finite difference method [23] and finite element method [3], share the similarity that the approximation stencil has compact support, resulting in the sparsity of stiffness matrix (or Hessian in the nonlinear case). Advantages of these methods are obvious for low dimensional PDEs (the dimension $K \leq 3$). However, the number of unknowns grows exponentially as K increases and classical methods run into the curse of dimensionality. In another line, spectral method [31] uses basis functions without compact support and thus sacrifices the sparsity, but often has the exponential accuracy. However, the number of modes used in the spectral method also grows exponentially as K increases. Sparse grid method [5,8,14] mitigates the aforementioned situation to some extent ($K \leq 9$ typically). Therefore, high-dimensional PDEs are far out of the capability of classical methods.

The popularity of DNNs in scientific computing results from its ability to approximate a high-dimensional function without the curse of dimensionality. To illustrate this, we focus on methods in which the loss function is defined as an integral over a bounded domain in high dimensions; see the deep Ritz method [11] and the deep Galerkin method [32] for examples. The success of DNNs relies on composition of functions without compact support and sampling strategy for approximating the high-dimensional integral. It is known that the choice of approximate functions in DNNs is of particular importance. For example, in the current work, the approximate function in one block of DNN consists of two linear transformations, two nonlinear activation functions, and one shortcut connection. Besides, since the network architecture is chosen a priori, the number of parameters can be independent of K or only grows linearly as K increases. On the other hand, only a fixed number of samples (or at most linear growth) is used to approximate the high-dimensional integral. Altogether, DNNs can overcome the curse of dimensionality when solving high-dimensional PDEs. In [2], the above step of numerical quadrature is viewed as approximating the expected risk by its empirical risk using Monte Carlo (MC) method. Consequently, the full gradient of the loss function is approximated by a finite number of samples and the stochastic gradient descent (SGD) method is used to find the optimal set of parameters in the network. It is shown that such a procedure converges under some assumptions.

From the perspective of numerical analysis, using N i.i.d. random points, MC method approximates an integral with $\mathcal{O}(N^{-\frac{1}{2}})$ error [24]. It is also known that using N carefully chosen (deterministic) points, quasi-Monte Carlo (QMC) method approximates an integral with $\mathcal{O}(\frac{(\log N)^K}{N^{-1}})$ error and the logarithmic factor can be removed under some assumptions [9, 27, 28, 34]. Therefore, it is natural to replace MC method by QMC method in the community of machine learning. One example is the usage of QMC method in variational inference and QMC method has been proved to perform better