INTERNATIONAL JOURNAL OF
NUMERICAL ANALYSIS AND MODELING
Volume 18, Number 4, Pages 427–441



# A DEEP LEARNING GALERKIN METHOD FOR THE SECOND-ORDER LINEAR ELLIPTIC EQUATIONS

JIAN LI*, WEN ZHANG, AND JING YUE

**Abstract.** In this paper we propose a Deep Learning Galerkin Method (DGM) based on the deep neural network learning algorithm to approximate the general second-order linear elliptic problem. This method is a combination of Galerkin Method and machine learning. The DGM uses the deep neural network instead of the linear combination of basis functions. Our algorithm is meshfree and we train the neural network by randomly sampling the space points and using the gradient descent algorithm to satisfy the differential operators and boundary conditions. Moreover, the approximate ability of a neural networks' solution to the exact solution is proved by the convergence of the loss function and the convergence of the neural network to the exact solution in $L^2$ norm under certain conditions. Finally, some numerical experiments reflect the approximation ability of the neural networks intuitively.

**Key words.** Deep learning Galerkin method, deep neural network, second-order linear elliptic equations, convergence, numerical experiments.

## 1. Introduction

Mathematical models in many fields can be described by partial differential equations (PDEs). As early as the establishment of the calculus theory, PDEs were used to describe various natural phenomena and were applied to various scientific or engineering technologies. The high-dimensional partial differential equations are applied to physics, engineering, aerospace and other aspects. The well-known examples include the Schrödinger equation in quantum many-body problem, the nonlinear Black-Scholes equation for pricing financial derivatives, the Hamilton-Jacobi-Bellman equation in dynamic programming and so on. Unfortunately, the solutions of most PDEs cannot be expressed in the form of analytical solutions, so their numerical solutions are particularly important. Though many numerical methods have been developed so far for solving PDEs, such as finite difference method, finite element method and finite volume method, these methods still have certain limitations. As for the higher dimensional problems, the computational cost of the surge in grid points goes up exponentially with the dimensionality. As a result, solving numerical solutions has been a longstanding challenge.

With the explosive growth of available data and computing resources, the deep neural network model has shown remarkable success in artificial intelligence[1, 2, 3, 4, 5]. Recently, [6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17] have proposed that it can use the neural networks to solve PDEs. The deep neural networks with many layers seem to do a surprisingly good job in modeling complicated datasets. Besides, many effective algorithms are proposed to solve some high-dimensional PDEs in [18, 27, 28, 30, 31, 33, 34, 35, 36, 37]. In these papers, most of them only give numerical computations and illustrate the validity of numerical solutions through

Received by the editors April 1, 2020 and, in revised form, March 15, 2021.
2000 *Mathematics Subject Classification.* 35J15, 65N12, 74S30.
*Corresponding author.

various pictures. But there are no strict proofs about the existence and uniqueness of the exact solutions as well as the convergence of the error between the numerical solutions and the exact solutions. Here, based on the framework of [6], we directly apply the DGM for solving the second-order PDEs without using Monte Carlo Method. This method is the merger of the Galerkin Method and machine learning, which is different from the traditional Galerkin Method. The DGM uses the deep neural network instead of the linear combination of basis functions. We train the neural network by randomly sampling the space points and using the gradient descent algorithm to satisfy the differential operators and boundary conditions. We don't need to form a grid in this process. This is also an important reason why the DGM can solve the high-dimensional PDEs. Obviously, the method presented is much simpler but more effective. Moreover, we obtain the convergence of the loss function and the neural network, respectively. Finally, the performance of the method is demonstrated by some numerical experiments.

The rest of paper is organized as follows. In the next section, we will introduce a method that solves high dimensional PDEs with meshfree deep learning algorithm. In section 3, the theorem to illustrate the convergence of the loss function is proved. We also give the proof of another theorem to guarantee the convergence of neural network's solution in section 4. Finally, a series of numerical experiments are given in section 5.

## 2. Methodology

Let $\Omega \subset \mathbb{R}^d$, $(d = 2, 3)$ be a bounded set with a sufficiently smooth boundary $\partial\Omega$. We consider a class of the second-order linear elliptic equations in combination with Dirichlet boundary conditions:

$$\alpha u(x) - \Delta u(x) = f(x), \quad in\ \Omega, \tag{1}$$

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

where $\alpha > 0$ is a positive constant.

In the following, recall the classical Sobolev spaces:

$$H^k(\Omega) = \Big\{\nu | \nu \in L^2(\Omega), D_w^k(\nu) \in L^2(\Omega), \forall \alpha : |\alpha| \le k\Big\},$$
$$H_0^k(\Omega) = \Big\{\nu \in H^k(\Omega) : \nu|_{\partial\Omega} = 0\Big\}.$$

Especially, $L^2(\Omega) = \Big\{\nu(x) | \Big(\int_\Omega |\nu(x)|^2 dx\Big)^{\frac{1}{2}} < \infty\Big\}$ is sometimes defined by $H^0(\Omega)$. Here, $D_w^k(\nu)$ is the generalized $k$-order derivative of $\nu$ and its classical norm is equipped with the norm $\|\nu\|_k$.

For simplicity of notations, let us denote

$$\mathcal{G}[u](x) = \alpha u(x) - \Delta u(x) - f(x).$$

By the Lax-Milgram theorem, the second-order linear elliptic problem (1)-(2) has a unique solution

$$u \in H^2(\Omega) \text{ such that } \|u\|_2 \le C(\|f\|_0 + \|g\|_{3/2,\partial\Omega}), \tag{3}$$

where $H^{3/2}(\partial\Omega)$ is equipped with the trace norm

$$\|\phi\|_{3/2,\partial\Omega} = \inf\big\{\|u\|_2 \big| u \in H^2(\Omega)\ and\ u\big|_{\partial\Omega} = \phi\big\}.$$