Commun. Comput. Phys.
doi: 10.4208/cicp.OA-2020-0193

Vol. **28**, No. 5, pp. 2042-2074
November 2020

On the Convergence of Physics Informed Neural Networks for Linear Second-Order Elliptic and Parabolic Type PDEs

Yeonjong Shin[1,*], Jérôme Darbon[1] and George Em Karniadakis[1]

[1] *Division of Applied Mathematics, Brown University, Providence, RI 02912, USA.*

Received 29 September 2020; Accepted (in revised version) 16 October 2020

Abstract. Physics informed neural networks (PINNs) are deep learning based techniques for solving partial differential equations (PDEs) encountered in computational science and engineering. Guided by data and physical laws, PINNs find a neural network that approximates the solution to a system of PDEs. Such a neural network is obtained by minimizing a loss function in which any prior knowledge of PDEs and data are encoded. Despite its remarkable empirical success in one, two or three dimensional problems, there is little theoretical justification for PINNs.
As the number of data grows, PINNs generate a sequence of minimizers which correspond to a sequence of neural networks. We want to answer the question: Does the sequence of minimizers converge to the solution to the PDE? We consider two classes of PDEs: linear second-order elliptic and parabolic. By adapting the Schauder approach and the maximum principle, we show that the sequence of minimizers strongly converges to the PDE solution in C^0. Furthermore, we show that if each minimizer satisfies the initial/boundary conditions, the convergence mode becomes H^1. Computational examples are provided to illustrate our theoretical findings. To the best of our knowledge, this is the first theoretical work that shows the consistency of PINNs.

AMS subject classifications: 65M12, 41A46, 35J25, 35K20

Key words: Physics informed neural networks, convergence, Hölder regularization, elliptic and parabolic PDEs, Schauder approach.

1 Introduction

Machine learning techniques using deep neural networks have been successfully applied in various fields [24] such as computer vision and natural language processing. A notable advantage of using neural networks is its efficient implementation using a dedicated

*Corresponding author. *Email addresses:* yeonjong_shin@brown.edu (Y. Shin), jerome_darbon@brown.edu (J. Darbon), george_karniadakis@brown.edu (G. E. Karniadakis)

hardware (see [8, 22]). Such techniques have also been applied in solving partial differential equations (PDEs) [4, 10, 22, 23, 32, 35], and it has become a new sub-field under the name of Scientific Machine Learning (SciML) [2, 26]. The term Physics-Informed Neural Networks (PINNs) was introduced in [32] and it has become one of the most popular deep learning methods in SciML. PINNs employ a neural network as a solution surrogate and seek to find the best neural network guided by data and physical laws expressed as PDEs.

A series of works have shown the effectiveness of PINNs in one, two or three dimensional problems: fractional PDEs [30, 36], stochastic differential equations [18, 39], biomedical problems [33], and fluid mechanics [27]. Despite such remarkable success in these and related areas, PINNs lack theoretical justification. In this paper, we provide a mathematical justification of PINNs.

One of the main goals of PINNs is to approximate the solution to the PDE. For readers' convenience, we provide a concrete example to explain the PINNs method. Let us consider a 1D Poisson equation on $U=(0,1)$ with the Dirichlet boundary conditions:

$$\mathcal{L}[u](x)=f(x), \quad \forall x\in U, \qquad \mathcal{B}[u](x)=g(x), \quad \forall x\in \partial U,$$

where the differential operator $\mathcal{L}$ is the Laplace Δ operator and the boundary operator $\mathcal{B}$ is the identity operator. Suppose that the differential operator $\mathcal{L}$ and the boundary operator $\mathcal{B}$ are known to us, however, the PDE data (i.e., f and g) are only known at some sample points. In other words, although we do not know what f and g are exactly, we can access them through their pointwise evaluations. Here and throughout the paper, we assume the high regularity setting for PDEs where point-wise evaluations are defined. We refer to a set of available pointwise values of f and g as the training data set. The loss functionals are then designed to penalize functions that fail to satisfy both governing equations (PDEs) and boundary conditions *on the training data*. For example, if $(j/5, f(j/5))$ for $j=1,\cdots,4$, and $(0, g(0)$ and $(1, g(1))$ are given to us as the training data, a prototype PINN loss functional [32] is defined by

$$\text{Loss}(u)=\frac{1}{4}\sum_{j=1}^{4}(\Delta u(j/4)-f(j/4))^2+\frac{1}{2}\left[(u(0)-g(0))^2+(u(1)-g(1))^2\right].$$

PINNs seek to find a neural network that minimizes the loss in a class of neural networks. A minimizer serves as an approximation to the solution to the PDE. Since neural networks are parameterized with finitely many variables, the loss functional restricted to it becomes a function of network parameters, which is called the loss function. With a slight abuse of terminology, we shall not distinguish the loss functional and the loss function and both will be simply called the loss function. In Fig. 1, we provide a schematic of PINNs.

PINNs are different approaches to the traditional variational principle that minimizes an energy functional [1, 6, 11]. The most distinctive difference between them is that not all PDEs satisfy a variational principle, however, the formulation of PINNs does not require