INTERNATIONAL JOURNAL OF
NUMERICAL ANALYSIS AND MODELING
Volume 21, Number 5, Pages 609–628

doi: 10.4208/ijnam2024-1024

LEAST-SQUARES NEURAL NETWORK (LSNN) METHOD FOR LINEAR ADVECTION-REACTION EQUATION: NON-CONSTANT JUMPS

ZHIQIANG CAI, JUNPYO CHOI*, AND MIN LIU

Abstract. The least-squares ReLU neural network (LSNN) method was introduced and studied for solving linear advection-reaction equation with discontinuous solution in [4, 5]. The method is based on an equivalent least-squares formulation and [5] employs ReLU neural network (NN) functions with $\lceil \log_2(d+1) \rceil + 1$-layer representations for approximating solutions. In this paper, we show theoretically that the method is also capable of accurately approximating non-constant jumps along discontinuous interfaces that are not necessarily straight lines. Theoretical results are confirmed through multiple numerical examples with $d = 2, 3$ and various non-constant jumps and interface shapes, showing that the LSNN method with $\lceil \log_2(d+1) \rceil + 1$ layers approximates solutions accurately with degrees of freedom less than that of mesh-based methods and without the common Gibbs phenomena along discontinuous interfaces having non-constant jumps.

Key words. Least-squares method, ReLU neural network, linear advection-reaction equation, discontinuous solution.

1. Introduction

For decades, extensive research has been conducted on numerical methods for linear advection-reaction equations to develop precise and efficient numerical schemes. A major challenge in numerical simulation is that the solution of the equation is discontinuous along an interface because of a discontinuous inflow boundary condition, where the discontinuous interface can be the streamline from the inflow boundary. Traditional mesh-based numerical methods often exhibit oscillations near the discontinuity (called the Gibbs phenomena), which are not suitable for many applications and may not be extended to nonlinear hyperbolic conservation laws.

Recently, the application of neural networks (NNs) for solving partial differential equations have achieved significant accomplishments. For the linear advection-reaction problem, the least-squares ReLU neural network (LSNN) method was introduced and studied in [4, 5]. The method is based on an equivalent least-squares formulation studied in [2, 6] and [5] employs ReLU neural network (NN) functions with $\lceil \log_2(d+1) \rceil + 1$-layer representations for approximating the solution. The LSNN method is capable of automatically approximating the discontinuous solution accurately since the free hyperplanes of ReLU NN functions adapt to the solution (see [3, 4, 5]). Moreover, for problems with unknown locations of discontinuity interfaces, it is quite easy to see that the LSNN method uses much fewer number of degrees of freedom than the mesh-based methods (see, e.g., [3, 4]).

Approximation properties of ReLU NN functions to step functions were examined and employed in [4, 5]. In particular, it was shown theoretically that two- or $\lceil \log_2(d+1) \rceil + 1$-layer ReLU NN functions are necessary and sufficient to approximate a step function with any given accuracy $\varepsilon > 0$ when the discontinuous interface is a hyperplane or general hyper-surface, respectively. This approximation

Received by the editors on May 30, 2023 and, accepted on December 15, 2023.
2000 *Mathematics Subject Classification.* 65N15, 65N99.
*Corresponding author.

property was used to establish *a priori* error estimates of the LSNN method for the linear advection-reaction problem.

The jump of the discontinuous solution of the problem, however, is generally non-constant when the reaction coefficient is non-zero. The main purpose of this paper is to establish *a priori* error estimates (see Theorem 3.2) for the LSNN method without making the assumption that the jump is constant as in [5]. To this end, we decompose the solution as the sum of discontinuous and continuous parts (see (10)), so that the discontinuous part of the solution can be described as a cylindrical surface on one subdomain and zero otherwise. Then we construct a continuous piecewise linear (CPWL) function with a sharp transition layer along the discontinuous interface to approximate the discontinuous piecewise cylindrical surface accurately. From [10, 11, 1, 5], we know that the CPWL function is a ReLU NN function $\mathbb{R}^d \to \mathbb{R}$ with a $\lceil \log_2(d+1) \rceil + 1$-layer representation, from which it follows that the discontinuous part of the solution can be approximated by this class of functions for any prescribed accuracy. Then Theorem 3.2 follows.

The rest of the paper is organized as follows. In Section 2, we introduce the linear advection-reaction problem, and briefly review and discuss properties of ReLU NN functions and the LSNN method in [5]. Then theoretical convergence analysis is conducted in Section 3, showing that discretization error of the method for the problem mainly depends on the continuous part of the solution. Finally, to demonstrate the effectiveness of the method, we provide numerical results for test problems with various non-constant jumps in Section 4. Section 5 summarizes the work.

2. Problem formulation and the LSNN method

Let Ω be a bounded domain in $\mathbb{R}^d$ $(d \geq 2)$ with Lipschitz boundary $\partial\Omega$, and denote the advective velocity field by $\boldsymbol{\beta}(\mathbf{x}) = (\beta_1, \cdots, \beta_d)^T \in C^0(\bar{\Omega})^d$. Define the inflow part of the boundary $\Gamma = \partial\Omega$ by

$$\Gamma_- = \{\mathbf{x} \in \Gamma : \boldsymbol{\beta}(\mathbf{x}) \cdot \boldsymbol{n}(\mathbf{x}) < 0\} \tag{1}$$

where $\boldsymbol{n}(\mathbf{x})$ is the unit outward normal vector to Γ at $\mathbf{x} \in \Gamma$. Consider the linear advection-reaction equation

$$\left\{ \begin{array}{rcll} u_{\boldsymbol{\beta}} + \gamma\, u & = & f, & \text{in } \Omega, \\ u & = & g, & \text{on } \Gamma_-, \end{array} \right. \tag{2}$$

where $u_{\boldsymbol{\beta}}$ denotes the directional derivative of u along $\boldsymbol{\beta}$. Assume that $\gamma \in C^0(\bar{\Omega})$, $f \in L^2(\Omega)$, and $g \in L^2(\Gamma_-)$ are given scalar-valued functions.

For the convenience of the reader, this section briefly reviews properties of ReLU neural network (NN) functions and the least-squares ReLU neural network (LSNN) method in [5]. A function $\mathcal{N} : \mathbb{R}^d \to \mathbb{R}^c$ is called a ReLU neural network (NN) function if it can be expressed as a composition of functions

$$N^{(L)} \circ \cdots \circ N^{(2)} \circ N^{(1)} \text{ with } L > 1, \tag{3}$$

where $N^{(l)} : \mathbb{R}^{n_{l-1}} \to \mathbb{R}^{n_l}$ $(n_0 = d,\ n_L = c)$ is affine linear when $l = L$, and affine linear with the rectified linear unit (ReLU) activation function σ applied to each component when $1 \leq l \leq L-1$. Each affine linear function takes the form $\boldsymbol{\omega}^{(l)}\mathbf{x} - \mathbf{b}^{(l)}$ for $\mathbf{x} \in \mathbb{R}^{n_{l-1}}$ where $\boldsymbol{\omega}^{(l)} \in \mathbb{R}^{n_l \times n_{l-1}}$, $\mathbf{b}^{(l)} \in \mathbb{R}^{n_l}$ are weight and bias matrices, respectively. For $n \in \mathbb{N}$, denote the collection of all ReLU NN functions from $\mathbb{R}^d$ to $\mathbb{R}$ with depth L and the number of hidden neurons $n(= n_1 + \cdots + n_{L-1})$