Journal of Computational Mathematics
Vol.38, No.3, 2020, 502–527.

http://www.global-sci.org/jcm
doi:10.4208/jcm.1901-m2018-0160

RELU DEEP NEURAL NETWORKS AND LINEAR FINITE ELEMENTS*

Juncai He
School of Mathematical Sciences, Peking University, Beijing 100871, China
Email: juncaihe@pku.edu.cn
Lin Li
Beijing International Center for Mathematical Research, Peking University, Beijing 100871, China
Email: lilin1993@pku.edu.cn
Jinchao Xu and Chunyue Zheng
Department of Mathematics, Pennsylvania State University, State College, PA 16802, USA
Email: xu@math.psu.edu, cmz5199@psu.edu

Abstract

In this paper, we investigate the relationship between deep neural networks (DNN) with rectified linear unit (ReLU) function as the activation function and continuous piecewise linear (CPWL) functions, especially CPWL functions from the simplicial linear finite element method (FEM). We first consider the special case of FEM. By exploring the DNN representation of its nodal basis functions, we present a ReLU DNN representation of CPWL in FEM. We theoretically establish that at least 2 hidden layers are needed in a ReLU DNN to represent any linear finite element functions in $\Omega \subseteq \mathbb{R}^d$ when $d \geq 2$. Consequently, for $d = 2, 3$ which are often encountered in scientific and engineering computing, the minimal number of two hidden layers are necessary and sufficient for any CPWL function to be represented by a ReLU DNN. Then we include a detailed account on how a general CPWL in $\mathbb{R}^d$ can be represented by a ReLU DNN with at most $\lceil \log_2(d+1) \rceil$ hidden layers and we also give an estimation of the number of neurons in DNN that are needed in such a representation. Furthermore, using the relationship between DNN and FEM, we theoretically argue that a special class of DNN models with low bit-width are still expected to have an adequate representation power in applications. Finally, as a proof of concept, we present some numerical results for using ReLU DNNs to solve a two point boundary problem to demonstrate the potential of applying DNN for numerical solution of partial differential equations.

Mathematics subject classification: 26B40, 65N30, 65N99.
Key words: Finite element method, Deep neural network, Piecewise linear function.

1. Introduction

In recent years, deep learning models have achieved unprecedented success in various tasks of machine learning or artificial intelligence, such as computer vision, natural language processing and reinforcement learning [1]. One main technique in deep learning is deep neural network. A typical DNN model is based on a hierarchy of composition of linear functions and a given nonlinear activation function. However, why DNN models can work so well is still unclear.

* Received July 25, 2018 / Revised version received October 9, 2018 / Accepted January 17, 2019 / Published online May 27, 2019 /

Mathematical analysis of DNN can be carried out using many different approaches. One approach is to study the approximation properties of the function class provided by DNN. The approximation property of DNN is relevant to the so-called expressive power [2] of a DNN model. Early studies of approximation properties of DNN can be traced back in [3] and [4] where the authors established some approximation properties for the function classes given by a feedforward neural network with a single hidden layer. Further error estimates for such neural networks in terms of number of neurons can be found in [5] for sinusoidal activation functions and in [6] for more general sigmoidal activation functions. There are many other papers on this topic during the 90s and a good review of relevant works can be found in [7] and [8].

There are many different choices of activation functions. In fact, as shown in [9], a neural network with a single hidden layer can approximate any continuous function for any activation function which is not a polynomial. Among all the activation functions, the so-called rectified linear unit (ReLU) activation function [10], namely $\mathrm{ReLU}(x) = \max(x, 0)$, has emerged to be one of the most popular activation functions used in the deep learning literature and applications. [11] presents an approximation of ReLU DNNs by relating to wavelets. Recently, [12] establish L^∞ and L^2 error bounds for functions of many variables that are approximated by linear combinations of ReLU. [13] presents rates of approximation by deep CNNs for functions in the Sobolev space $H^r(\Omega)$ with $r > 2 + d/2$. This paper is devoted to some further mathematical analysis of DNN models with ReLU as the activation function.

It is not difficult to see that the following statement is true: "*Every ReLU DNN function in $\mathbb{R}^d$ represents a continuous piecewise linear (CPWL) function defined on a number of polyhedral subdomains.*" One important recent development is that the converse of the above statement has also been proven true. More specifically, the following result is established by [14] based on an earlier result by [15] on lattice representation of DNN: "*Every CPWL function in $\mathbb{R}^d$ can be represented by a ReLU DNN model with at most $\lceil \log_2(d+1) \rceil$ hidden layers.*"

Motivated by this result, we study the following two questions on the DNN representation of a given CWPL function:

1. How many numbers of neurons are needed?

2. What is the minimal number of layers that are needed?

To answer the first question, in this paper, we will go through the proof of this representation result to give some explicit estimations of the number of neurons that are needed in a DNN to represent a given CPWL function. As a result, we find that the number of neurons that are needed for a DNN to represent a CPWL on m-subdomains can be as large as $\mathcal{O}(d2^{mm!})$!

In order to obtain DNN representation with fewer numbers of neurons, in this paper, we consider a special class of CPWL functions, namely the linear finite element (LFE) functions [16] defined on a collection of special subdomains, namely simplexes in $\mathbb{R}^d$. As every finite element function can be written as a linear combination of nodal basis functions, it suffices to study DNN representation of any given nodal basis function. To represent a nodal basis function by a DNN, we do not need to consider the complicated domain partition related with lattice representation, which is important in representing general piecewise linear functions in $\mathbb{R}^d$ (see [14, 15]). We prove that a linear finite element function with N degrees of freedom can be represented by a ReLU DNN with at most $\mathcal{O}(d\kappa^d N)$ number of neurons with $\mathcal{O}(d)$ hidden layers where $\kappa \geq 2$) depends on the shape regularity of the underlying finite element grid.

To answer the second question, we will again consider the linear finite element functions. In this paper, we will show (see Theorem 4.1) that at least 2 hidden layers are needed for a ReLU