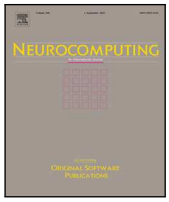
Neurocomputing 571 (2024) 127174

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom



Convergence of deep ReLU networks

Yuesheng Xu [a,1], Haizhang Zhang [b,*,2]

[a] Department of Mathematics & Statistics, Old Dominion University, Norfolk, VA 23529, USA
[b] School of Mathematics (Zhuhai), Sun Yat-sen University, Zhuhai, PR China

ARTICLE INFO

Communicated by H.R. Karimi

Keywords:
Deep learning
ReLU networks
Activation domains
Infinite product of matrices

ABSTRACT

We explore convergence of deep neural networks with the popular ReLU activation function, as the depth of the networks tends to infinity. To this end, we introduce the notion of activation domains and activation matrices of a ReLU network. By replacing applications of the ReLU activation function by multiplications with activation matrices on activation domains, we obtain an explicit expression of the ReLU network. We then identify the convergence of the ReLU networks as convergence of a class of infinite products of matrices. Sufficient and necessary conditions for convergence of these infinite products of matrices are studied. As a result, we establish necessary conditions for ReLU networks to converge that the sequence of weight matrices converges to the identity matrix and the sequence of the bias vectors converges to zero as the depth of ReLU networks increases to infinity. Moreover, we obtain sufficient conditions in terms of the weight matrices and bias vectors at hidden layers for pointwise convergence of deep ReLU networks. These results provide mathematical insights to convergence of deep neural networks. Experiments are conducted to mathematically verify the results and to illustrate their potential usefulness in initialization of deep neural networks.

1. Introduction

Deep neural networks have achieved great successes for a wide range of machine learning problems including face recognition, speech recognition, game intelligence, natural language processing, and autonomous navigation. It is generally agreed that four ingredients contribute to the successes. The first two of them are the availability of vast amounts of training data, and recent dramatic improvements in computing and storage power. The third one is a class of efficient numerical algorithms such as the Stochastic Gradient Decent (SGD) algorithms, Adaptive Boosting (AdaBoost) algorithms, and the Expectation–Maximization algorithm (EM). The fourth ingredient, also the most important one, is powerful neural network architectures, such as Convolutional Neural Networks (CNN), Long-Short Time Memory (LSTM) networks, Recurrent Neural Networks (RNN), Generative Adversarial Networks (GAN), Deep Belief Networks (DBN), and Residual Networks (ResNet), which provide a superior way of representing data. We refer to a survey paper [1] and monograph [2] for an in-depth overview of deep learning.

Compared to the vast development in engineering and applications, research on the mathematical theory of deep neural networks is still at its infancy, and yet is undergoing rapid progress. Many interesting papers on the approximation and expressive powers of deep neural networks have appeared in the past several years. We provide here a brief review. More details can be found in two recent surveys [3,4]. Poggio, Mhaskar, Rosasco, Miranda, and Liao [5] proved that deep neural networks approximate a class of functions with special compositional structure exponentially better than shallow networks. Montanelli and Du [6] and Yarotsky [7] estimated the number of parameters needed for deep neural networks to achieve a certain error tolerance in approximating functions in the Koborov space and differential functions, respectively. Montanelli and Yang [8] achieved error bounds for deep ReLU networks approximation of multivariate functions using the Kolmogorov–Arnold superposition theorem. These three pieces of work indicated that deep neural networks are able to lessen the curse of dimensionality. E and Wang [9] proved that for analytic functions in a low dimension, the convergence rate of the deep neural network approximation is exponential. Zhou [10] established the universality of deep convolutional neural networks. Shen, Yang, and Zhang put forward in a series of works [11–13] optimal approximation rates for ReLU networks in terms of width and depth to approximate an arbitrary continuous or Hölder continuous function. Daubechies, DeVore, Foucart, Hanin, and Petrova [14] showed that deep neural networks possess

* Corresponding author.
E-mail addresses: y1xu@odu.edu (Y. Xu), zhhaizh2@sysu.edu.cn (H. Zhang).

[1] Supported in part by US National Science Foundation under grants DMS-1912958 and DMS-2208386, and by the US National Institutes of Health under grant R21CA263876.

[2] Supported in part by National Natural Science Foundation of China under grants 12371103, 11971490 and 12126610.

https://doi.org/10.1016/j.neucom.2023.127174
Received 10 January 2023; Received in revised form 1 October 2023; Accepted 26 December 2023
Available online 28 December 2023

greater approximation power than traditional methods of nonlinear approximation such as variable knot splines and n-term approximation from dictionaries. Wang [15] presented a mathematical introduction to generative adversarial nets. Lipschitz and proximal properties of neural networks were investigated in [16–19]. Investigations on theoretical properties of deep neural networks via the neural tangent kernel were conducted in [20–22].

In this paper, we study convergence of deep ReLU networks from a different perspective. We are interested in knowing whether or not a deep ReLU network with a fixed width and an increasing depth will converge to a meaningful function (as a function of the input variable), as its depth tends to infinity. It is well-known that in linear approximations (for example, Fourier analysis [23] and wavelet analysis [24]), issues regarding convergence of an expansion such as Fourier expansion and wavelet expansion are fundamental. In particular, in classical analysis, convergence of Fourier expansions with given coefficients is a basic issue. As deep neural networks are used more and more in approximation as a function class, convergence of a sequence of neural networks approximating to a function has become a pressing and interesting issue. Along this line, the first question is: What requirements should we impose to the weight matrices and the bias vectors to guarantee that the related ReLU deep neural network will converge to a meaningful function as its number of layers tends to infinity? This paper attempts to answer this question. The neural networks to be considered here are not tied with a specific target function. Convergence of neural networks that result from approximation of a given function will be investigated in a different occasion.

It has long been understood that a neural network with the ReLU activation function results in a piecewise linear function. The first novelty of this work is to identify a subdomain that corresponds to a linear component of the ReLU network as an *activation domain* and to an *activation matrix* which is a diagonal matrix whose diagonal entries are either 1 or 0. The identification allows us to replace the applications of the ReLU activation function, often a source of technical difficulties, by multiplications with the activation matrices. Making use of this observation, we put forward a useful representation for deep ReLU networks, by which we formulate the convergence of deep ReLU networks as convergence of a class of infinite products of matrices. Necessary and sufficient conditions for convergence of such infinite products of matrices are then established. Based on this understanding, we provide necessary conditions and rather week sufficient conditions for a deep ReLU network to converge. The necessary conditions supply mathematical guidelines for further development of deep ReLU networks. Moreover, the sufficient conditions enable us to interpret the design strategy of the well-known deep residual networks, which have been widely used in image classification, with an insightful mathematical explanation.

The rest of this paper is organized as follows. In Section 2, we review the definition and notation of neural networks and define the notion of convergence of neural networks when new layers are paved to the existing network so that the depth is increasing to infinity. In Section 3, we introduce the notions of the activation domain and activation matrix, with which we present an explicit expression for deep ReLU networks. Based on this expression, we connect convergence of deep ReLU networks with the existence of two limits involving infinite products of matrices. Conditions for convergence of such infinite products of matrices are examined in Section 4. Finally, in Section 5 we revisit the convergence of deep ReLU networks by presenting sufficient conditions for the pointwise convergence of the deep ReLU networks. Moreover, as an application of the result established, we provide a mathematical interpretation to the design of the successful deep residual networks.

2. Deep neural networks and convergence

In this section, we recall the definition of the deep neural network and formulate its convergence problem to be studied in this paper.

We consider general fully connected feed-forward neural networks with fixed width m and increasing depth n, for $m, n \in \mathbb{N}$, from input domain $[0,1]^d \subseteq \mathbb{R}^d$ to the output space $\mathbb{R}^{d'}$. For each i with $1 \le i \le n$, let $\mathbf{W}_i$ and $\mathbf{b}_i$ denote respectively the weight matrix and bias vector of the ith hidden layer. That is, $\mathbf{b}_i \in \mathbb{R}^m$ for $1 \le i \le n$, $\mathbf{W}_1 \in \mathbb{R}^{m\times d}$, and $\mathbf{W}_i \in \mathbb{R}^{m\times m}$ for $2 \le i \le n$. The weight matrix $\mathbf{W}_o$ and bias vector $\mathbf{b}_o$ of the output layer satisfy $\mathbf{W}_o \in \mathbb{R}^{d'\times m}$ and $\mathbf{b}_o \in \mathbb{R}^{d'}$. The structure of such a deep neural network is determined after the choice of an activation function.

Widely-used activation functions in neural networks include the **ReLU** function

$$\mathrm{ReLU}(x) := \max(x, 0), \quad x \in \mathbb{R}$$

and the logistic sigmoid function

$$S(x) := \frac{1}{1+e^{-x}}, \quad x \in \mathbb{R}.$$

After an activation function σ is chosen, the structure of the resulting deep neural network may be illustrated as follows:

$$\underset{\text{input}}{x \in [0,1]^d} \xrightarrow[\sigma]{\mathbf{W}_1,\mathbf{b}_1} \underset{\text{1st layer}}{x^{(1)}} \xrightarrow[\sigma]{\mathbf{W}_2,\mathbf{b}_2} \underset{\text{2nd layer}}{x^{(2)}} \to \cdots \to \xrightarrow[\sigma]{\mathbf{W}_n,\mathbf{b}_n} \underset{n\text{th layer}}{x^{(n)}} \xrightarrow{\mathbf{W}_o,\mathbf{b}_o} \underset{\text{output}}{y \in \mathbb{R}^{d'}}. \tag{2.1}$$

Here

$$x^{(k)} := \sigma(\mathbf{W}_k x^{(k-1)} + \mathbf{b}_k), \quad 1 \le k \le n \ \text{ with } \ x^{(0)} = x, \tag{2.2}$$

$$y := \mathbf{W}_o x^{(n)} + b_o, \tag{2.3}$$

and the activation function σ is applied to a vector componentwise. Thus, the above deep neural network determines a continuous function $x \to y$ from $[0,1]^d$ to $\mathbb{R}^{d'}$.

Consecutive compositions of functions are typical operations used in deep neural networks. To have a compact form, below we define the notation for consecutive compositions of functions.

Definition 2.1 (*Consecutive Composition*)**.** Let $f_1, f_2, \dots, f_n$ be a finite sequence of functions such that the range of f_i is contained in the domain of f_{i+1}, $1 \le i \le n-1$, the consecutive composition of $\{f_i\}_{i=1}^n$ is defined to be function

$$\bigodot_{i=1}^{n} f_i := f_n \circ f_{n-1} \circ \cdots \circ f_2 \circ f_1,$$

whose domain is that of f_1.

Note that whenever the consecutive composition notation is used, the order of compositions given in Definition 2.1 is always assumed.

Using the notation defined in Definition 2.1 for consecutive compositions of functions, Eqs. (2.2) and (2.3) may be rewritten as

$$x^{(k)} = \left(\bigodot_{i=1}^{k} \sigma(\mathbf{W}_i \cdot + \mathbf{b}_i)\right)(x), \quad 1 \le k \le n$$

and

$$y = \mathbf{W}_o \left(\bigodot_{i=1}^{n} \sigma(\mathbf{W}_i \cdot + \mathbf{b}_i)\right)(x) + b_o, \quad x \in [0,1]^d,$$

respectively. We are concerned with convergence of the above functions determined by the deep neural network as n increases to infinity. One sees that the output layer is a linear function of $x^{(n)}$ and thus, it does not affect the convergence. By this observation, we introduce the following definition.

Definition 2.2 (*Convergence of Neural Networks*)**.** Let $\mathbf{W} := \{\mathbf{W}_n\}_{n=1}^{\infty}$ with $\mathbf{W}_1 \in \mathbb{R}^{m\times d}$, $\mathbf{W}_n \in \mathbb{R}^{m\times m}$, $n \ge 2$ be a sequence of weight matrices,

and $\mathbf{b} := \{\mathbf{b}_n\}_{n=1}^{\infty}$ with $\mathbf{b}_n \in \mathbb{R}^m$ be a sequence of bias vectors. Define the deep neural network by

$$\mathcal{N}_n(x) := \left(\bigodot_{i=1}^{n} \sigma(\mathbf{W}_i \cdot + \mathbf{b}_i)\right)(x), \quad x \in [0,1]^d. \tag{2.4}$$

We say the deep neural network $\mathcal{N}_n$ determined by $\mathbf{W}$, $\mathbf{b}$ and a chosen activation function σ converges with respect to some norm $\|\cdot\|$ to a limit function $\mathcal{N}$ if

$$\lim_{n\to\infty} \|\mathcal{N}_n - \mathcal{N}\| = 0.$$

The goal of this paper is to understand what conditions are required for the weight matrices $\mathbf{W}_n$ and the bias vectors $\mathbf{b}_n$ to ensure convergence of the deep neural network when the activation function is chosen to be ReLU.

3. Convergence of ReLU networks

In this section, we consider convergence of a deep ReLU network $\mathcal{N}_n$ as the number n of layers goes to infinity. For this purpose, we introduce an algebraic formulation of a deep ReLU network convenient for convergence analysis.

It has been understood [3] that the neural network (2.1) with σ being the ReLU activation function determines a function

$$f_n(x) = W_o \mathcal{N}_n(x) + b_o, \quad x \in [0,1]^d,$$

that is piecewise linear. Our novelty is to identify the linear components of $\mathcal{N}_n$ and their associated subdomains by using a sequence of activation matrices.

We begin with analyzing a one layer ReLU network $\mathcal{N}_1$, which has the form

$$\mathcal{N}_1(x) := \sigma(\mathbf{W}_1 x + \mathbf{b}_1), \quad x \in [0,1]^d,$$

where σ is the ReLU activation function. Note that the m components of $\mathbf{W}_1 x + \mathbf{b}_1$ are linear functions $\ell_j(x)$, $x \in [0,1]^d$ for $j = 1, 2, \dots, m$. Hence,

$$\mathcal{N}_1(x) = [\sigma(\ell_j(x)) : j = 1, 2, \dots, m]^T. \tag{3.1}$$

According to the definition of the ReLU activation function, we observe for $j = 1, 2, \dots, m$ that

$$\sigma(\ell_j(x)) = 0, \quad \text{if} \quad \ell_j(x) \le 0, \quad \text{and} \quad \sigma(\ell_j(x)) = \ell_j(x), \quad \text{if} \quad \ell_j(x) > 0. \tag{3.2}$$

When $\sigma(\ell_j(x)) = 0$, we say that the node with $\ell_j(x)$ is **deactivated**, and otherwise, it is **activated**. Apparently, there are at most 2^m different activation patterns at the first layer. To describe these patterns, we introduce a set of $m \times m$ diagonal matrices whose diagonal entries are either 1 or 0.

Specifically, we define the set of **activation matrices** by

$$D_m := \left\{ \operatorname{diag}(a_1, a_2, \dots, a_m) : a_i \in \{0,1\}, 1 \le i \le m \right\}.$$

An element of D_m is either the identity matrix or its degenerated matrix (some diagonal entries degenerated to zero). The support of an activation matrix $J \in D_m$ is defined by

$$\operatorname{supp} J := \{k : \ J_{kk} = 1, \quad 1 \le k \le m\}.$$

Clearly, an activation matrix $J \in D_m$ is uniquely determined by its support. The set D_m of the activation matrices has exactly 2^m elements since each of the m diagonal entries of an element in the set has exactly two different choices. This matches the number of possible different activation patterns of a ReLU neural network: Each element of the set D_m corresponds to an activation pattern. For this reason, it is convenient to use D_m as an index set.

Definition 3.1 (*Activation Domains of One Layer Network*)**.** For a weight matrix $\mathbf{W}$ with m rows and a bias vector $\mathbf{b} \in \mathbb{R}^m$, the activation domain of $\sigma(\mathbf{W}x + \mathbf{b})$ with respect to a diagonal matrix $J \in D_m$ is

$$D_{J,\mathbf{W},\mathbf{b}} := \Big\{ x \in \mathbb{R}^{m'} : (\mathbf{W}x + \mathbf{b})_j > 0 \ \text{ for } j \in \operatorname{supp} J \text{ and } (\mathbf{W}x + \mathbf{b})_j \le 0 \text{ for } j \notin \operatorname{supp} J \Big\}.$$

Note that the integer m' in Definition 3.1 may be chosen to be d (when it is used to define activation domains of the first layer) or m (when it is used to define activation domains of layers which are not the first layer).

In Definition 3.1, we use an activation matrix $J \in D_m$ to associate an activation pattern of the m components of $\mathbf{W}x + \mathbf{b}$. As a result, Definition 3.1 enables us to construct a partition of the unit cube $[0,1]^d$ that corresponds to the piecewise linear nature of the function $\mathcal{N}_1$ and allows us to reexpress $\mathcal{N}_1$ in a piecewise linear manner. Specifically, we have that

$$[0,1]^d = \bigcup_{I_1 \in D_m} (D_{I_1,\mathbf{W}_1,\mathbf{b}_1} \cap [0,1]^d). \tag{3.3}$$

By Eqs. (3.1) and (3.3), the one layer ReLU network can be reexpressed as

$$\mathcal{N}_1(x) = I_1(\mathbf{W}_1 x + \mathbf{b}_1), \quad x \in D_{I_1,\mathbf{W}_1,\mathbf{b}_1}, \quad \text{for} \quad I_1 \in D_m. \tag{3.4}$$

Clearly, on each activation domain $D_{I_1,\mathbf{W}_1,\mathbf{b}_1}$, $\mathcal{N}_1$ is a linear function. The essence of Eq. (3.4) is that we are able to replace the application of the ReLU activation function by multiplication with an activation matrix in D_m. This will lead to great convenience in processing ReLU networks. We remark that some of the 2^m activation domains might be empty. In fact, by [25], the number of activation domains with nonempty interior does not exceed $\sum_{k=0}^{d} \binom{m}{k}$.

For a deep ReLU neural network with n layers, we need a sequence of n activation matrices $\bar{\mathbf{I}}_n := (I_1, I_2, \dots, I_n) \in (D_m)^n$ to identify its different activation patterns on the n hidden layers, where I_k marks the activation pattern at the kth layer. We next define activation domains of a multi-layer network.

Definition 3.2 (*Activation Domains of a Multi-Layer Network*)**.** For $\bar{\mathbf{W}}_n := (\mathbf{W}_1, \dots, \mathbf{W}_n) \in \mathbb{R}^{m\times d} \times (\mathbb{R}^{m\times m})^{n-1}$, $\bar{\mathbf{b}}_n := (\mathbf{b}_1, \dots, \mathbf{b}_n) \in (\mathbb{R}^m)^n$, the activation domain of

$$\bigodot_{i=1}^{n} \sigma(\mathbf{W}_i \cdot + \mathbf{b}_i)$$

with respect to $\bar{\mathbf{I}}_n := (I_1, \dots, I_n) \in (D_m)^n$ is defined recursively by

$$D_{\bar{\mathbf{I}}_1,\bar{\mathbf{W}}_1,\bar{\mathbf{b}}_1} = D_{I_1,\mathbf{W}_1,\mathbf{b}_1} \cap [0,1]^d$$

and

$$D_{\bar{\mathbf{I}}_n,\bar{\mathbf{W}}_n,\bar{\mathbf{b}}_n} = \left\{ x \in D_{\bar{\mathbf{I}}_{n-1},\bar{\mathbf{W}}_{n-1},\bar{\mathbf{b}}_{n-1}} : \left(\bigodot_{i=1}^{n-1} \sigma(\mathbf{W}_i \cdot + \mathbf{b}_i)\right)(x) \in D_{I_n,\mathbf{W}_n,\mathbf{b}_n} \right\}.$$

We have the following observation regarding the activation domain.

Proposition 3.3. *The sequence of the activation domains $D_{\bar{\mathbf{I}}_n,\bar{\mathbf{W}}_n,\bar{\mathbf{b}}_n}$ are nested, that is,*

$$D_{\bar{\mathbf{I}}_{n+1},\bar{\mathbf{W}}_{n+1},\bar{\mathbf{b}}_{n+1}} \subseteq D_{\bar{\mathbf{I}}_n,\bar{\mathbf{W}}_n,\bar{\mathbf{b}}_n}, \quad n \in \mathbb{N}.$$

Moreover, for each $n \in \mathbb{N}$,

$$D_{\bar{\mathbf{I}}_n,\bar{\mathbf{W}}_n,\bar{\mathbf{b}}_n} = \left\{ x \in D_{I_1,\mathbf{W}_1,\mathbf{b}_1} \cap [0,1]^d : \left(\bigodot_{i=1}^{k-1} \sigma(\mathbf{W}_i \cdot + \mathbf{b}_i)\right)(x) \in D_{I_k,\mathbf{W}_k,\mathbf{b}_k}, \ 2 \le k \le n \right\}. \tag{3.5}$$

Proof. That $D_{\bar{\mathbf{I}}_n,\bar{\mathbf{W}}_n,\bar{\mathbf{b}}_n}$ are nested follows directly from the definition. Equality (3.5) may be proved by induction on n. □

The activation domain $D_{\bar{\mathbf{I}}_n,\bar{\mathbf{W}}_n,\bar{\mathbf{b}}_n}$ characterizes the inputs $x \in [0,1]^d$ such that when those inputs are going through the ReLU neural network (2.1), at the kth hidden layer ($1 \le k \le n$), exactly the nodes with index in $\operatorname{supp} I_k$ are activated. There are at most 2^{nm} activation domains $D_{\bar{\mathbf{I}}_n,\bar{\mathbf{W}}_n,\bar{\mathbf{b}}_n}$ corresponding to all choices of sequences of diagonal matrices $\bar{\mathbf{I}}_n \in (\mathcal{D}_m)^n$, and a large number of them might be empty or have zero Lebesgue measure.

For each positive integer n, the activation domains

$$D_{\bar{\mathbf{I}}_n,\bar{\mathbf{W}}_n,\bar{\mathbf{b}}_n}, \quad \text{for} \quad \bar{\mathbf{I}}_n := (I_1,\dots,I_n) \in (\mathcal{D}_m)^n,$$

form a partition of the unit cube $[0,1]^d$. That is, for each $n \in \mathbb{N}$,

$$[0,1]^d = \bigcup_{\bar{\mathbf{I}}_n \in (\mathcal{D}_m)^n} D_{\bar{\mathbf{I}}_n,\bar{\mathbf{W}}_n,\bar{\mathbf{b}}_n}$$

By using these activation domains, we are able to write down an explicit expression of the ReLU network $\mathcal{N}_n$ with applications of the ReLU activation function replaced by multiplications with the activation matrices.

We now establish a representation of the ReLU network based on its activation domains and activation matrices. To this end, we need a notation to denote the product of matrices with a prescribed order. Specifically, we write

$$\prod_{i=1}^{n} \mathbf{W}_i = \mathbf{W}_n \mathbf{W}_{n-1} \cdots \mathbf{W}_1.$$

For $n,k \in \mathbb{N}$, we also adopt the following convention that

$$\prod_{i=k}^{n} \mathbf{W}_i = \mathbf{W}_n \mathbf{W}_{n-1} \cdots \mathbf{W}_k, \quad \text{for} \quad n \ge k, \quad \text{and} \quad \prod_{i=k}^{n} \mathbf{W}_i = I, \quad \text{for} \quad n < k,$$

where I denotes the $m \times m$ identity matrix.

Theorem 3.4. *It holds that*

$$\mathcal{N}_n(x) = \left(\prod_{i=1}^{n} I_i \mathbf{W}_i\right) x + \sum_{i=1}^{n} \left(\prod_{j=i+1}^{n} I_j \mathbf{W}_j\right) I_i \mathbf{b}_i, \quad x \in D_{\bar{\mathbf{I}}_n,\bar{\mathbf{W}}_n,\bar{\mathbf{b}}_n}, \tag{3.6}$$
$$\bar{\mathbf{I}}_n := (I_1,\dots,I_n) \in (\mathcal{D}_m)^n.$$

Proof. We prove by induction on n. When $n = 1$, by (3.4), the result is true. Suppose that (3.6) holds for $n-1$. Now let $x \in D_{\bar{\mathbf{I}}_n,\bar{\mathbf{W}}_n,\bar{\mathbf{b}}_n}$. Then

$$\mathcal{N}_n(x) = \sigma\left(\mathbf{W}_n\left(\bigodot_{i=1}^{n-1} \sigma(\mathbf{W}_i \cdot + \mathbf{b}_i)\right)(x) + \mathbf{b}_n\right).$$

By Definition 3.2,

$$\left(\bigodot_{i=1}^{n-1} \sigma(\mathbf{W}_i \cdot + \mathbf{b}_i)\right)(x) \in D_{I_n,\mathbf{W}_n,\mathbf{b}_n}.$$

We then get by (3.4) and induction that

$$\begin{aligned}
\mathcal{N}_n(x) &= I_n\left(\mathbf{W}_n\left(\bigodot_{i=1}^{n-1} \sigma\left(\mathbf{W}_i \cdot + \mathbf{b}_i\right)\right)(x) + \mathbf{b}_n\right) \\
&= I_n \mathbf{W}_n\left(\left(\prod_{i=1}^{n-1} I_i \mathbf{W}_i\right) x + \sum_{i=1}^{n-1}\left(\prod_{j=i+1}^{n-1} I_j \mathbf{W}_j\right) I_i \mathbf{b}_i\right) + I_n \mathbf{b}_n \\
&= \left(\prod_{i=1}^{n} I_i \mathbf{W}_i\right) x + \sum_{i=1}^{n}\left(\prod_{j=i+1}^{n} I_j \mathbf{W}_j\right) I_i \mathbf{b}_i,
\end{aligned}$$

which proves (3.6). □

The representation of a deep ReLU network established in Theorem 3.4 is crucial for further investigation of the network. The piecewise linear property of a ReLU network follows immediately from this representation. It is also helpful for developing the convergence results of ReLU Networks later in this paper.

In the remaining part of this section, we formulate the convergence of deep ReLU networks as a problem about convergence of infinite products of matrices. Denote by $\|\cdot\|_p$ the ℓ^p-norm on $\mathbb{R}^m$, $1 \le p \le +\infty$. For a Lebesgue measurable subset $\Omega \subseteq \mathbb{R}^d$, by $L^p(\Omega,\mathbb{R}^m)$ we denote the space of all real-valued functions $f : \Omega \to \mathbb{R}^m$ such that each component of f is Lebesgue measurable on Ω and such that

$$\|f\|_p := \begin{cases} \left(\int_\Omega \|f(x)\|_p^p dx\right)^p, & 1 \le p < +\infty, \\ \operatorname*{ess\,sup}_{x\in\Omega} \|f(x)\|_\infty, & p = +\infty \end{cases}$$

is finite. Also, $C(\Omega,\mathbb{R}^m)$ is the space of all continuous functions from Ω to $\mathbb{R}^m$.

Theorem 3.4 allows us to present a necessary and sufficient condition for ReLU neural networks $\mathcal{N}_n$ to converge to a function in $L^p([0,1]^d,\mathbb{R}^m)$, as $n \to \infty$. Let $\mathbf{W} := \{\mathbf{W}_n\}_{n=1}^\infty$ with $\mathbf{W}_1 \in \mathbb{R}^{m\times d}$, $\mathbf{W}_n \in \mathbb{R}^{m\times m}$, $n > 1$ and $\mathbf{b} := \{\mathbf{b}_n\}_{n=1}^\infty$ with $\mathbf{b}_n \in \mathbb{R}^m$ be a sequence of weight matrices and bias vectors, respectively. Suppose $\mathcal{N} \in C([0,1]^d,\mathbb{R}^m)$. It follows from Theorem 3.4 that the ReLU neural networks $\mathcal{N}_n$ converge to $\mathcal{N}$ in $L^p([0,1]^d,\mathbb{R}^m)$ if and only if

$$\lim_{n\to\infty} \sum_{\bar{\mathbf{I}}_n \in (\mathcal{D}_m)^n} \int_{D_{\bar{\mathbf{I}}_n,\bar{\mathbf{W}}_n,\bar{\mathbf{b}}_n}} \left\| \left(\prod_{i=1}^{n} I_i \mathbf{W}_i\right) x + \sum_{i=1}^{n}\left(\prod_{j=i+1}^{n} I_j \mathbf{W}_j\right) I_i \mathbf{b}_i - \mathcal{N}(x) \right\|_p^p dx = 0, \quad 1 \le p < +\infty \tag{3.7}$$

and

$$\lim_{n\to\infty} \max_{\bar{\mathbf{I}}_n \in (\mathcal{D}_m)^n} \sup_{x \in D_{\bar{\mathbf{I}}_n,\bar{\mathbf{W}}_n,\bar{\mathbf{b}}_n}} \left\| \left(\prod_{i=1}^{n} I_i \mathbf{W}_i\right) x + \sum_{i=1}^{n}\left(\prod_{j=i+1}^{n} I_j \mathbf{W}_j\right) I_i \mathbf{b}_i - \mathcal{N}(x) \right\|_\infty = 0, \quad p = \infty. \tag{3.8}$$

This necessary and sufficient condition together with Theorem 3.4 leads to useful necessary conditions and sufficient conditions for the sequence of ReLU neural networks to converge. They will be presented next. To this end, we first establish a technical lemma.

Lemma 3.5. *Let $A_n \in \mathbb{R}^{m\times d}$, $b_n \in \mathbb{R}^m$, $n \in \mathbb{N}$ and let $1 \le p \le +\infty$. Then the sequence of linear functions*

$$A_n x + b_n$$

converges in $L^p(\Omega,\mathbb{R}^m)$ on a bounded subset $\Omega \subseteq \mathbb{R}^d$ that has positive Lebesgue measure if and only if both $\{A_n\}$ and $\{b_n\}$ converge.

Proof. We first prove the sufficient condition. Suppose that both $\{A_n\}$ and $\{b_n\}$ converge. Then, clearly, $A_n x + b_n$ converges uniformly on Ω as Ω is bounded. As a result, $\{A_n x + b_n\}$ converges in $L^p(\Omega,\mathbb{R}^m)$ for all $1 \le p \le +\infty$.

Conversely, suppose that $\{A_n x + b_n\}$ converges to some limit function $\mathbf{u} := (u_1,u_2,\dots,u_m)^T$ in $L^p(\Omega,\mathbb{R}^m)$ for some $p \in [1,+\infty]$, where Ω has positive Lebesgue measure. Let $b_n := (b_{n1},b_{n2},\dots,b_{nm})^T$ and $A_n := [A_{n,jk} : 1 \le j \le m, 1 \le k \le d]$. Thus, for each j with $1 \le j \le m$, we have that

$$(A_n x + b_n)_j = b_{nj} + \sum_{k=1}^{d} A_{n,jk} x_k \to u_j \text{ in } L^p(\Omega) \text{ as } n \to \infty. \tag{3.9}$$

As Ω has a positive measure, $C(\Omega)$ is infinite-dimensional. Therefore, there exists $g \in C(\Omega)$ such that

$$\int_\Omega g(x) x_k dx = 0, \quad \text{for all} \quad 1 \le k \le d \quad \text{and} \quad \int_\Omega g(x) dx = 1.$$

Eq. (3.9) ensures that

$$\lim_{n\to\infty} b_{nj} = \lim_{n\to\infty} \int_\Omega g(x)(A_n x + b_n)_j dx = \int_\Omega g(x) u_j(x) dx,$$

which implies that for every $1 \le j \le m$, b_{nj} converges as $n \to \infty$. Hence, $\{b_n\}$ converges. Similarly, for each l with $1 \le l \le d$, there exists a function $h_l \in C(\Omega)$ such that

$$\int_\Omega h_l(x) x_k dx = \delta_{l,k}, \quad \text{for all} \quad 1 \le k \le d \quad \text{and} \quad \int_\Omega h_l(x) dx = 0.$$

Again, by (3.9), we have that

$$\lim_{n\to\infty} A_{n,jl} = \lim_{n\to\infty} \int_\Omega h_l(x)(A_n x + b_n)_j dx = \int_\Omega h_l(x) u_j(x) dx,$$

which proves the convergence of $\{A_n\}$. □

We are now ready to prove the main result of this section.

Theorem 3.6. *Let $\mathbf{W} := \{\mathbf{W}_n\}_{n=1}^\infty$ with $\mathbf{W}_1 \in \mathbb{R}^{m\times d}$, $\mathbf{W}_n \in \mathbb{R}^{m\times m}$, $n > 1$ and $\mathbf{b} := \{\mathbf{b}_n\}_{n=1}^\infty$ with $\mathbf{b}_n \in \mathbb{R}^m$ be a sequence of weight matrices and bias vectors, respectively.*

1. **(Necessary condition for convergence)** *If the sequence of ReLU networks $\{\mathcal{N}_n\}_{n=1}^\infty$ converges in $L^p([0,1]^d, \mathbb{R}^m)$ then for all sequences of diagonal matrices $\mathbb{I} = (I_n \in \mathcal{D}_m : \ n \in \mathbb{N})$ such that the set*

$$\bigcap_{n=1}^\infty D_{\bar{\mathbf{I}}_n, \bar{\mathbf{W}}_n, \bar{\mathbf{b}}_n}$$

has positive Lebesgue measure, the two limits

$$\prod_{n=1}^\infty I_n \mathbf{W}_n := \lim_{n\to\infty} \prod_{i=1}^n I_i \mathbf{W}_i \tag{3.10}$$

and

$$\lim_{n\to\infty} \sum_{i=1}^n \left(\prod_{j=i+1}^n I_j \mathbf{W}_j \right) I_i \mathbf{b}_i \tag{3.11}$$

both exist.

2. **(Sufficient condition for pointwise convergence)** *If for all sequences of diagonal matrices $\mathbb{I} = (I_n \in \mathcal{D}_m : \ n \in \mathbb{N})$, the above two limits both exist, then the sequence of ReLU neural networks $\{\mathcal{N}_n\}_{n=1}^\infty$ converges pointwise on $[0,1]^d$.*

Proof. We prove the first claim of this theorem. If for a sequence of diagonal matrices $\mathbb{I} = (I_n \in \mathcal{D}_m : \ n \in \mathbb{N})$,

$$D_{\mathbb{I}} := \bigcap_{n=1}^\infty D_{\bar{\mathbf{I}}_n, \bar{\mathbf{W}}_n, \bar{\mathbf{b}}_n}$$

has positive Lebesgue measure, then by (3.7) and (3.8),

$$\begin{aligned}\|\mathcal{N}_n - \mathcal{N}\|_{L^p(D_{\mathbb{I}}, \mathbb{R}^m)} &\le \|\mathcal{N}_n - \mathcal{N}\|_{L^p(D_{\bar{\mathbf{I}}_n, \bar{\mathbf{W}}_n, \bar{\mathbf{b}}_n}, \mathbb{R}^m)} \\ &\le \|\mathcal{N}_n - \mathcal{N}\|_{L^p([0,1]^d, \mathbb{R}^m)} \to 0, \ n \to \infty.\end{aligned}$$

It implies

$$\mathcal{N}_n(x) = \left(\prod_{i=1}^n I_i \mathbf{W}_i \right) x + \sum_{i=1}^n \left(\prod_{j=i+1}^n I_j \mathbf{W}_j \right) I_i \mathbf{b}_i$$

converges in $D_{\mathbb{I}}$ with respect to the chosen L^p-norm $\|\cdot\|$. The proof may be completed by applying Lemma 3.5 with $\Omega := D_{\mathbb{I}}$.

Next, we establish the second claim. For every $x \in [0,1]^d$, there exists a sequence of diagonal matrices $\mathbb{I} = (I_n : \ I_n \in \mathcal{D}_m, \ n \in \mathbb{N})$ such that

$$x \in \bigcap_{n=1}^\infty D_{\bar{\mathbf{I}}_n, \bar{\mathbf{W}}_n, \bar{\mathbf{b}}_n}.$$

Thus, by (3.6)

$$\mathcal{N}_n(x) = \left(\prod_{i=1}^n I_i \mathbf{W}_i \right) x + \sum_{i=1}^n \left(\prod_{j=i+1}^n I_j \mathbf{W}_j \right) I_i \mathbf{b}_i, \quad n \in \mathbb{N}.$$

Therefore, the existence of the two limits (3.10) and (3.11) are sufficient for pointwise convergence of $\{\mathcal{N}_n(x)\}$. □

Theorem 3.6 lays a foundation for studying the convergence issue of deep ReLU networks.

4. Infinite products of matrices

This section is devoted to investigation of convergence of infinite products of matrices, which arise in the study of convergence of deep ReLU networks.

It follows from Theorem 3.6 that the convergence of ReLU networks is reduced to existence of the two limits (3.10) and (3.11). Specifically, the convergence of the infinite product of matrices

$$\prod_{n=1}^\infty I_n \mathbf{W}_n, \quad \text{for any} \quad I_n \in \mathcal{D}_m, \tag{4.1}$$

appears in the two limits. We hence raise the following question: What conditions on the matrices $\mathbf{W}_n$, $n \in \mathbb{N}$, will guarantee the convergence of the infinite product (4.1) for all choices $I_n \in \mathcal{D}_m$, $n \in \mathbb{N}$? We first answer this question.

There is a well-known sufficient condition ([26], page 127) for convergence of infinite products of matrices, which can be considered as a generalization of the convergence of infinite products $\prod_{n=1}^\infty (1 + x_i)$ of scalars. The result states that if

$$\mathbf{W}_n = I + \mathbf{P}_n \text{ and } \sum_{n=1}^\infty \|\mathbf{P}_n\| < +\infty, \tag{4.2}$$

where I is the identity matrix and $\|\cdot\|$ is any matrix norm satisfying $\|AB\| \le \|A\|\|B\|$, then the infinite product

$$\prod_{n=1}^\infty \mathbf{W}_n$$

converges. This result was extended by Artzrouni [27]. Our question differs from those results in having the diagonal matrices I_n in (4.1) arbitrarily chosen from $\mathcal{D}_m$. Nevertheless, we manage to prove that this condition (4.2) remains sufficient for the convergence of (4.1). We proceed to establish this result.

Let $\|\cdot\|$ be a norm on $\mathbb{R}^m$ that is nondecreasing on the modules of vector components:

$$\begin{aligned}&\|\mathbf{a}\| \le \|\mathbf{b}\| \text{ whenever } |a_i| \le |b_i|, \ 1 \le i \le m, \\ &\qquad \text{for } \ \mathbf{a} = (a_1, a_2, \dots, a_m), \mathbf{b} = (b_1, b_2, \dots, b_m) \in \mathbb{R}^m.\end{aligned} \tag{4.3}$$

This requirement on a vector norm is mild and it is satisfied by the ℓ^p-norms for all $1 \le p \le +\infty$. We then define its induced matrix norm on $\mathbb{R}^{m\times m}$, also denoted by $\|\cdot\|$, by

$$\|A\| = \sup_{x\in\mathbb{R}^m, x\neq 0} \frac{\|Ax\|}{\|x\|}, \quad \text{for} \quad A \in \mathbb{R}^{m\times m}.$$

Clearly, this matrix norm has the property that

$$\|AB\| \le \|A\|\|B\| \text{ for all matrices } A, B \tag{4.4}$$

and

$$\|I_i\| \le 1 \text{ for each } I_i \in \mathcal{D}_m. \tag{4.5}$$

Note that the Frobenius norm satisfies (4.4) but does not satisfy (4.5).

Our first observation regards the product of activation matrices.

Lemma 4.1. *If $j \ge 2$ and $I_i \in \mathcal{D}_m$ for $i = j, j+1, \dots, n$, then*

$$\lim_{n\to\infty} \prod_{i=j}^n I_i = \mathcal{I}_j,$$

for some matrix $\mathcal{I}_j \in \mathcal{D}_m$, and there exist a positive integer N such that

$$\prod_{i=j}^n I_i = \mathcal{I}_j, \quad \textit{whenever} \quad n > N. \tag{4.6}$$

Proof. We first note that for all $n \in \mathbb{N}$, there holds $\prod_{i=j}^n I_i \in \mathcal{D}_m$ and

$$\operatorname{supp}\left(\prod_{i=j}^n I_i \right) = \bigcap_{i=j}^n \operatorname{supp} I_i.$$

It follows that

$$\emptyset \subseteq \operatorname{supp}\left(\prod_{i=j}^{n+1} I_i \right) \subseteq \operatorname{supp}\left(\prod_{i=j}^n I_i \right), \tag{4.7}$$

where $\emptyset$ denotes the empty set. Therefore, the limit

$$\lim_{n\to\infty} \operatorname{supp}\left(\prod_{i=j}^{n} I_j\right) = \bigcap_{i=j}^{\infty} \operatorname{supp} I_i$$

exists. Note that there exists a unique diagonal matrix $\mathcal{I}_j \in \mathcal{D}_m$ such that

$$\operatorname{supp} \mathcal{I}_j = \bigcap_{i=j}^{\infty} \operatorname{supp} I_i$$

and thus,

$$\lim_{n\to\infty} \prod_{i=j}^{n} I_i = \mathcal{I}_j.$$

Since the set $\mathcal{D}_m$ contains only a finite number of matrices, according to (4.7), there exists a positive integer N such that

$$\operatorname{supp}\left(\prod_{i=j}^{n} I_i\right) = \operatorname{supp}\left(\prod_{i=j}^{N} I_i\right), \quad \text{for all} \quad n > N.$$

Thus, there exists a unique diagonal matrix $\mathcal{I}_j \in \mathcal{D}_m$ such that

$$\mathcal{I}_j = \prod_{i=j}^{N} I_i$$

and

$$\prod_{i=j}^{n} I_i = \mathcal{I}_j, \quad \text{for all} \quad n > N. \quad \square$$

Lemma 4.2. *If a sequence $\{a_n\}_{n=1}^{\infty}$ satisfies $a_n \geq 0$ and $\sum_{n=1}^{\infty} a_n < +\infty$, then for all $p \in \mathbb{N}$,*

$$\sum_{i=p+1}^{\infty} a_i + \sum_{l=2}^{\infty} \sum_{\substack{1\leq i_1<i_2<\cdots<i_l \\ i_l>p}} \prod_{k=1}^{l} a_{i_k} \leq \left(\sum_{i=p+1}^{\infty} a_i\right) \exp\left(\sum_{i=1}^{\infty} a_i\right). \tag{4.8}$$

Proof. Recall the expansion

$$e^x = \sum_{l=0}^{\infty} \frac{x^l}{l!}, \quad \text{for all} \quad x \in \mathbb{R}. \tag{4.9}$$

Substituting $x := \sum_{i=1}^{\infty} a_i$ in Eq. (4.9) yields that

$$\exp\left(\sum_{i=1}^{\infty} a_i\right) = \sum_{l=0}^{\infty} \frac{1}{l!} \left(\sum_{i=1}^{\infty} a_i\right)^l.$$

Multiplying both sides of the above equation by the sum $\sum_{i=p+1}^{\infty} a_i$ gives that

$$\left(\sum_{i=p+1}^{\infty} a_i\right) \exp\left(\sum_{i=1}^{\infty} a_i\right) = \left(\sum_{i=p+1}^{\infty} a_i\right) \sum_{l=0}^{\infty} \frac{1}{l!} \left(\sum_{i=1}^{\infty} a_i\right)^l. \tag{4.10}$$

Note that for $l \geq 1$

$$\left(\sum_{i=p+1}^{\infty} a_i\right) \frac{1}{(l-1)!} \left(\sum_{i=1}^{\infty} a_i\right)^{l-1} \geq \sum_{\substack{1\leq i_1<i_2<\cdots<i_l \\ i_l>p}} \prod_{k=1}^{l} a_{i_k}.$$

Combining this inequality with Eq. (4.10) proves the desired inequality of this lemma. $\square$

When the infinite sum in Lemma 4.2 is reduced to a finite sum, we obtain the following special result that was originally used in [26] without a proof. If a_i, $1 \leq i \leq n$, are nonnegative numbers, by setting $a_i = 0$ for $i > n$, we then obtain from Lemma 4.2 that for all $p < n$,

$$\sum_{i=p+1}^{n} a_i + \sum_{l=2}^{n} \sum_{\substack{1\leq i_1<i_2<\cdots<i_l\leq n \\ i_l>p}} \prod_{k=1}^{l} a_{i_k} \leq \left(\sum_{i=p+1}^{n} a_i\right) \exp\left(\sum_{i=1}^{n} a_i\right). \tag{4.11}$$

We next provide a sufficient condition on the matrices which ensures convergence of the infinite product (4.1). In [27], it was proved that if

$$\sum_{i=1}^{\infty} \|A_i\| < +\infty$$

and

$$\|U_i\| = 1 \text{ for all } i \in \mathbb{N} \tag{4.12}$$

then

$$\prod_{i=1}^{\infty} (U_i + A_i) \tag{4.13}$$

converges. Our infinite products (4.1) differ from (4.13) in that we have I_n's arbitrarily chosen from $\mathcal{D}_m$. Also, the assumption (4.12) does not apply to our question. We shall make use of the special property (4.6) of I_n's, which making our approach more direct and simple than that in [27].

Theorem 4.3. *Let $\|\cdot\|$ be a matrix norm satisfying (4.4) and (4.5). If $\mathbf{W}_1 \in \mathbb{R}^{m\times d}$, $\mathbf{W}_n \in \mathbb{R}^{m\times m}$, $n \geq 2$, are matrices satisfying*

$$\mathbf{W}_n = I + \mathbf{P}_n, n \geq 2, \text{ and } \sum_{n=2}^{\infty} \|\mathbf{P}_n\| < +\infty, \tag{4.14}$$

then the infinite product (4.1) converges for all $I_n \in \mathcal{D}_m$, $n \in \mathbb{N}$.

Proof. It suffices to prove that the infinite product of matrices

$$\prod_{n=2}^{\infty} I_n \mathbf{W}_n$$

converges under the assumed conditions.

We compute that

$$\prod_{i=2}^{n} I_i \mathbf{W}_i = \prod_{i=2}^{n} (I_i + I_i \mathbf{P}_i).$$

Expanding the product on the right hand side of the above equation yields

$$\prod_{i=2}^{n} I_i \mathbf{W}_i = \prod_{i=2}^{n} I_i + \sum_{l=1}^{n-1} \sum_{2\leq j_1<j_2<\cdots<j_l\leq n} \left(\prod_{k=j_l+1}^{n} I_k\right) \times \left(\prod_{i=2}^{l} \left(I_{j_i} \mathbf{P}_{j_i} \prod_{k=j_{i-1}+1}^{j_i-1} I_k\right)\right) I_{j_1} \mathbf{P}_{j_1} \left(\prod_{k=2}^{j_1-1} I_k\right). \tag{4.15}$$

According to (4.6), we assume that $n' > n > p$ are large enough integers so that

$$\prod_{i=j}^{n} I_i = \prod_{i=j}^{n'} I_i = \mathcal{I}_j, \quad 2 \leq j \leq p+1$$

for some $\mathcal{I}_j \in \mathcal{D}_m$. This fact ensures that for $1 \leq l \leq n-1$, the terms in

$$\sum_{2\leq j_1<j_2<\cdots<j_l\leq n} \left(\prod_{k=j_l+1}^{n} I_k\right) \left(\prod_{i=2}^{l} \left(I_{j_i} \mathbf{P}_{j_i} \prod_{k=j_{i-1}+1}^{j_i-1} I_k\right)\right) I_{j_1} \mathbf{P}_{j_1} \left(\prod_{k=2}^{j_1-1} I_k\right),$$

which appear in (4.15) and those in

$$\sum_{2\leq j_1<j_2<\cdots<j_l\leq n'} \left(\prod_{k=j_l+1}^{n'} I_k\right) \left(\prod_{i=2}^{l} \left(I_{j_i} \mathbf{P}_{j_i} \prod_{k=j_{i-1}+1}^{j_i-1} I_k\right)\right) I_{j_1} \mathbf{P}_{j_1} \left(\prod_{k=2}^{j_1-1} I_k\right)$$

which appear in (4.15) with n replaced by n' are identical if $j_l \leq p$.

Now, for $n' > n > p$ we consider the difference

$$\prod_{i=2}^{n} I_i \mathbf{W}_i - \prod_{i=2}^{n'} I_i \mathbf{W}_i$$

and use (4.15) with the fact pointed out above so that the identical terms appearing in the two products are canceled. By applying the

matrix norm to the resulting sum, we get by (4.4) and (4.5)

$$\left\|\prod_{i=2}^{n} I_i\mathbf{W}_i - \prod_{i=2}^{n'} I_i\mathbf{W}_i\right\| \le \sum_{j=p+1}^{n}\|\mathbf{P}_j\| + \sum_{j=p+1}^{n'}\|\mathbf{P}_j\| + \sum_{l=2}^{n-1}\sum_{\substack{2\le j_1<j_2<\cdots<j_l\le n\\ j_1>p}}\prod_{k=1}^{l}\|\mathbf{P}_{j_k}\|$$
$$+\sum_{l=2}^{n-1}\sum_{\substack{2\le j_1<j_2<\cdots<j_l\le n'\\ j_1>p}}\prod_{k=1}^{l}\|\mathbf{P}_{j_k}\| + \sum_{l=n}^{n'-1}\sum_{2\le j_1<j_2<\cdots<j_l\le n'}\prod_{k=1}^{l}\|\mathbf{P}_{j_k}\|.$$

Invoking inequality (4.11) we obtain from the last inequality for large enough positive integers $n' > n$

$$\left\|\prod_{i=2}^{n} I_i\mathbf{W}_i - \prod_{i=2}^{n'} I_i\mathbf{W}_i\right\| \le 2\left(\sum_{i=p+1}^{\infty}\|\mathbf{P}_i\|\right)\exp\left(\sum_{i=2}^{\infty}\|\mathbf{P}_i\|\right). \tag{4.16}$$

Finally, the second inequality of (4.14) ensures that for $\varepsilon > 0$, there exists $p \in \mathbb{N}$ such that

$$\sum_{j=p+1}^{\infty}\|\mathbf{P}_j\| < \varepsilon. \tag{4.17}$$

Using estimate (4.17) in the right hand side of (4.16) yields

$$\left\|\prod_{i=2}^{n} I_i\mathbf{W}_i - \prod_{i=2}^{n'} I_i\mathbf{W}_i\right\| \le 2\varepsilon\exp\left(\sum_{i=2}^{\infty}\|\mathbf{P}_i\|\right),$$

which together with the second inequality of condition (4.14) proves the convergence of the infinite product (4.1). □

We next deal with the second limit (3.11). Our first task is to formulate a necessary condition, showing that the linear function $\mathbf{W}_n x + b_n$ on the nth layer will be close to the identity mapping for sufficiently large n.

Theorem 4.4. *Let $\|\cdot\|$ be a norm on $\mathbb{R}^m$ that satisfies (4.3) and $\|\cdot\|$ be its induced matrix norm. Suppose that the matrices $\mathbf{W}_n$, $n \ge 2$, satisfy*

$$\mathbf{W}_n = I + \mathbf{P}_n, n \ge 2, \ \sum_{n=2}^{\infty}\|\mathbf{P}_n\| < +\infty, \ \text{and} \ \sum_{i=n+1}^{\infty}\|\mathbf{P}_i\| = o\left(\frac{1}{n}\right), \ n \to \infty, \tag{4.18}$$

and that the vectors $\mathbf{b}_n$, $n \in \mathbb{N}$, are bounded. If the limit (3.11) exists for all choices of matrices $I_i \in \mathcal{D}_m$, $i \in \mathbb{N}$, then

$$\lim_{n\to\infty}\mathbf{W}_n = I \tag{4.19}$$

and

$$\lim_{n\to\infty}\mathbf{b}_n = 0. \tag{4.20}$$

Proof. The second inequality of condition (4.18) implies $\|\mathbf{P}_n\| \to 0$ as $n \to \infty$. Thus, using the first equation of condition (4.18), we conclude Eq. (4.19).

It remains to prove Eq. (4.20). Since the limit (3.11) exists for all $I_i \in \mathcal{D}_m$, we let $I_i = I$ for $i \ge 2$ to get that

$$\lim_{n\to\infty}\sum_{i=1}^{n}\left(\prod_{j=i+1}^{n}\mathbf{W}_j\right)\mathbf{b}_i \tag{4.21}$$

exists. By similar analysis as those in the proof of Theorem 4.3, we conclude that

$$\left\|\prod_{j=i+1}^{n}\mathbf{W}_j - \prod_{j=i+1}^{\infty}\mathbf{W}_j\right\| \le \left(\sum_{j=n+1}^{\infty}\|\mathbf{P}_j\|\right)\exp\left(\sum_{j=i+1}^{\infty}\|\mathbf{P}_j\|\right) \le C_1\left(\sum_{j=n+1}^{\infty}\|\mathbf{P}_j\|\right), \tag{4.22}$$

where

$$C_1 := \exp\left(\sum_{j=2}^{\infty}\|\mathbf{P}_j\|\right).$$

Noting that $\mathbf{b}_n$, $n \in \mathbb{N}$, are bounded, we may let

$$C_2 := \sup_{n\in\mathbb{N}}\|\mathbf{b}_n\| < +\infty. \tag{4.23}$$

Employing (4.22) and (4.23) yields the estimate

$$\left\|\sum_{i=1}^{n}\left(\prod_{j=i+1}^{n}\mathbf{W}_j\right)\mathbf{b}_i - \sum_{i=1}^{n}\left(\prod_{j=i+1}^{\infty}\mathbf{W}_j\right)\mathbf{b}_i\right\| \le \sum_{i=1}^{n}\left\|\prod_{j=i+1}^{n}\mathbf{W}_j - \prod_{j=i+1}^{\infty}\mathbf{W}_j\right\|\|\mathbf{b}_i\|$$
$$\le C_1C_2n\left(\sum_{j=n+1}^{\infty}\|\mathbf{P}_j\|\right).$$

By the third condition in (4.18) and the existence of the limit (4.21), we observe that

$$\lim_{n\to\infty}\sum_{i=1}^{n}\left(\prod_{j=i+1}^{\infty}\mathbf{W}_j\right)\mathbf{b}_i$$

also exists. It follows that

$$\lim_{i\to\infty}\left(\prod_{j=i+1}^{\infty}\mathbf{W}_j\right)\mathbf{b}_i = 0. \tag{4.24}$$

Notice that

$$\prod_{j=i+1}^{\infty}\mathbf{W}_j - I = \prod_{j=i+1}^{\infty}(I+\mathbf{P}_j) - I = \sum_{j=i+1}^{\infty}\mathbf{P}_j + \sum_{l=2}^{\infty}\sum_{i+1\le j_1<j_2<\cdots<j_l}\prod_{k=1}^{l}\mathbf{P}_{j_k}.$$

It follows from Lemma 4.2 with $a_n := \|\mathbf{P}_n\|$ that for i big enough,

$$\left\|\prod_{j=i+1}^{\infty}\mathbf{W}_j - I\right\| \le \sum_{j=i+1}^{\infty}\|\mathbf{P}_j\| + \sum_{l=2}^{\infty}\sum_{i+1\le j_1<j_2<\cdots<j_l}\prod_{k=1}^{l}\|\mathbf{P}_{j_k}\|$$
$$\le \left(\sum_{j=i+1}^{\infty}\|\mathbf{P}_j\|\right)\exp\left(\sum_{j=i+1}^{\infty}\|\mathbf{P}_j\|\right).$$

Therefore, for big enough i,

$$\left\|\prod_{j=i+1}^{\infty}\mathbf{W}_j - I\right\| < \frac{1}{2}.$$

By a classical result from function analysis ([28], page 193), we conclude that for big enough i,

$$\prod_{j=i+1}^{\infty}\mathbf{W}_j = I + \left(\prod_{j=i+1}^{\infty}\mathbf{W}_j - I\right)$$

is invertible and its inverse satisfies

$$\left\|\left(\prod_{j=i+1}^{\infty}\mathbf{W}_j\right)^{-1}\right\| \le \frac{1}{1-\left\|\prod_{j=i+1}^{\infty}\mathbf{W}_j - I\right\|} \le 2.$$

Consequently, for big enough i,

$$\|\mathbf{b}_i\| = \left\|\left(\prod_{j=i+1}^{\infty}\mathbf{W}_j\right)^{-1}\left(\prod_{j=i+1}^{\infty}\mathbf{W}_j\right)\mathbf{b}_i\right\| \le 2\left\|\left(\prod_{j=i+1}^{\infty}\mathbf{W}_j\right)\mathbf{b}_i\right\|,$$

which together with (4.24) ensures the validity of Eq. (4.20). □

The necessary conditions given in Theorem 4.4 for a ReLU network to converge provide mathematical guidelines for further construction of deep ReLU networks.

Our next task is to establish a useful sufficient condition guaranteeing the existence of limit (3.11).

Theorem 4.5. *Let $\|\cdot\|$ be a norm on $\mathbb{R}^m$ that satisfies (4.3) and $\|\cdot\|$ be its induced matrix norm. If*

$$\sum_{n=1}^{\infty}\|\mathbf{b}_n\| < +\infty, \tag{4.25}$$

$$\prod_{j=i}^{\infty}I_j\mathbf{W}_j \ \text{converges for every } i \ge 2, \tag{4.26}$$

and there exists a positive constant C such that

$$\prod_{j=i}^{n}\|\mathbf{W}_j\| \le C \text{ for all } 2 \le i \le n < +\infty, \tag{4.27}$$

then the limit (3.11) exists.

Proof. It suffices to show that

$$\mathbf{c}_n := \sum_{i=1}^{n}\Big(\prod_{j=i+1}^{n} I_j\mathbf{W}_j\Big)I_i\mathbf{b}_i, \quad n \in \mathbb{N}$$

forms a Cauchy sequence in $\mathbb{R}^m$. Let $\varepsilon > 0$ be arbitrary. By condition (4.25), there exists some $p \in \mathbb{N}$ such that

$$\sum_{i=p+1}^{\infty}\|\mathbf{b}_i\| < \varepsilon. \tag{4.28}$$

According to hypothesis (4.26), when $n' > n$ are big enough, it holds for all $i = 1, 2, \dots, p$ that

$$\Big\|\prod_{j=i+1}^{n'} I_j\mathbf{W}_j - \prod_{j=i+1}^{n} I_j\mathbf{W}_j\Big\| \le \varepsilon. \tag{4.29}$$

For such $n' > n > p$, we estimate $\|\mathbf{c}_{n'} - \mathbf{c}_n\|$. To this end, we let

$$\mathbf{d}_{n',n,p} := \sum_{i=1}^{p}\Big(\prod_{j=i+1}^{n'} I_j\mathbf{W}_j - \prod_{j=i+1}^{n} I_j\mathbf{W}_j\Big)I_i\mathbf{b}_i.$$

Then, it follows from condition (4.29) that for big enough $n' > n$,

$$\|\mathbf{d}_{n',n,p}\| \le \sum_{i=1}^{p}\Big\|\prod_{j=i+1}^{n'} I_j\mathbf{W}_j - \prod_{j=i+1}^{n} I_j\mathbf{W}_j\Big\|\|\mathbf{b}_i\| \le \varepsilon\sum_{i=1}^{p}\|\mathbf{b}_i\|. \tag{4.30}$$

Note that

$$\mathbf{c}_{n'} - \mathbf{c}_n = \mathbf{d}_{n',n,p} + \sum_{i=p+1}^{n'}\Big(\prod_{j=i+1}^{n'} I_j\mathbf{W}_j\Big)I_i\mathbf{b}_i + \sum_{i=p+1}^{n}\Big(\prod_{j=i+1}^{n} I_j\mathbf{W}_j\Big)I_i\mathbf{b}_i. \tag{4.31}$$

Employing (4.30), (4.5), (4.27), and (4.28), we have for big enough $n' > n$ that

$$\begin{aligned}\|\mathbf{c}_{n'} - \mathbf{c}_n\| &\le \|\mathbf{d}_{n',n,p}\| + \sum_{i=p+1}^{n'}\Big(\prod_{j=i+1}^{n'}\|\mathbf{W}_j\|\Big)\|\mathbf{b}_i\| + \sum_{i=p+1}^{n}\Big(\prod_{j=i+1}^{n}\|\mathbf{W}_j\|\Big)\|\mathbf{b}_i\| \\ &\le \varepsilon\sum_{i=1}^{p}\|\mathbf{b}_i\| + 2C\sum_{i=p+1}^{\infty}\|\mathbf{b}_i\| \\ &\le \varepsilon\Big(\sum_{i=1}^{p}\|\mathbf{b}_i\| + 2C\Big).\end{aligned}$$

This shows that $\mathbf{c}_n$ is a Cauchy sequence and thus it converges. □

We remark that when $\mathbf{W}_i, I_i$ all equal the identity matrix, limit (3.11) becomes

$$\lim_{n\to\infty}\sum_{i=1}^{n}\mathbf{b}_i.$$

Thus, condition (4.25) is almost necessary for the existence of limit (3.11). The other two conditions (4.26) and (4.27) are weaker than condition (4.14), as explained in the proof of Theorem 5.1 to be presented in the next section.

5. Sufficient conditions for convergence of ReLU networks

In this section, we present sufficient conditions for deep ReLU networks to converge pointwise by using results established in the previous two sections. Moreover, we demonstrate that these sufficient conditions provide mathematical interpretation to the well-known deep residual networks which have achieved remarkable success in image classification.

We now establish sufficient conditions on their weight matrices and bias vectors for deep ReLU networks to converge pointwise.

Theorem 5.1. *Let $\|\cdot\|$ be a norm on $\mathbb{R}^m$ that satisfies (4.3) and $\|\cdot\|$ be its induced matrix norm. If the weight matrices $\mathbf{W}_n$, $n \ge 2$, satisfy*

$$\mathbf{W}_n = I + \mathbf{P}_n, \quad n \ge 2, \quad \sum_{n=2}^{\infty}\|\mathbf{P}_n\| < +\infty \tag{5.1}$$

and the bias vectors $\mathbf{b}_i$, $i \in \mathbb{N}$, satisfy

$$\sum_{n=1}^{\infty}\|\mathbf{b}_n\| < +\infty, \tag{5.2}$$

then the ReLU neural networks $\mathcal{N}_n$ converge pointwise on $[0,1]^d$.

Proof. According to Theorem 3.6, it suffices to show under the given conditions of this theorem, limits (3.10) and (3.11) exist for all $I_n \in \mathcal{D}_m$, $n \in \mathbb{N}$.

Since the vector norm on $\mathbb{R}^m$ satisfies (4.3), its induced matrix norm satisfies conditions (4.4) and (4.5). By Theorem 4.3, condition (5.1) ensures that limit (3.10) exists for all $I_n \in \mathcal{D}_m$, $n \in \mathbb{N}$.

It remains to confirm that limit (3.11) exists for all $I_n \in \mathcal{D}_m$, $n \in \mathbb{N}$. By the proof of Theorem 4.3, condition (4.26) is satisfied when condition (5.1) is fulfilled. We can also verify by using properties of the exponential function that

$$\prod_{j=i}^{n}\|\mathbf{W}_j\| \le \prod_{j=i}^{n}(1 + \|\mathbf{P}_j\|) \le \prod_{j=i}^{n}\exp(\|\mathbf{P}_j\|) \le \exp\Big(\sum_{j=2}^{\infty}\|\mathbf{P}_j\|\Big),$$
$$2 \le i \le n < +\infty.$$

Therefore, condition (4.27) is also satisfied with the constant

$$C := \exp\Big(\sum_{j=2}^{\infty}\|\mathbf{P}_j\|\Big).$$

By Theorem 4.5, limit (3.11) exists for all $I_n \in \mathcal{D}_m$, $n \in \mathbb{N}$.

Finally, by part 2 of Theorem 3.6, we conclude that the ReLU deep network $\mathcal{N}_n$ converges pointwise on $[0,1]^d$ as n tends to infinity. □

We remark that under the conditions in Theorem 5.1 or Theorem 4.4, it holds

$$\lim_{n\to\infty}\mathbf{W}_n x + \mathbf{b}_n = x, \quad x \in \mathbb{R}^m,$$

which reveals that for deep layers, the linear function $\mathbf{W}_n x + \mathbf{b}_n$ will be close to the identity mapping. Thus the deep weight layers of a ReLU network apply gradual changes to the ultimate input–output relation determined by the network. This may justify the design strategy of the successful deep Residual Networks (ResNets) for image recognition [29, 30].

6. Experiments

In this section, we present two experiments to verify the main result Theorem 5.1 mathematically, and demonstrate the usefulness of the result in deep learning applications. For the first aim, we shall first randomly generate a sequence of weight matrices and bias vectors that satisfy the sufficient conditions in Theorem 5.1 and then check if the neural network will converge as the number of layers increases. For the second aim, we shall conduct experiments with very deep DNN in different strategies on the benchmark dataset MNIST. We shall see that the strategy that initializes the parameters according to the sufficient conditions in Theorem 5.1 makes the DNN easier to train and obtain better accuracy on the test set.

6.1. Mathematical verification

To verify Theorem 5.1 mathematically, we shall conduct multiple experiments with different input dimensions and hidden dimensions. For each experiment, we set the depth of network to be $n = 1000$ to see whether the neural networks $\{\mathcal{N}_i\}_{i=1}^{n}$ defined by (2.1) will be convergent. We utilize the following steps to randomly generate the

Table 6.1
Error $\|\mathcal{N}_{i+1} - \mathcal{N}_i\|_\infty$ for $d = 2$.

m \ layer index i	1	9	99	999
10	0.26796682	0.00902423	0.00010435	0.00000127
100	0.27893739	0.00955302	0.00010594	0.00000109
500	0.24848135	0.01025245	0.00009931	0.00000100

Table 6.2
Error $\|\mathcal{N}_{i+1} - \mathcal{N}_i\|_\infty$ for $d = 3$.

m \ layer index i	1	9	99	999
10	0.26884935	0.01346173	0.00011699	0.00000109
100	0.25358571	0.01112940	0.00010544	0.00000109
500	0.29031739	0.01011117	0.00010057	0.00000105

weight matrices and bias vectors that satisfy the sufficient conditions in Theorem 5.1.

1. Since the first weight matrix is not involved in the sufficient conditions, we directly generate a $m \times d$ matrix $\mathbf{W}_1$ with each element generated from the standard normal distribution.
2. To generate matrices that meet condition (5.1), we let

$$\mathbf{W}_i = \mathbf{I} + \mathbf{P}_i = \mathbf{I} + \frac{1}{i^2}\mathbf{M}_i \odot \mathbf{D}_i,\ 2 \le i \le n,$$

where $\odot$ denotes the Hadamard product (namely, component-wise product) of matrices, entries of $\mathbf{M}_i$ are randomly chosen as -1 or 1, and $\mathbf{D}_i$ is a randomly generated doubly stochastic matrix whose entries are first generated from the uniform distribution on $[0, 1]$ and then normalized so that the sum of each row and column equals 1.
The matrices $\mathbf{W}_i$ generated in the above manner satisfies the sufficient condition (5.1), as by the Riesz–Thorin interpolation theorem (see [31], page 200), one has for every $1 \le p \le +\infty$,

$$\sum_{i=2}^{n} \|\mathbf{P}_i\|_p = \sum_{i=1}^{n} \frac{1}{i^2}\|\mathbf{M}_i \odot \mathbf{D}_i\|_p \le \sum_{i=1}^{n} \frac{1}{i^2}\|\mathbf{M}_i \odot \mathbf{D}_i\|_1^{\frac{1}{p}}\|\mathbf{M}_i \odot \mathbf{D}_i\|_\infty^{1-\frac{1}{p}}$$
$$= \sum_{i=1}^{n} \frac{1}{i^2}\|\mathbf{D}_i\|_1^{\frac{1}{p}}\|\mathbf{D}_i\|_\infty^{1-\frac{1}{p}} = \sum_{i=1}^{n} \frac{1}{i^2} < \frac{\pi^2}{6}.$$

3. The bias vectors $\mathbf{b}_i$ are generated by

$$\mathbf{b}_i = \frac{\tilde{\mathbf{b}}_i}{i^2\|\tilde{\mathbf{b}}_i\|_\infty},\ 1 \le i \le n$$

where the entries of $\tilde{\mathbf{b}}_i$ are generated from the standard normal distribution. Condition (5.2) is also satisfied, as

$$\sum_{i=2}^{n} \|\mathbf{b}_i\|_p \le \sum_{i=1}^{n} m^{\frac{1}{p}}\|\mathbf{b}_i\|_\infty = m^{\frac{1}{p}}\sum_{i=1}^{n}\frac{1}{i^2} < m^{\frac{1}{p}}\frac{\pi^2}{6}.$$

With above preparations, we shall perform experiments for $(d, m) \in \{2, 3\} \times \{10, 100, 500\}$. In our implementation, we construct an equally-distributed grid for $[0, 1]^d$ with 100^d grid points, and then compute the maximal error $\|\mathcal{N}_{i+1} - \mathcal{N}_i\|_\infty$ between adjacent layers at the grid points. The results are tabulated in Tables 6.1 and 6.2 for $d = 2$ and $d = 3$, respectively. We also plot the error $\|\mathcal{N}_{i+1} - \mathcal{N}_i\|_\infty$ in Fig. 6.1. Both the tables and figures show that the error $\|\mathcal{N}_{i+1} - \mathcal{N}_i\|_\infty$ is dramatically decreasing to zero as the number of layers increases. Therefore, these experiments confirm that the sufficient conditions for convergence of DNN in Theorem 5.1 are mathematically correct.

6.2. Experiment on the MNIST dataset

We shall conduct experiments on the MNIST dataset to indicate that parameters of DNN initialized according the sufficient conditions in Theorem 5.1 may be able to accelerate the training process.

We shall train our models with Keras [32]. Recall that the structure of DNN discussed in this work could be illustrated as

$$x \in [0, 1]^d \xrightarrow[\sigma]{\mathbf{W}_1, \mathbf{b}_1} x^{(1)} \xrightarrow[\sigma]{\mathbf{W}_2, \mathbf{b}_2} x^{(2)} \to \cdots \to \xrightarrow[\sigma]{\mathbf{W}_n, \mathbf{b}_n} x^{(n)} \xrightarrow{\mathbf{W}_o, \mathbf{b}_o} y \in \mathbb{R}^{d'}.$$

input 1st layer 2nd layer nth layer output

We set the width $m = 784$ and the depth $n = 10, 100, 1000$. The input dimension $d = 784$ is given by the dataset. Also, we shall formulate the problem as regression. To this end, we set the output dimension $d' = 1$, and fix the parameters in the output layer by $\mathbf{W}_o = (\frac{1}{784}, \ldots, \frac{1}{784})$ and $\mathbf{b}_o = 0$. Notice that the prediction label will be determined by $\min(\lfloor y + \frac{1}{2} \rfloor, 9)$. The network structure is summarized in Table 6.3.

The experiment is to illustrate the potential usefulness of our theoretical results to deep learning. Specifically, we propose to initialize the weight matrices and bias vectors of a deep neural network according to the sufficient conditions in Theorem 5.1. In this manner, the network tends to converge faster so that the training process can be accelerated. We illustrate the effectiveness of this initialization strategy on the benchmark dateset MNIST below.

For $1 \le i \le n$, we denote by $\mathbf{P}_i$ and $\hat{\mathbf{b}}_i$ the weight matrix and bias vector at the ith layer, which will be determined with two different strategies as follows.

Strategy 1 (*proposed initialization strategy according to* Theorem 5.1): We design the parameters according to the sufficient conditions established in the paper by

$$\mathbf{W}_i = \mathbf{I} + \frac{\mathbf{P}_i}{i^2},\ \mathbf{b}_i = \frac{\hat{\mathbf{b}}_i}{i^2},\ 1 \le i \le n,$$

where $\|\mathbf{P}_i\|_\infty \le 784$ and $\|\hat{\mathbf{b}}_i\|_\infty \le 784$ for all $1 \le i \le n$.

Strategy 2 (a usual initialization strategy in deep learning): The second strategy is the usual one without any constraints on the parameters

$$\mathbf{W}_i = \mathbf{P}_i,\ \mathbf{b}_i = \hat{\mathbf{b}}_i,\ 1 \le i \le n.$$

Our purpose is to see if the first strategy according to our theorem will lead to a more stable training process, a lower loss and a higher accuracy. To this end, for each strategy, we shall train the network for 50 epochs with Mean Squared Error (MSE) as the loss function. For a fair performance comparison of the two strategies, we shall evaluate the two strategies with different parameter initialization methods for the parameters in weight matrices, while all biases will be initialized with zeros. All the initialization methods utilized in our experiments are described in details in Table 6.4. In addition, we train the network with the Adam optimizer in both strategies. Adam is a replacement optimization algorithm for stochastic gradient descent for training deep learning models. And we set the learning rates as 10^{-3} for all cases with strategy 1, while the learning rates for strategy 2 are set as 10^{-3}, 10^{-4}, 10^{-6} for 10-layer, 100-layer, 1000-layer network, respectively. We use such learning rates since several cases in strategy 2 will be unstable

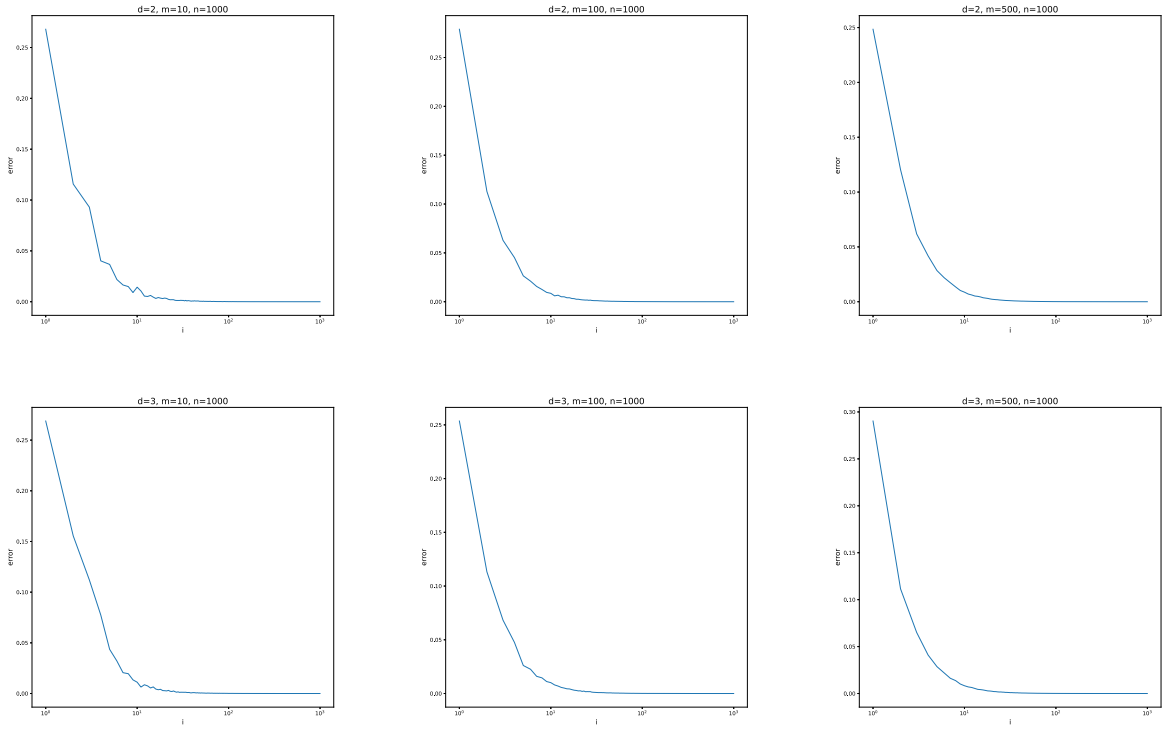


Fig. 6.1. Plot of the error $\|\mathcal{N}_{i+1} - \mathcal{N}_i\|_\infty$ for various dimensions and widths.

Table 6.3
The DNN Architecture.

weight dimension	bias dimension	description
784×784	784	10, 100, 1000 layers with ReLU activation
–	–	Output layer (Linear, Average, Not trainable)

Table 6.4
Descriptions of initialization methods.

Initialization method	Description
He normal [33]	Each trainable parameter in weight matrices is generated by a truncated normal distribution with mean 0 and standard deviation $\frac{\sqrt{2}}{28}$ where the values more than two standard deviations from the mean are discarded and redrawn.
He uniform [33]	Each trainable parameter in weight matrices is generated by a uniform distribution within $\left[-\frac{\sqrt{6}}{28}, \frac{\sqrt{6}}{28}\right]$.
Zeros/Identity	For strategy 1, each trainable parameter in weight matrices is initially zero which would make $\mathbf{W}_i = \mathbf{I}$ for all $1 \le i \le n$. For strategy 2, each $\mathbf{P}_i$ is initially identity matrix, which would also make $\mathbf{W}_i = \mathbf{I}$ for all $1 \le i \le n$.

with 10^{-3} learning rate when n is large, and such issue would not happen in strategy 1.

The detailed performance of the two strategies is shown in Table 6.5. One could see that the best MSE on training set, the best MSE on test set and the best accuracy on training set are all obtained with strategy 1. Although the best accuracy on test set is obtained with strategy 2, strategy 1 is more robust for training under different settings with competitive performance. Note that the performance of strategy 2 with a 100-layer network is significantly poorer and more biased than other cases, which may be caused by the gradient vanishing phenomenon.

In particular, for the 1000-layer networks, the MSE on the training set and test set of each strategy in the training process are also plotted in Fig. 6.2, and listed in Tables 6.6 and 6.7 respectively for every 5 epochs. We observe from these tables and plots that the initialization strategy of parameters according to the sufficient conditions (5.1) and (5.2) indeed leads to a faster training process. This is predicted by the mathematical result Theorem 5.1. It also leads to a better performance on both the training and test sets, which is a surprise.

Finally, we shall check whether the sufficient conditions in Theorem 5.1 are satisfied in the training processes of each strategy. We shall only check for the 1000-layer networks since they are sufficiently deep. Recalling the Riesz–Thorin interpolation theorem, we shall compute $\sum_{i=1}^n \max(\|\mathbf{P}_i\|_1, \|\mathbf{P}_i\|_\infty)/i^2$ and $\sum_{i=1}^n \|\mathbf{b}_i\|_\infty/i^2$ for strategy 1, and compute $\sum_{i=1}^n \max(\|\mathbf{P}_i - \mathbf{I}\|_1, \|\mathbf{P}_i - \mathbf{I}\|_\infty)$ and $\sum_{i=1}^n \|\mathbf{b}_i\|_\infty$ for strategy 2. These two quantities are plotted in Figs. 6.3 and 6.4, respectively. One sees that the trained network under Strategy 1 does satisfy the sufficient conditions in Theorem 5.1, while that under Strategy 2 does not.

7. Conclusion

Deep learning based on deep neural networks (DNNs) has achieved great successes in machine learning. DNNs constitute a highly efficient system to represent high-dimensional complicated functions. A DNN

Table 6.5
Performances of different strategies.

Strategy	Initialization method	Depth	MSE on training set	MSE on test set	Accuracy on training set	Accuracy on test set
1	He normal	10	0.0281	0.3934	95.03%	91.19%
		100	0.0234	0.2962	99.75%	96.56%
		1000	0.0055	0.3021	**99.97%**	96.77%
	He uniform	10	0.0174	0.3114	99.83%	96.5%
		100	**0.0039**	0.3047	99.96%	96.54%
		1000	0.0081	0.3030	99.40%	94.68%
	Zeros	10	0.0290	0.3403	99.24%	95.66%
		100	0.0195	0.2987	99.57%	96.61%
		1000	0.0160	**0.2897**	99.54%	96.74%
2	He normal	10	0.0431	0.3063	99.18%	97.15%
		100	8.370	8.4172	9.74%	9.82%
		1000	0.2792	1.2793	87.30%	79.72%
	He uniform	10	0.0608	0.2915	99.01%	96.96%
		100	0.0782	0.3350	98.35%	96.41%
		1000	0.2716	1.3350	86.08%	77.44%
	Identity	10	0.0716	0.3471	98.56%	96.54%
		100	0.0303	0.3298	99.23%	96.40%
		1000	0.0092	0.3129	99.85%	**97.34%**

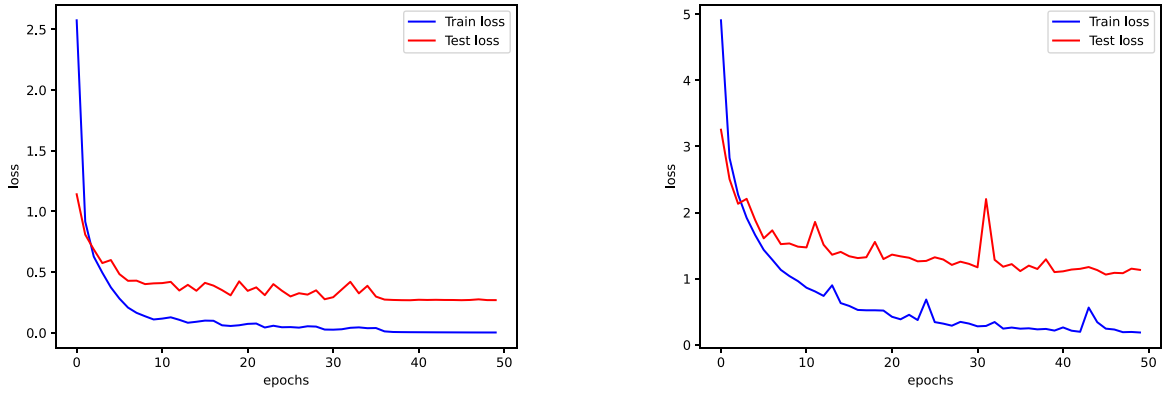


Fig. 6.2. Convergence plot of strategy 1 (left) and strategy 2 (right).

Table 6.6
Best MSE among all initialization methods of each strategy on the training set for every 5 epochs when $n = 1000$.

Strategy \ Epoch	5	10	15	20	25	30	35	40	45	50
1	0.3746	0.1103	0.0914	0.0626	0.0462	0.0272	0.0384	0.0056	0.0039	0.0030
2	1.6608	0.9643	0.6321	0.5191	0.6849	0.3226	0.2621	0.2164	0.3436	0.1875

Table 6.7
Best MSE among all initialization methods on the test set for every 5 epochs when $n = 1000$.

Strategy \ Epoch	5	10	15	20	25	30	35	40	45	50
1	0.5997	0.4077	0.3471	0.4240	0.3472	0.2770	0.3874	0.2688	0.2708	0.2697
2	1.8877	1.4853	1.4060	1.2980	1.2699	1.2253	1.2209	1.1008	1.1311	1.1342

in applications usually possesses a massive amount of parameters. In practice, a DNN is trained on given data and is considered to be convergent once a certain accuracy is attained. However, it is hard to give explanation to how such a nonlinear system with so many parameters achieves convergence. In other words, people did not know what mathematical conditions the parameters have to satisfy in order for the DNN to converge.

In this paper, we establish mathematical sufficient conditions on the parameters of a DNN to ensure that it converges to a well-defined function as the number of layers increases to infinity. For a sequence of ReLU neural networks defined by

$$\mathcal{N}_n(x) = \left(\bigodot_{i=1}^{n} \sigma(\mathbf{W}_i \cdot + \mathbf{b}_i) \right)(x), \quad x \in [0,1]^d,$$

the established sufficient conditions ensuring the convergence of $\mathcal{N}_n$ as n tends to infinity are convergence of two infinite series:

1. The weight matrices $\mathbf{W}_n$ satisfy

$$\mathbf{W}_n = I + \mathbf{P}_n, \quad n \geq 2, \quad \sum_{n=2}^{\infty} \|\mathbf{P}_n\| < +\infty.$$

2. The bias vectors $\mathbf{b}_n$ satisfy

$$\sum_{n=1}^{\infty} \|\mathbf{b}_n\| < +\infty.$$

One sees that the conditions are in simple mathematical form and hence easy to comprehend and apply. The results provide insightful understanding of the convergence of deep neural networks. As far as

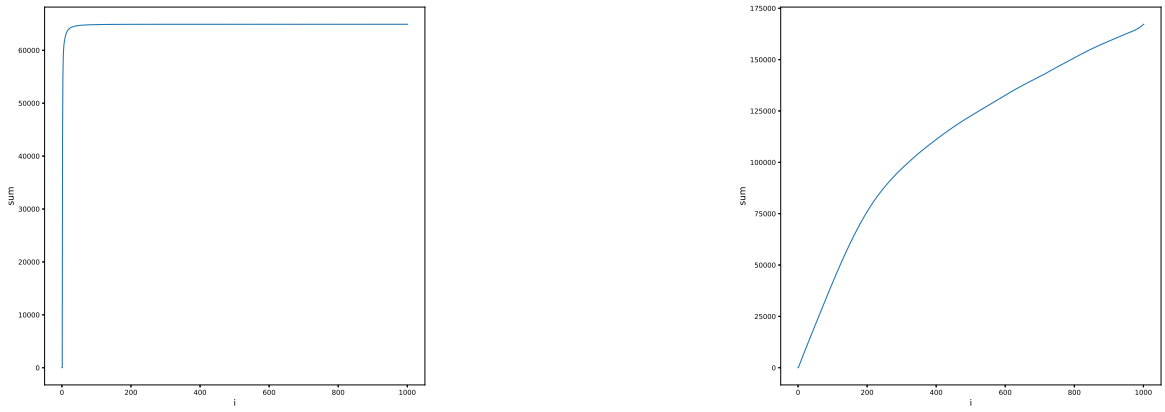


Fig. 6.3. Sufficient condition for weight matrices of the trained 1000-layer networks with Strategy 1 under zero initialization (left) and Strategy 2 under identity initialization(right).

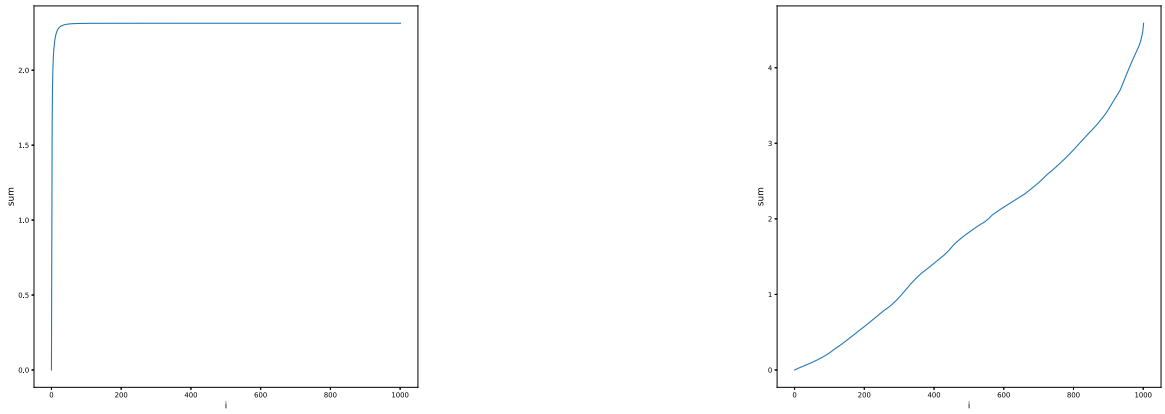


Fig. 6.4. Sufficient condition for bias vectors of the trained 1000-layer networks with Strategy 1 under zero initialization (left) and Strategy 2 under identity initialization(right).

we know, such sufficient conditions on the parameters of a DNN to ensure its convergence are new in the literature. For their potential applications to deep learning, we propose to initialize the weight matrices and bias vectors of a deep neural network according to the sufficient conditions above. By the mathematical analysis in the paper, a DNN initialized in this way tends to converge faster so that the training process can be accelerated.

The sufficient conditions are verified by mathematical experiments on randomly generated weight matrices and bias vectors. Experiments on the MNIST dataset are also conducted to illustrate the proposed initialization strategy. Specifically, it is shown in the experiments that parameters of a DNN initialized according to the sufficient conditions may lead to a faster training process.

CRediT authorship contribution statement

Yuesheng Xu: Discussed to form the research question and ideas of the paper, Discussed, Checked, Corrected the proofs, Improved the presentation of the paper, Discussed the results and contributed to the final manuscript. **Haizhang Zhang:** Discussed to form the research question and ideas of the paper, Conceived and proved the main results of the paper, Writing – original draft, Discussed the results and contributed to the final manuscript.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

Acknowledgments

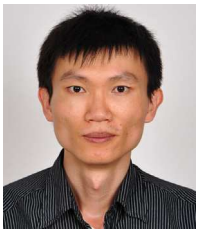
We would like to express gratitude to the reviewers for comments and suggestions that help improve the manuscript. We would also like to thank Wentao Huang for helping with the numerical experiments.

References

[1] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (7553) (2015) 436–444, 2015.
[2] I. Goodfellow, Y. Bengio, A. Courville, Deep Learning, MIT Press, Cambridge, 2016.
[3] R. DeVore, B. Hanin, G. Petrova, Neural network approximation, Acta Numerica 30 (2021) 327–444.
[4] D. Elbrächter, D. Perekrestenko, P. Grohs, H. Bölcskei, Deep neural network approximation theory, IEEE Trans. Inform. Theory 67 (2021) 2581–2623.
[5] T. Poggio, H. Mhaskar, L. Rosasco, B. Miranda, Q. Liao, Why and when can deep-but not shallow-networks avoid the curse of dimensionality: A review, Int. J. Autom. Comput. 14 (2017) 503–519.
[6] H. Montanelli, Q. Du, New error bounds for deep networks using sparse grids, SIAM J. Math. Data Sci. 1 (1) (2019) 78–92.
[7] D. Yarotsky, Error bounds for approximations with deep relu networks, Neural Netw. 94 (2017) 103–114.
[8] H. Montanelli, H. Yang, Error bounds for deep ReLU networks using the Kolmogorov–Arnold superposition theorem, Neural Netw. 129 (2020) 1–6.
[9] W. E, Q. Wang, Exponential convergence of the deep neural network approximation for analytic functions, Sci. China Math. 61 (10) (2018) 1733–1740.
[10] D.X. Zhou, Universality of deep convolutional neural networks, Appl. Comput. Harmon. Anal. 48 (2) (2020) 787–794.

[11] Z. Shen, H. Yang, S. Zhang, Deep network approximation characterized by number of neurons, Commun. Comput. Phys. 28 (5) (2020) 1768–1811.
[12] Z. Shen, H. Yang, S. Zhang, Deep network with approximation error being reciprocal of width to power of square root of depth, Neural Comput. 33 (4) (2021) 1005–1036.
[13] Z. Shen, H. Yang, S. Zhang, Optimal approximation rate of ReLU networks in terms of width and depth, J. Math. Pures Appl. 157 (2022) 101–135.
[14] I. Daubechies, R. DeVore, S. Foucart, B. Hanin, G. Petrova, Nonlinear approximation and (deep) ReLU networks, Constr. Approx. 55 (2022) 127–172.
[15] Y. Wang, A mathematical introduction to generative adversarial nets (GAN), 2020, arXiv:2009.00169.
[16] P.L. Combettes, J.-C. Pesquet, Lipschitz certificates for layered network structures driven by averaged activation operators, SIAM J. Math. Data Sci. 2 (2) (2020) 529–557.
[17] M. Hasannasab, J. Hertrich, S. Neumayer, G. Plonka, S. Setzer, G. Steidl, Parseval proximal neural networks, J. Fourier Anal. Appl. 26 (4) (2020) 31, Paper No. 59.
[18] K. Scaman, A. Virmaux, Lipschitz regularity of deep neural networks: analysis and efficient estimation, in: 32nd Conference on Neural Information Processing Systems, NeurIPS 2018, Montréal, Canada.
[19] D. Zou, R. Balan, M. Singh, On Lipschitz bounds of general convolutional neural networks, IEEE Trans. Inform. Theory 66 (3) (2020) 1738–1759.
[20] B. Hanin, M. Nica, Finite depth and width corrections to the neural tangent kernel, 2019, arXiv:1909.05989.
[21] A. Jacot, F. Gabriel, C. Hongler, Neural tangent kernel: convergence and generalization in neural networks, in: 32nd Conference on Neural Information Processing Systems, NeurIPS 2018, Montréal, Canada.
[22] Q. Nguyen, M. Mondelli, G.F. Montufar, Tight bounds on the smallest eigenvalue of the neural tangent kernel for deep ReLU networks, in: Proceedings of the 38th International Conference on Machine Learning, PMLR 139, 2021, pp. 8119–8129.
[23] E. Stein, R. Shakarchi, Fourier Analysis. An Introduction, Princeton University Press, Princeton, NJ, 2003.
[24] I. Daubechies, Ten Lectures on Wavelets, SIAM, Philadelphia, 1992.
[25] T. Zaslavsky, Facing up to arrangements: face-count formulas for partitions of space by hyperplanes, Mem. Amer. Math. Soc. 1 (1) (1975) no. 154.
[26] J.H.M. Wedderburn, Lectures on Matrices, Dover, New York, 1964.
[27] M. Artzrouni, On the convergence of infinite products of matrices, Linear Algebra Appl. 74 (1986) 11–21.
[28] P.D. Lax, Functional Analysis, Wiley-Interscience, New York, 2002.
[29] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2016, pp. 770–778.
[30] K. He, X. Zhang, S. Ren, J. Sun, Identity mappings in deep residual networks, in: B. Leibe, J. Matas, N. Sebe, M. Welling (Eds.), Computer Vision – ECCV 2016, in: Lecture Notes in Computer Science, vol. 9908, Springer, Cham.
[31] G.B. Folland, Real Analysis: Modern Techniques and their Applications, John Wiley & Sons, 1999, p. 40.
[32] F. Chollet, et al., Keras, 2015, Available at: https://github.com/fchollet/keras.
[33] K. He, X. Zhang, S. Ren, J. Sun, Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification, in: 2015 IEEE International Conference on Computer Vision, ICCV, pp. 1026–1034.

Yuesheng Xu received B.S. and M.S. degrees from Zhongshan (Sun Yat-sen) University, Guangdong, China, in 1982 and 1985, respectively, and a Ph.D. degree from Old Dominion University, Norfolk, VA, in 1989. He is currently a professor of data science and mathematics at Old Dominion University, Norfolk, VA. He was the Eberly Chair Professor of Mathematics at West Virginia University from 2001 to 2003, Professor of Mathematics at Syracuse University from 2003 to 2013, and Guohua Chair Professor of Mathematics at Sun Yat-sen University from 2009 to 2017. He was the Managing Editor of Advances in Computational Mathematics from 2009–2012. Prof. Xu's research interests include numerical analysis, applied harmonic analysis, image and signal processing, and machine learning. His research was supported by US National Science Foundation, DoD, DoE, NASA, NIH and Natural Science Foundation of China.



Haizhang Zhang received the B.S. degree in mathematics and applied mathematics from Beijing Normal University in 2003, the M.S. degree in computational mathematics from the Chinese Academy of Sciences in 2006, and the Ph.D. degree in mathematics from Syracuse University in 2009. From June 2009 to May 2010, he was a postdoctoral research fellow at University of Michigan, Ann Arbor. Since June 2010, he has been a Professor with Sun Yat-sen University. Prof. Zhang's research interests include applied and computational harmonic analysis, machine learning, sampling theory, and time–frequency analysis.