applied sciences

MDPI

Article

ReLU Network with Bounded Width Is a Universal Approximator in View of an Approximate Identity

Sunghwan Moon

Department of Mathematics, Kyungpook National University, Daegu 41566, Korea; sunghwan.moon@knu.ac.kr

Abstract: Deep neural networks have shown very successful performance in a wide range of tasks, but a theory of why they work so well is in the early stage. Recently, the expressive power of neural networks, important for understanding deep learning, has received considerable attention. Classic results, provided by Cybenko, Barron, etc., state that a network with a single hidden layer and suitable activation functions is a universal approximator. A few years ago, one started to study how width affects the expressiveness of neural networks, i.e., a universal approximation theorem for a deep neural network with a Rectified Linear Unit (ReLU) activation function and bounded width. Here, we show how any continuous function on a compact set of $\mathbb{R}^{n_{in}}, n_{in} \in \mathbb{N}$ can be approximated by a ReLU network having hidden layers with at most $n_{in} + 5$ nodes in view of an approximate identity.

Keywords: deep neural nets; ReLU network; universal approximation theory; a feed-forward neural network

Citation: Moon, S. ReLU Network with Bounded Width Is a Universal Approximator in View of an Approximate Identity. *Appl. Sci.* **2021**, *11*, 427. https://doi.org/doi:10.3390/app11010427

Received: 19 November 2020
Accepted: 31 December 2020
Published: 4 January 2021

1. Introduction

Over the past several years, deep neural networks have achieved state-of-the-art performance in a wide range of tasks such as image recognition/segmentation and machine translation (see the review article [1] and recent book [2] for more background). In spite of their outstanding successes, many aspects of why they work so well are still not well understood. Some works to date have focused on the problem of explaining and quantifying the expressivity of a deep neural network [3–5].

This line of research about the expressive power of neural networks, its ability to approximate functions, dates back to at least the work of Cybenko [6], in which a fully connected neural network with a sigmoid activation function and one single hidden layer can approximate any continuous univariate function on a bounded domain with an arbitrarily small error. Barron [7], Hornik et al. [8], and Funahashi [9] generalized the sigmoid function to a large class of activation functions to get universal approximation. However, they do not consider the number of nodes of a hidden layer; actually, the number of nodes of a hidden layer can be exponentially increased in the input dimension.

Because the outstanding performance of deep neural networks has recently been exhibited, there is a lot of literature concerning their expressive power theoretically. In 2011, Delalleau and Bengio [10] gave a family of functions which can be represented much more efficiently with a deep network than with a neural network with one hidden layer. Mhaskar and Poggio [4] provided conditions under which deep convolutional neural networks perform much better in function approximation than a neural network with one hidden layer. Eldan and Shamir [11] constructed a three-layer network which cannot be realized by any 2-layer if the number of nodes of an output layer is no more than an exponential bound.

Actually, the Rectified Linear Units (ReLU) activation function is the most popular choice in practical use of the neural network [12]. In this reason, most of the recent results on the universal approximation theory is about the ReLU network [5,13–20]. Cohen et al. [13] provided the deep convolutional neural network with the ReLU activation function that

cannot be realized by a shallow network if the number of nodes of its hidden layer is no more than an exponential bound.

Lu et al. [14] presented a universal approximation theorem for deep neural networks with ReLU activation functions and hidden layers with a bounded width in 2017, since the expressive power of depth in ReLU networks with a bounded width has received a lot of attention. Hanin proved universal approximation theorem using convex function theory in [15,16], and Lin and Jegelka showed this theorem for residual networks with one-neuron hidden layers in [19].

Ohn and Kim [18] showed that the ReLU activation function can be represented by a linear combination of piecewise linear activation function and, using this fact, extended many results in [5,17,20] to any continuous piecewise linear activation function. However, their results are not about the bounded width.

Here, we study a universal approximation theorem for ReLU networks in view of approximate identity by constructing a network having $n_{in}k^{n_{in}} + 1$ hidden layers with width of at most $n_{in} + 5$. As a measure of approximation error, we adopt the supremum norm and L^p distance.

The remainder of the paper is organized as follows: in the next subsection, we introduce some notations. Sections 2 and 3 are devoted to our theorem statement and its proof, respectively. Finally, Section 4 offers a conclusion.

Notations

Let $\mathbb{Z}_k = \{1, 2, \cdots, k\}$ and $\mathbf{x} = (x_1, x_2, \cdots, x_n) \in \mathbb{R}^n$ be the input. The network architecture $(l, \mathbf{n})$ consists of several hidden layers $l \in \mathbb{N}$ and a width vector $\mathbf{n} = (n_0, n_1, \cdots, n_{l+1}) \in \mathbb{N}^{l+2}, n_0 = n_{in}$ [20]. A neural network with network architecture $(l, \mathbf{n})$ is then any function of the form

$$f_{(l,\mathbf{n})} : \mathbb{R}^{n_0} \to \mathbb{R}^{n_{l+1}},$$
$$f_{(l,\mathbf{n})}(\mathbf{x}) = L^{l+1} \circ \mathrm{ReLU} \circ L^l \circ \mathrm{ReLU} \circ \cdots \circ L^2 \circ \mathrm{ReLU} \circ L^1(\mathbf{x}),$$

where

$$L^i : \mathbb{R}^{n_{i-1}} \to \mathbb{R}^{n_i} \quad i = 1, \cdots, l+1,$$

and $\mathrm{ReLU}(x) = \max\{x, 0\}$,

$$\mathrm{ReLU}(x_1, x_2, \cdots, x_m) = (\mathrm{ReLU}(x_1), \mathrm{ReLU}(x_2), \cdots, \mathrm{ReLU}(x_m)).$$

Here, we set $n_{l+1} = 1$. To this end, we define the space of network functions with the given network architecture

$$\mathcal{F}(l, \mathbf{n}) := \{f_{(l,\mathbf{n})} : f_{(l,\mathbf{n})}(\mathbf{x}) = L^{l+1} \circ \cdots \circ \mathrm{ReLU} \circ L^1(\mathbf{x}), (l, \mathbf{n}) \in \mathbb{N} \times \mathbb{N}^{l+2}\}.$$

Our network of interest is 'a feedforward neural network' which is an artificial neural network wherein connections between the nodes do not form a cycle. That is, in this network, the information moves in a forward direction, from the input nodes to the output nodes through the hidden nodes (if any).

For later discussion, we introduce the positive part $f_+ = \max(f, 0)$ and negative parts $f_- = \max(-f, 0)$ of a function f. Then, $f_+ - f_-$ is equal to f.

2. Main Result

Here is our universal approximation theorem for a ReLU network with a bounded width:

Theorem 1. *For any $M > 0$ and $n_{in} \in \mathbb{N}$, let $f : [-M, M]^{n_{in}} \to \mathbb{R}$ be a continuous function. For $\epsilon > 0$, there exists $k \in \mathbb{N}$ such that $f_{(l,\mathbf{n})} \in \mathcal{F}(n_{in}k^{n_{in}} + 1, n_{in}, n_{in} + 5, \cdots, n_{in} + 5, 2, 1)$ satisfies*

$$\sup_{\mathbf{x} \in [-M,M]^n} |f(\mathbf{x}) - f_{(l,\mathbf{n})}(\mathbf{x})| < \epsilon.$$

To prove this theorem, we explicitly construct a ReLU network with a bounded width in Section 3.2. In our construction, each of the first $n_{in}+3$ nodes in each hidden layer is connected to only one node of the previous layer, and each, only for two $n_{in}+4$ and $n_{in}+5$ nodes in each hidden layer, is connected to less than four nodes of the previous layer (for more details, see Section 3.2). This means that the non-fully connected ReLU network can be working.

We can show that there is a feed-forward neural network that approximates any $f \in L^p(\mathbb{R}^{n_{in}})$:

Corollary 1. *Let $n_{in} \in \mathbb{N}$ and $f \in L^p(\mathbb{R}^{n_{in}}), 1 \leq p < \infty$. For $\epsilon > 0$, there exists $k \in \mathbb{N}$ such that $f_{(l,\mathbf{n})} \in \mathcal{F}(n_{in}k^{n_{in}}+1, n_{in}, n_{in}+5, \cdots, n_{in}+5, 2, 1)$ satisfies*

$$||f(\mathbf{x}) - f_{(l,\mathbf{n})}(\mathbf{x})||_p < \epsilon.$$

Proof. For any $\epsilon > 0$, there is a continuous function f_c on $\mathbb{R}^{n_{in}}$ with compact support in $[-M, M]^{n_{in}}$ such that

$$\int_{\mathbb{R}^{n_{in}}} |f(\mathbf{x}) - f_c(\mathbf{x})|^p d\mathbf{x} < \frac{\epsilon^p}{2^p}.$$

Using Theorem 1, we can find $k \in \mathbb{N}$ such that a ReLU neural network architecture $(l, \mathbf{n})$ with a represented function $f_{(l,\mathbf{n})}$ satisfying

$$\sup_{\mathbf{x}\in[-M,M]^{n_{in}}} |f_c(\mathbf{x}) - f_{(l,\mathbf{n})}(\mathbf{x})| < \frac{\epsilon}{2^{n_{in}+1}M^{n_{in}}}.$$

Therefore, we have

$$||f(\mathbf{x}) - f_{(l,\mathbf{n})}(\mathbf{x})||_p \leq ||f(\mathbf{x}) - f_c(\mathbf{x})||_p + ||f_c(\mathbf{x}) - f_{(l,\mathbf{n})}(\mathbf{x})||_p < \epsilon.$$

□

3. Proof of Theorem 1

To convey our idea more clearly, we first construct a ReLU network with $n_{in} = 1$ and $k+1$ hidden layers with width at most 6. Then, we construct a ReLU network with general n_{in}.

3.1. One-Dimensional, Input

Let $L^{1,temp} : \mathbb{R} \to \mathbb{R}^3$ and $L^{2,temp} : \mathbb{R}^3 \to \mathbb{R}$ with $L^{1,temp}(x) = (x-1, x, x+1)$ and $L^{2,temp}(x_1, x_2, x_3) = x_1 - 2x_2 + x_3$. Then, the function $f_{(1,1,3,1)}$ represented by $L^{2,temp} \circ \mathrm{ReLU} \circ L^{1,temp}$ is

$$\begin{aligned} f_{(1,1,3,1)}(\mathbf{x}) &= L^{2,temp} \circ \mathrm{ReLU} \circ L^{1,temp}(x) \\ &= \mathrm{ReLU}(x-1) - 2\,\mathrm{ReLU}(x) + \mathrm{ReLU}(x+1) \\ &= \begin{cases} 0 & \text{if} \quad |x| > 1, \\ 1-|x| & \text{if} \quad -1 < x < 1. \end{cases} \end{aligned}$$

We notice that, for a (B-spline) function $g(x) = L^{2,temp} \circ \mathrm{ReLU} \circ L^{1,temp}(x)$ and $\lambda > 0$, $g_\lambda(x) = \lambda^{-1} g(\lambda^{-1}x)$ is an approximate identity: for any continuous function $f(x)$ on $[-M, M]$, $f * g_\lambda$ converges to $f(x)$ uniformly. For any $\epsilon > 0$, there is $\lambda > 0$ such that $\sup_{-M\leq x\leq M} |f * g_\lambda(x) - f(x)| < \epsilon/2$. By mensuration by division, there is $k \in \mathbb{N}$ such that

$$\sup_{-M\leq x\leq M} \left| f * g_\lambda(x) - \sum_{j=1}^{k} \frac{2M}{k} f\left(-M + \frac{2Mj}{k}\right) g_\lambda\left(x - \left(-M + \frac{2Mj}{k}\right)\right) \right| < \epsilon/2$$

and thus

$$\sup_{-M\le x\le M}\left| f(x)-\sum_{j=1}^{k}\frac{2M}{k}f\left(-M+\frac{2Mj}{k}\right)g_\lambda\left(x-\left(-M+\frac{2Mj}{k}\right)\right)\right|<\epsilon.$$

Now, for a fixed $k\in\mathbb{N}$, we can make a network architecture $(l,\mathbf{n})=(k+1,1,6,6,\cdots,6,2,1)$ with ReLU activations, input, and output dimensions 1, and $k+1$ hidden layer width at most 6, such that

$$\sup_{-M\le x\le M}\left|f(x)-f_{(l,\mathbf{n})}(x)\right|<\epsilon.$$

Let $L^{1,1}:\mathbb{R}\to\mathbb{R}^6$, $L^{1,j}:\mathbb{R}^6\to\mathbb{R}^6, j=2,\cdots,k$, $L^{k+1}:\mathbb{R}^6\to\mathbb{R}^2$, and $L^{k+2}:\mathbb{R}^2\to\mathbb{R}$ with

$$L^{1,1}(x)=\begin{pmatrix} x+M\\ \lambda^{-1}(x-(-M+2M/k))-1\\ \lambda^{-1}(x-(-M+2M/k))\\ \lambda^{-1}(x-(-M+2M/k))+1\\ 0\\ 0\end{pmatrix},$$

$$L^{1,j}(\mathbf{x})=\begin{pmatrix} x_1\\ \lambda^{-1}(x_1-M-(-M+\frac{2Mj}{k}))-1\\ \lambda^{-1}(x_1-M-(-M+\frac{2Mj}{k}))\\ \lambda^{-1}(x_1-M-(-M+\frac{2Mj}{k}))+1\\ x_5+\frac{2M}{k\lambda}f_+(-M+\frac{2M(j-1)}{k})(x_2-2x_3+x_4)\\ x_6+\frac{2M}{k\lambda}f_-(-M+\frac{2M(j-1)}{k})(x_2-2x_3+x_4)\end{pmatrix}\quad\text{for}\quad j=2,\cdots,k,$$

$$L^{k+1}(\mathbf{x})=\begin{pmatrix} x_5+\frac{M}{k\lambda}f_+(M)(x_2-2x_3+x_4)\\ x_6+\frac{M}{k\lambda}f_-(M)(x_2-2x_3+x_4)\end{pmatrix},\quad\text{and}$$

$$L^{k+2}(\mathbf{x})=(x_1-x_2).$$

(The index 1 in our index $(1,j)$ is unnecessary, but it is included because it suggests how to generalize to input of arbitrary dimension.) We note that

$$L^{1,2}\circ\mathrm{ReLU}\circ L^{1,1}(x)=\begin{pmatrix} x+M\\ \lambda^{-1}(x-(-M+\frac{2M}{k}))-1\\ \lambda^{-1}(x-(-M+\frac{2M}{k}))\\ \lambda^{-1}(x-(-M+\frac{2M}{k}))+1\\ \frac{2M}{k\lambda}f_+(-M+\frac{2M}{k})g(\lambda^{-1}x-\lambda^{-1}(-M+\frac{2M}{k}))\\ \frac{2M}{k\lambda}f_-(-M+\frac{2M}{k})g(\lambda^{-1}x-\lambda^{-1}(-M+\frac{2M}{k}))\end{pmatrix}.$$

and the fifth and sixth components of $L^{1,j}$ are non-negative. Since for a non-negative function $h : \mathbb{R} \to \mathbb{R}$, $\mathrm{ReLU}(h) = h$, we have for $\alpha \geq 0$,

$$\mathrm{ReLU}\left(h(x) + \alpha g_\lambda\left(x - \left(-M + \frac{2Mj}{k}\right)\right)\right) = h(x) + \alpha g_\lambda\left(x - \left(-M + \frac{2Mj}{k}\right)\right)$$

and thus (inductively) we have

$$\begin{aligned}
&L^{k+2} \circ \cdots \circ \mathrm{ReLU} \circ L^{1,1}(x) \\
&= \sum_{j=1}^{k} \frac{2M}{k} f_+\left(-M + \frac{2Mj}{k}\right) g_\lambda\left(x - \left(-M + \frac{2Mj}{k}\right)\right) \\
&\quad - \sum_{j=1}^{k} \frac{2M}{k} f_-\left(-M + \frac{2Mj}{k}\right) g_\lambda\left(x - \left(-M + \frac{2Mj}{k}\right)\right) \\
&= \sum_{j=1}^{k} \frac{2M}{k} f\left(-M + \frac{2Mj}{k}\right) g_\lambda\left(x - \left(-M + \frac{2Mj}{k}\right)\right).
\end{aligned}$$

3.2. General-Dimensional Input

To generalize our idea on input dimension 1 to general input dimension n_{in}, the main difficult part is how to make a ReLU network an approximate identity, for example, a kind of a B-spline function on $\mathbb{R}^2$. We emphasize that we need a function with the shape of a B-spline function, not an exact B-spline function on $\mathbb{R}^2$ (see Figure 1). For $n = 2$, we have

$$\begin{aligned}
&\mathrm{ReLU}(x_2 + (\mathrm{ReLU}(x_1 + 1) - 2\,\mathrm{ReLU}(x_1) + \mathrm{ReLU}(x_1 - 1))) - 2\,\mathrm{ReLU}(x_2) \\
&+ \mathrm{ReLU}(x_2 - (\mathrm{ReLU}(x_1 + 1) - 2\,\mathrm{ReLU}(x_1) + \mathrm{ReLU}(x_1 - 1))) \\
&= \begin{cases} 0 & \text{if} \quad |x_1| + |x_2| > 1 \\ 1 - |x_1| - |x_2| & \text{if} \quad |x_1| + |x_2| < 1. \end{cases}
\end{aligned}$$

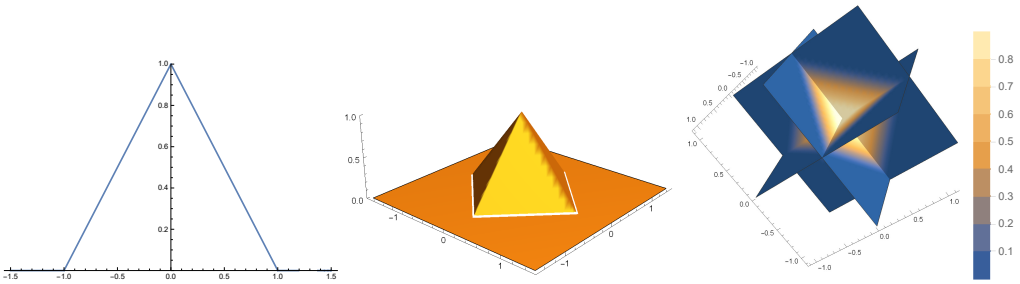

Figure 1. $g(\mathbf{x})$ Left: 1-dimension Center: 2-dimension Right: 3-dimension with range.

In general,

Lemma 1. *For $b \in \mathbb{R}$, let $h(x_1, b) = \mathrm{ReLU}(x_1 + b) - 2\,\mathrm{ReLU}(x_1) + \mathrm{ReLU}(x_1 - b)$. Then, we have*

$$h(x_n, h(x_{n-1}, h(x_{n-2}, \cdots, h(x_1, 1)))) = \begin{cases} 0 & \text{if} \quad \sum_{j=1}^{n} |x_j| \geq 1, \\ 1 - \sum_{j=1}^{n} |x_j| & \text{if} \quad \sum_{j=1}^{n} |x_j| < 1. \end{cases}$$

In addition, we have

$$\int_{\mathbb{R}^n} h(x_n, h(x_{n-1}, h(x_{n-2}, \cdots, h(x_1, 1))))\mathrm{d}\mathbf{x} = \frac{2^n}{(n+1)!}. \tag{1}$$

We notice that for

$$g(\mathbf{x}) = h(x_n, h(x_{n-1}, h(x_{n-2}, \cdots, h(x_1, 1)))), \quad \mathbf{x} \in \mathbb{R}^n$$

(see Figure 1) and $\frac{(n+1)!}{2^n} g_\lambda(\mathbf{x}) = \frac{(n+1)!}{2^n \lambda^n} g(\lambda^{-1}\mathbf{x})$, $\lambda > 0$ is an approximate identity: for a continuous function f on $[-M, M]^n$, $\frac{(n+1)!}{2^n} f * g_\lambda$ converges to $f(\mathbf{x})$ uniformly. For any $\epsilon > 0$, there is $\lambda > 0$ such that

$$\sup_{\mathbf{x} \in [-M,M]^{n_{in}}} \left| \frac{(n+1)!}{2^n} f * g_\lambda(\mathbf{x}) - f(\mathbf{x}) \right| < \frac{\epsilon}{2}. \tag{2}$$

Proof. We will use mathematical induction on n. We have already shown the case of $n = 1$. Suppose

$$h(x_n, h(x_{n-1}, \cdots, h(x_1, 1))) = \begin{cases} 0 & \text{if} \quad \sum_{j=1}^{n} |x_j| \geq 1, \\ 1 - \sum_{j=1}^{n} |x_j| & \text{if} \quad \sum_{j=1}^{n} |x_j| < 1. \end{cases}$$

For simplicity, let $g(\mathbf{x}) = h(x_n, h(x_{n-1}, \cdots, h(x_1, 1)))$, $\mathbf{x} \in \mathbb{R}^n$. If $\sum_{j=1}^{n+1} |x_j| \geq 1$, then $|x_{n+1}| \geq 1 - \sum_{j=1}^{n} |x_j|$ and thus $|x_{n+1}| \geq g(\mathbf{x})$. For $x_{n+1} \geq 0$,

$$\begin{aligned} h(x_{n+1}, g(\mathbf{x})) &= \mathrm{ReLU}(x_{n+1} + g(\mathbf{x})) - 2\,\mathrm{ReLU}(x_{n+1}) + \mathrm{ReLU}(x_{n+1} - g(\mathbf{x})) \\ &= (x_{n+1} + g(\mathbf{x})) - 2x_{n+1} + (x_{n+1} - g(\mathbf{x})) \end{aligned}$$

is equal to zero. In addition, for $x_{n+1} < 0$, we can show that $h(x_{n+1}, g(\mathbf{x}))$ is also zero. Now, we consider $\sum_{j=1}^{n+1} |x_j| < 1$. Then, by hypothesis, $g(\mathbf{x}) = 1 - \sum_{j=1}^{n} |x_j|$. For $x_{n+1} > 0$, we have

$$\begin{aligned} h(x_{n+1}, g(\mathbf{x})) &= \mathrm{ReLU}(x_{n+1} + g(\mathbf{x})) - 2\,\mathrm{ReLU}(x_{n+1}) + \mathrm{ReLU}(x_{n+1} - g(\mathbf{x})) \\ &= x_{n+1} + 1 - \sum_{j=1}^{n} |x_j| - 2x_{n+1} = 1 - \sum_{j=1}^{n+1} |x_j|, \end{aligned}$$

since $x_{n+1} - 1 + \sum_{j=1}^{n} |x_j| < 0$. Similarly, for $x_{n+1} < 0$, we have

$$h(x_{n+1}, g(\mathbf{x})) = x_{n+1} + 1 - \sum_{j=1}^{n} |x_j| = 1 - \sum_{j=1}^{n+1} |x_j|.$$

Direct integral calculation follows (1). □

Again, similar to the one-dimensional case, by mensuration by division, there is $k \in \mathbb{N}$ such that

$$\sup_{\mathbf{x} \in [-M,M]^{n_{in}}} \left| \begin{aligned} &\frac{(n_{in}+1)!}{2^{n_{in}}} f * g_\lambda(\mathbf{x}) - \frac{(n_{in}+1)! M^{n_{in}}}{k^{n_{in}}} \sum_{\mathbf{j} \in \mathbb{Z}_k^{n_{in}}} f\left(-(M, \cdots, M) + \frac{2M\mathbf{j}}{k}\right) \\ &\qquad \times g_\lambda\left(\mathbf{x} - \left(-(M, \cdots, M) + \frac{2M\mathbf{j}}{k}\right)\right) \end{aligned} \right| < \frac{\epsilon}{2}$$

and thus, with (2), we have for some $\lambda > 0$ and $k \in \mathbb{N}$,

$$\left| f(\mathbf{x}) - \frac{(n_{in}+1)!M^{n_{in}}}{k^{n_{in}}} \sum_{\mathbf{j}\in\mathbb{Z}_k^{n_{in}}} f\left(-(M,\cdots,M)+\frac{2M\mathbf{j}}{k}\right) \times g_\lambda\left(\mathbf{x}-\left(-(M,\cdots,M)+\frac{2M\mathbf{j}}{k}\right)\right) \right| < \epsilon.$$

Now, our goal is to construct a ReLU network satisfying

$$f_{(l,\mathbf{n})}(\mathbf{x}) = \frac{(n_{in}+1)!M^{n_{in}}}{k^{n_{in}}} \sum_{\mathbf{j}\in\mathbb{Z}_k^{n_{in}}} f\left(-(M,\cdots,M)+\frac{2M\mathbf{j}}{k}\right) g_\lambda\left(\mathbf{x}-\left(-(M,\cdots,M)+\frac{2M\mathbf{j}}{k}\right)\right).$$

Since we have an approximate identity $\frac{(n_{in}+1)!}{2^{n_{in}}}g$, how to make the ReLU network remains. Here, the indexing part is not difficult, but it is slightly cumbersome. Similarly to the case $n_{in} = 1$, for

$$(i,\mathbf{j}) = (i, j_1, j_2, \cdots, j_{n_{in}}) \in \{1,\cdots,n_{in}\} \times \mathbb{Z}_k^{n_{in}} \backslash \{(1,1,\cdots,1)\}$$

(i index for dimension and $\mathbf{j}$ index for mensuration by division), let

$$L^{1,\cdots,1} : \mathbb{R}^{n_{in}} \to \mathbb{R}^{n_{in}+5}, L^{i,\mathbf{j}} : \mathbb{R}^{n_{in}+5} \to \mathbb{R}^{n_{in}+5},$$
$$L^{n_{in}k^{n_{in}}+1} : \mathbb{R}^{n_{in}+5} \to \mathbb{R}^2, \text{ and } L^{n_{in}k^{n_{in}}+2} : \mathbb{R}^2 \to \mathbb{R}$$

with

$$L^{1,1,\cdots,1}(\mathbf{x}) = \begin{pmatrix} x_1+M \\ x_2+M \\ \vdots \\ x_{n_{in}}+M \\ \lambda^{-1}(x_1-(-M+2M/k))-1 \\ \lambda^{-1}(x_1-(-(M+2M/k)) \\ \lambda^{-1}(x_1-(-(M+2M/k))+1 \\ 0 \\ 0 \end{pmatrix},$$

$$L^{i,\mathbf{j}}(\mathbf{x}) = \begin{pmatrix} x_1 \\ \vdots \\ x_{n_{in}} \\ \lambda^{-1}(x_i-M-(-M+2Mj_i/k))-(x_{n_{in}+1}-2x_{n_{in}+2}+x_{n_{in}+3}) \\ \lambda^{-1}(x_i-M-(-M+2Mj_i/k)) \\ \lambda^{-1}(x_i-M-(-M+2Mj_i/k))+(x_{n_{in}+1}-2x_{n_{in}+2}+x_{n_{in}+3}) \\ x_{n_{in}+4} \\ x_{n_{in}+5} \end{pmatrix}$$
$$\text{for} \quad i = 2,\cdots,n_{in},$$

$$L^{1,\mathbf{j}}(\mathbf{x}) = \begin{pmatrix} x_1 \\ \vdots \\ x_{n_{in}} \\ \lambda^{-1}(x_1 - M - (-M + \frac{2Mj_1}{k})) - (x_{n_{in}+1} - 2x_{n_{in}+2} + x_{n_{in}+3}) \\ \lambda^{-1}(x_1 - M - (-M + \frac{2Mj_1}{k})) \\ \lambda^{-1}(x_1 - M - (-M + \frac{2Mj_1}{k})) + (x_{n_{in}+1} - 2x_{n_{in}+2} + x_{n_{in}+3}) \\ x_{n_{in}+4} + \frac{(n_{in}+1)!M^{n_{in}}}{k^{n_{in}}\lambda^{n_{in}}} f_+\left(-(M,\cdots,M) + \frac{2M(\mathbf{j}-\mathbf{1_j})}{k}\right) \\ \qquad \times (x_{n_{in}+1} - 2x_{n_{in}+2} + x_{n_{in}+3})) \\ x_{n_{in}+5} + \frac{(n_{in}+1)!M^{n_{in}}}{k^{n_{in}}\lambda^{n_{in}}} f_-\left(-(M,\cdots,M) + \frac{2M(\mathbf{j}-\mathbf{1_j})}{k}\right) \\ \qquad \times (x_{n_{in}+1} - 2x_{n_{in}+2} + x_{n_{in}+3}) \end{pmatrix},$$

$$(\mathbf{j} \neq (1,1,\cdots,1))$$

$$L^{n_{in}k^{n_{in}}+1}(\mathbf{x}) = \begin{pmatrix} x_{n_{in}+4} + \frac{(n_{in}+1)!M^{n_{in}}}{k^{n_{in}}\lambda^{n_{in}}} f_+(M,\cdots,M)(x_{n_{in}+1} - 2x_{n_{in}+2} + x_{n_{in}+3}) \\ x_{n_{in}+5} + \frac{(n_{in}+1)!M^{n_{in}}}{k^{n_{in}}\lambda^{n_{in}}} f_-(M,\cdots,M)(x_{n_{in}+1} - 2x_{n_{in}+2} + x_{n_{in}+3}) \end{pmatrix},$$

and $L^{n_{in}k^{n_{in}}+2}(\mathbf{x}) = (x_1 - x_2)$.

Here,

$$\mathbf{1_j} = \begin{cases} (1,0,\cdots,0) & \text{if } j_1 \neq 1, \\ (k-1,1,0\cdots,0) & \text{if } j_1 = 1 \text{ and } j_2 \neq 1, \\ \vdots & \\ (k-1,k-1,\cdots,k-1,\underset{l+1\text{-th}}{1},0,\cdots,0) & \text{if } j_1 = j_2 = \cdots = j_l = 1 \text{ and } j_{l+1} \neq 1, \\ \vdots & \\ (k-1,k-1,\cdots,k-1,1) & \text{if } j_1 = \cdots = j_{n_{in}-1} = 1 \text{ and } j_{n_{in}} \neq 1. \end{cases}$$

We stack in such a way that the index i is increased, and then the second index j_1 is increased, and then j_2, and so on:

$$\begin{array}{llll} (1,1,1,\cdots,1), & (2,1,1,\cdots,1), & (3,1,1,\cdots,1), & \cdots, \quad (n_{in},1,1,\cdots,1), \\ (1,2,1,\cdots,1), & (2,2,1,\cdots,1), & & \cdots, \quad (n_{in},2,1,\cdots,1), \\ & & \vdots & \\ (1,k,1,\cdots,1), & & & \cdots, \quad (n_{in},k,1,\cdots,1), \\ (1,1,2,1\cdots,1), & & & \cdots, \quad (n_{in},1,2,1\cdots,1), \\ & & \vdots & \\ (1,k,2,1,\cdots,1), & & & \cdots, \quad (n_{in},k,2,1,\cdots,1), \\ (1,1,3,1,\cdots,1), & & & \cdots, \quad (n_{in},1,3,1,\cdots,1), \\ & & \vdots & \\ (1,k,k,1,\cdots,1), & & & \cdots, \quad (n_{in},k,k,1,\cdots,1), \\ (1,1,1,2,1\cdots,1), & & & \cdots, \quad (n_{in},1,1,2,1\cdots,1), \\ & & \vdots & \\ (1,k-1,k,k,k\cdots,k), & & & \cdots, \quad (n_{in},k-1,k,k,k\cdots,k), \\ (1,k,k,k,k\cdots,k), & & & \cdots, \quad (n_{in},k,k,k,k\cdots,k) \end{array}$$

(which means that the index i is the first one where we increase) and, if we omit the index i, then

$$\begin{array}{llll}
(1,1,1,\cdots,1), & (2,1,1,\cdots,1), & (3,1,1,\cdots,1),\cdots, & (k,1,1,\cdots,1),\\
(1,2,1,\cdots,1), & (2,2,1,\cdots,1), & \cdots, & (k,2,1,\cdots,1),\\
 & \vdots & & \\
(1,k,1,\cdots,1), & (2,k,1,\cdots,1), & \cdots,\ (k-1,k,1,\cdots,1), & (k,k,1,\cdots,1),\\
(1,1,2,\cdots,1), & (2,1,2,\cdots,1), & \cdots,\ (k-1,1,2,\cdots,1), & (k,1,2,\cdots,1),\\
 & \vdots & & \\
(1,k,\cdots,k,k-1,k), & \cdots, & (k-1,k,\cdots,k-1,k), & (k,\cdots,k-1,k),\\
(1,k,k\cdots,k,k), & \cdots, & (k-1,k,k,\cdots,k,k), & (k,k\cdots,k,k).
\end{array} \tag{3}$$

In the second terms of the $n_{in}+4$ and $n_{in}+5$-th components of $L^{1,\mathbf{j}}$, $\mathbf{j} \neq (1,1,\cdots,1)$

$$f_{\pm}\left(-(M,\cdots,M)+\frac{2M(\mathbf{j}-\mathbf{1}_{\mathbf{j}})}{k}\right)(x_{n_{in}+1}-2x_{n_{in}+2}+x_{n_{in}+3}),$$

we want the value at the previous step and thus $\mathbf{j}-\mathbf{1}_{\mathbf{j}}$ is the exact previous one step of $\mathbf{j}$.

We want to notice the $n_{in}+1$, $n_{in}+2$, and $n_{in}+3$-th components of a function $f_{1,n_{in},n_{in}+5,n_{in}+5}(\mathbf{x})$ represented by $L^{2,1,1,\cdots,1}\circ \mathrm{ReLU}\circ L^{1,1,1,\cdots,1}(\mathbf{x})$ are

$$\begin{array}{l}
\lambda^{-1}\left(x_2-M-\left(-M+\frac{2M}{k}\right)\right)-h\left(\lambda^{-1}\left(x_1-\left(-M+\frac{2M}{k}\right)\right),1\right)\\
\lambda^{-1}\left(x_2-M-\left(-M+\frac{2M}{k}\right)\right)\\
\lambda^{-1}\left(x_2-M-\left(-M+\frac{2M}{k}\right)\right)+h\left(\lambda^{-1}\left(x_1-\left(-M+\frac{2M}{k}\right)\right),1\right),
\end{array}$$

and thus $n_{in}+1$, $n_{in}+2$, and $n_{in}+3$-th components of a function $f_{2,n_{in},n_{in}+5,n_{in}+5,n_{in}+5}(\mathbf{x})$ becomes

$$\begin{array}{l}
\lambda^{-1}\left(x_3-M-\left(-M+\frac{2M}{k}\right)\right)-\bar{g}\left(\lambda^{-1}\left((x_1,x_2)-\left(-(M,M)+\frac{2M(1,1)}{k}\right)\right)\right)\\
\lambda^{-1}\left(x_3-M-\left(-M+\frac{2M}{k}\right)\right)\\
\lambda^{-1}\left(x_3-M-\left(-M+\frac{2M}{k}\right)\right)+\bar{g}\left(\lambda^{-1}\left((x_1,x_2)-\left(-(M,M)+\frac{2M(1,1)}{k}\right)\right)\right),
\end{array}$$

where $\bar{g}(x_1,x_2)=h(x_2,h(x_1,1))$. Thus, a function $f_{n_{in},n_{in},n_{in}+5,\cdots,n_{in}+5}(\mathbf{x})$ represented by $L^{1,2,1,\cdots,1}\circ\cdots\circ\mathrm{ReLU}\circ L^{1,1,\cdots,1}(\mathbf{x})$ is equal to

$$\begin{pmatrix} x_1+M \\ \vdots \\ x_{n_{in}}+M \\ \lambda^{-1}(x_1-M-(-M+\frac{2M}{k}))-(x_{n_{in}+1}-2x_{n_{in}+2}+x_{n_{in}+3}) \\ \lambda^{-1}(x_1-M-(-M+\frac{2M}{k})) \\ \lambda^{-1}(x_1-M-(-M+\frac{2M}{k}))+(x_{n_{in}+1}-2x_{n_{in}+2}+x_{n_{in}+3}) \\ \frac{(n_{in}+1)!M^{n_{in}}}{k^{n_{in}}\lambda^{n_{in}}}f_+\left(-(M,\cdots,M)+\frac{2M(1,\cdots,1)}{k}\right) \\ \qquad \times g\left(\lambda^{-1}\mathbf{x}-\lambda^{-1}\left(-(M,\cdots,M)+\frac{2M(1,\cdots,1)}{k}\right)\right) \\ \frac{(n_{in}+1)!M^{n_{in}}}{k^{n_{in}}\lambda^{n_{in}}}f_-\left(-(M,\cdots,M)+\frac{2M(1,\cdots,1)}{k}\right) \\ \qquad \times g\left(\lambda^{-1}\mathbf{x}-\lambda^{-1}\left(-(M,\cdots,M)+\frac{2M(1,\cdots,1)}{k}\right)\right) \end{pmatrix},$$

since, for a non-negative function h, $\mathrm{ReLU}(h(x))=h(x)$. Then, the $n_{in}+4$ and $n_{in}+5$-th components of $f_{l,\mathbf{n}}(\mathbf{x})$ represented by $L^{1,\bar{\mathbf{j}}}\circ\cdots\circ\mathrm{ReLU}\circ L^{1,1,1,\cdots,1}(\mathbf{x})$ are

$$\begin{pmatrix} \sum_{\mathbf{j}<\bar{\mathbf{j}}}\frac{(n_{in}+1)!M^{n_{in}}}{k^{n_{in}}\lambda^{n_{in}}}f_+\left(-(M,\cdots,M)+\frac{2M\mathbf{j}}{k}\right) \\ \qquad \times g\left(\lambda^{-1}\mathbf{x}-\lambda^{-1}\left(-(M,\cdots,M)+\frac{2M\mathbf{j}}{k}\right)\right) \\ \sum_{\mathbf{j}<\bar{\mathbf{j}}}\frac{(n_{in}+1)!M^{n_{in}}}{k^{n_{in}}\lambda^{n_{in}}}f_-\left(-(M,\cdots,M)+\frac{2M\mathbf{j}}{k}\right) \\ \qquad \times g\left(\lambda^{-1}\mathbf{x}-\lambda^{-1}\left(-(M,\cdots,M)+\frac{2M\mathbf{j}}{k}\right)\right) \end{pmatrix},$$

where $\mathbf{j}\le\bar{\mathbf{j}}$ means when we give the well order to $\mathbf{j}$ in ordering like in (3) (for example, $(3,1,1,\cdots,1)\le(1,2,1,\cdots,1)$).

Therefore, we have

$$\begin{aligned} &L^{kn_{in}+2}\circ\mathrm{ReLU}\circ L^{kn_{in}+1}\circ\mathrm{ReLU}\circ L^{(n_{in},k,\cdots,k)}\circ\cdots\circ L^{2,1,1,\cdots,1}\circ\mathrm{ReLU}\circ L^{1,1,\cdots,1}(\mathbf{x}) \\ =\ & \frac{(n_{in}+1)!M^{n_{in}}}{k^{n_{in}}}\sum_{\mathbf{j}\in\mathbb{Z}_k^{n_{in}}}f_+\left(-(M,\cdots,M)+\frac{2M\mathbf{j}}{k}\right)g_\lambda\left(\mathbf{x}-\left(-(M,\cdots,M)+\frac{2M\mathbf{j}}{k}\right)\right) \\ & -\frac{(n_{in}+1)!M^{n_{in}}}{k^{n_{in}}}\sum_{\mathbf{j}\in\mathbb{Z}_k^{n_{in}}}f_-\left(-(M,\cdots,M)+\frac{2M\mathbf{j}}{k}\right)g_\lambda\left(\mathbf{x}-\left(-(M,\cdots,M)+\frac{2M\mathbf{j}}{k}\right)\right) \\ =\ & \frac{(n_{in}+1)!M^{n_{in}}}{k^{n_{in}}}\sum_{\mathbf{j}\in\mathbb{Z}_k^{n_{in}}}f\left(-(M,\cdots,M)+\frac{2M\mathbf{j}}{k}\right)g_\lambda\left(\mathbf{x}-\left(-(M,\cdots,M)+\frac{2M\mathbf{j}}{k}\right)\right). \end{aligned}$$

4. Conclusions

The universal approximation theorem is the mathematical theory of artificial neural networks and the classic one states that a feed-forward network with a hidden layer and some activation function can approximate continuous functions on compact subsets. Here, we show that, for a given continuous functions on compact subsets, the ReLU network with $n_{in}k^{n_{in}}+1$ hidden layers of at most $n_{in}+5$ nodes and proper weights can approximate to the function using the approximate identity. To show this, we explicitly give

connection and weights. Our construction is rather partially rather than fully connected; especially, each of the first $n_{in} + 3$ nodes in each hidden layer is connected to only one node of the previous layer. This means that actually we don't need a fully connected network.

Actually, we are interested in the relation between the parameter k and error bound ϵ because, if we know this, we can estimate how many hidden layers should be constructed to get the allowable error. Thus, the error estimate depending on the parameter k is our next research line.

Funding: This work was supported by the National Research Foundation of Korea grant funded by the Korea government (MSIP) (2018R1D1A3B07041149).

Data Availability Statement: Not applicable.

Acknowledgments: The author thanks K Kim, J Kang, and J Jeong for fruitful discussions.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444. [CrossRef] [PubMed]
2. Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*; MIT Press: Cambridge, MA, USA, 2016.
3. Arora, R.; Basu, A.; Mianjy, P.; Mukherjee, A. Understanding deep neural networks with rectified linear units. In Proceedings of the International Conference on Learning Representations, Vancouver, BC, Canada, 30 April–3 May 2018.
4. Mhaskar, H.N.; Poggio, T. Deep vs. shallow networks: An approximation theory perspective. *Anal. Appl.* **2016**, *14*, 829–848. [CrossRef]
5. Yarotsky, D. Error bounds for approximations with deep ReLU networks. *Neural Netw.* **2017**, *94*, 103–114. [CrossRef] [PubMed]
6. Cybenko, G. Approximation by superpositions of a sigmoidal function. *Math. Control. Signals Syst.* **1989**, *2*, 303–314. [CrossRef]
7. Barron, A.R. Approximation and estimation bounds for artificial neural networks. *Mach. Learn.* **1994**, *14*, 115–133. [CrossRef]
8. Hornik, K.; Stinchcombe, M.; White, H. Multilayer feedforward networks are universal approximators. *Neural Netw.* **1989**, *2*, 359–366. [CrossRef]
9. Funahashi, K.-I. On the approximate realization of continuous mappings by neural networks. *Neural Netw.* **1989**, *2*, 183–192. [CrossRef]
10. Delalleau, O.; Bengio, Y. Shallow vs. Deep Sum-Product Networks. In *Advances in Neural Information Processing Systems 24*; Shawe-Taylor, J., Zemel, R.S., Bartlett, P.L., Pereira, F., Weinberger, K.Q., Eds.; Curran Associates, Inc.: Granada, Spain, 12–17 December 2011; pp. 666–674.
11. Eldan, R.; Shamir, O. The power of depth for feedforward neural networks. In Proceedings of the Conference on Learning Theory, New York, NY, USA, 23–26 June 2016; pp. 907–940.
12. Glorot, X.; Bordes, A.; Bengio, Y. Deep sparse rectifier neural networks. In Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, Ft. Lauderdale, FL, USA, 11–13 April 2011; pp. 315–323.
13. Cohen, N.; Sharir, O.; Shashua, A. On the expressive power of deep learning: A tensor analysis. In Proceedings of the Conference on Learning Theory, New York, NY, USA, 23–26 June 2016; pp. 698–728.
14. Lu, Z.; Pu, H.; Wang, F.; Hu, Z.; Wang, L. The expressive power of neural networks: A view from the width. In *Advances in Neural Information Processing Systems 30*; Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R., Eds.; Curran Associates, Inc.: Long Beach, CA, USA, 4–9 December 2017; pp. 6231–6239.
15. Hanin, B. Universal function approximation by deep neural nets with bounded width and relu activations. *Mathematics* **2019**, *7*, 992. [CrossRef]
16. Hanin, B.; Sellke, M. Approximating continuous functions by relu nets of minimal width. *arXiv* **2017**, arXiv:1710.11278.
17. Suzuki, T. Adaptivity of deep ReLU network for learning in Besov and mixed smooth Besov spaces: Optimal rate and curse of dimensionality. In Proceedings of the International Conference on Learning Representations, New Orleans, LA, USA, 6–9 May 2019.
18. Ohn, I.; Kim, Y. Smooth function approximation by deep neural networks with general activation functions. *Entropy* **2019**, *21*, 627. [CrossRef] [PubMed]
19. Lin, H.; Jegelka, S. Resnet with one-neuron hidden layers is a universal approximator. In Proceedings of the 32nd International Conference on Neural Information Processing Systems, Red Hook, NY, USA, 2–8 December 2018; Curran Associates, Inc.: Montréal, QC, Canada, 2018; pp. 6172–6181.
20. Schmidt-Hieber, J. Nonparametric regression using deep neural networks with ReLU activation function. *Ann. Stat.* **2020**, *48*, 1875–1897. [CrossRef]