entropy

MDPI

Article

Analytic Function Approximation by Path-Norm-Regularized Deep Neural Networks

Aleksandr Beknazaryan

Institute of Environmental and Agricultural Biology (X-BIO), University of Tyumen, Volodarskogo 6, 625003 Tyumen, Russia; a.beknazaryan@utmn.ru

Abstract: We show that neural networks with an absolute value activation function and with network path norm, network sizes and network weights having logarithmic dependence on $1/\varepsilon$ can ε-approximate functions that are analytic on certain regions of $\mathbb{C}^d$.

Keywords: deep neural networks; analytic functions; path norm regularization; exponential convergence

1. Introduction

Deep neural networks have found broad applications in many areas and disciplines, such as computer vision, speech and audio recognition and natural language processing. Two of the main characteristics of a given class of neural networks are its complexity and approximating capability. Once the activation function is selected, a class of networks is determined by the specification of the network architecture (namely, its depth and width) and the choice of network weights. Hence, the estimation of the complexity of a given class is carried out by regularizing (one of) those parameters, and the approximation properties of obtained regularized classes of networks are then investigated.

The capability of shallow networks of depth 1 to approximate continuous functions is shown in the universal approximation theorem ([1]), and approximations of integrable functions by networks with fixed width are presented in [2]. Network-architecture-constrained approximations of analytic functions are given in [3], where it is shown that ReLU networks with depth depending logarithmically on $1/\varepsilon$ and width $d+4$ can ε-approximate analytic functions on the closed subcubes of $(-1,1)^d$.

The weight regularization of networks is usually carried out by imposing an l_p-related constraint on network weights, $p \geq 0$. The most popular types of such constraints include the l_0, l_1 and the *path norm* regularizations (see, respectively, [4–6] and references therein). Approximations of β-smooth functions on $[0,1]^d$ by l_0-regularized sparse ReLU networks are given in [5,7], and exponential rates of approximations of analytic functions by l_0-regularized networks are derived in [8].

Path-norm-regularized classes of deep ReLU networks are considered in [4], where, together with other characteristics, the Rademacher complexities of those classes are estimated. The network size independence of those estimates makes the path norm regularization particularly remarkable. As the estimation only uses the Lipschitz continuity (with Lipschitz constant 1), the idempotency and the non-negative homogeneity of the ReLU function, it can be extended to networks with the absolute value activation function. Network characteristics similar to the path norm are also considered in the works [9,10], where they are called, respectively, a *variation* and a *basis-path norm*, and statistical features of classes of networks are described in terms of those characteristics.

The objective of the present paper is the construction of path-norm-regularized networks that exponentially fast approximate analytic functions. Our goal is to achieve such convergence rates with activations that are idempotent, non-negative homogeneous

check for updates

Citation: Beknazaryan, A. Analytic Function Approximation by Path Norm Regularized Deep Neural Networks. *Entropy* **2022**, *24*, 1136. https://doi.org/10.3390/e24081136

Academic Editors: Diego Oliva and Ali Rıza Yıldız

Received: 6 July 2022
Accepted: 15 August 2022
Published: 16 August 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2022 by the author. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

and Lipschitz continuous with Lipschitz constant 1 so that the constructed path-norm-regularized networks fall within the scope of network classes studied in [4]. It turns out that networks with an absolute value activation function may suit this goal better than the networks with an ReLU activation function. More precisely, we show that analytic functions can be ε-approximated by networks with an absolute value activation function $a(x)$ and with the path norm, the depth, the width and the weights all depending logarithmically on $1/\varepsilon$. Such an approximation holds (i) on any subset $(0, 1-\delta]^d \subset (0,1)^d$ for analytic functions on $(0,1)^d$ with absolutely convergent power series; (ii) on the whole hypercube $[0,1]^d$ for functions that can be analytically continued to certain subsets of $\mathbb{C}^d$. Note that, as the network weights, as well as the total number of weights, depend logarithmically on $1/\varepsilon$, then the l_1 weight norms of the constructed approximating deep networks are also of logarithmic dependence on $1/\varepsilon$.

Note that the absolute value activation function considered in this paper is among the common built-in activation functions of the software-based neural network evolving method NEAT-Python ([11]). Training algorithms for networks with an absolute value activation function are developed in the works [12,13]. In addition, the VC-dimensions and the structures of the loss surfaces of neural networks with piecewise linear activation functions, including the absolute value function, are described in the works [14,15].

Notation: For a matrix $W \in \mathbb{R}^{d_1 \times d_2}$, we denote by $|W| \in \mathbb{R}^{d_1 \times d_2}$ the matrix obtained by taking the absolute values of the entries of W: $|W|_{ij} = |W_{ij}|$. For brevity of presentation, we will say that the matrix $|W|$ is the *absolute value of the matrix* W (note that, in the literature, there are also other definitions of the notion of an absolute value of a matrix). The path norm of a neural network f is denoted by $\|f\|_\times$. For $\mathbf{x} = (x_1, \dots, x_d) \in \mathbb{R}^d$ and $\mathbf{k} = (k_1, \dots, k_d) \in \mathbb{N}_0^d$, the degree of the monomial $\mathbf{x}^{\mathbf{k}} = x_1^{k_1} \cdot \dots \cdot x_d^{k_d}$ is defined to be $\|\mathbf{k}\|_1 = \sum_{i=1}^d k_i$. To assure that the matrix–vector multiplications are able to be accomplished, the vectors from $\mathbb{R}^d$, according to the context, may be treated as matrices either from $\mathbb{R}^{d\times 1}$ or from $\mathbb{R}^{1\times d}$.

2. The Class of Approximant Networks

Neural networks are constituted of weight matrices, biases and nonlinear activation functions acting neuron-wise in the hidden layers. The biases, also called shift vectors, can be omitted by adding a fixed coordinate 1 to the input vector and correspondingly modifying the weight matrices. As the definition of the path norm of networks does not assume the presence of shift vectors, we will add a coordinate 1 to the input vector $\mathbf{x}$ and will consider classes of neural networks of the form

$$\mathcal{F}_\alpha(L, \mathbf{p}) = \{f : [0,1]^p \to \mathbb{R}^{p_{L+1}} \mid f(\mathbf{x}) = W_L \circ \alpha \circ W_{L-1} \circ \alpha \circ \dots \circ \alpha \circ W_0(1, \mathbf{x})\},$$

where $W_i \in \mathbb{R}^{p_{i+1} \times p_i}$ are the weight matrices, $i = 0, \dots, L$, and $\mathbf{p} = (p_0, p_1, \dots, p_{L+1})$ is the width vector, with $p_0 = p + 1$. The number of hidden layers L determines the depth of networks from $\mathcal{F}_\alpha(L, \mathbf{p})$ and, in each layer, the activation function $\alpha : \mathbb{R} \to \mathbb{R}$ acts element-wise on the input vector. For $f \in \mathcal{F}_\alpha(L, \mathbf{p})$ given by

$$f(\mathbf{x}) = W_L \circ \alpha \circ W_{L-1} \circ \alpha \circ \dots \circ \alpha \circ W_0(1, \mathbf{x}), \tag{1}$$

let

$$\|f\|_\times := \left\| \prod_{i=0}^{L} |W_i| \right\|_1 \tag{2}$$

be the *path norm* of f, where $\|\cdot\|_1$ denotes the l_1 norm of the $p_0 (= p+1)$ dimensional vector $\prod_{i=0}^L |W_i|$ obtained as a product of absolute values of the weight matrices of f. For $B > 0$, let

$$\mathcal{F}_\alpha(L, \mathbf{p}, B) = \{f \in \mathcal{F}_\alpha(L, \mathbf{p}), \|f\|_\times \le B\}$$

be a path-norm-regularized subclass of $\mathcal{F}_\alpha(L, \mathbf{p})$. As the results obtained in [4] indicate, the path norm regularizations are particularly well-suited for networks whose activation function α is

- Lipschitz continuous with Lipschitz constant 1;
- Idempotent, that is, $\alpha(\alpha(x)) = \alpha(x)$, $x \in \mathbb{R}$;
- Non-negative homogeneous, that is, $\alpha(cx) = c\alpha(x)$, for $c \geq 0$, $x \in \mathbb{R}$.

We therefore aim to choose an activation α possessing those properties such that analytic functions can be approximated by networks from $\mathcal{F}_\alpha(L, \mathbf{p}, B)$ with a small path norm constraint B. The most popular activation functions satisfying the above conditions are the ReLU function $\sigma(x) = \max\{0, x\}$ and the absolute value function $a(x) = |x|$. Below, we show that, with the absolute value activation function, the path norms of approximant networks may be significantly smaller than the path norms of the ReLU networks.

The standard technique of neural network function approximation relies on approximating the product function $(x, y) \mapsto xy$, which then allows us to approximate monomials and polynomials of any desired degree. In [7], the approximation of the product $xy = ((x+y)^2 - x^2 - y^2)/2$ is achieved by approximating the function $x \mapsto x^2$. The latter is based on the observation that, for the triangle wave

$$g_s(x) = \underbrace{g \circ g \circ \cdots \circ g}_{s \text{ times}}(x), \tag{3}$$

where $g : [0, 1] \to [0, 1]$ is defined by

$$g(x) = \begin{cases} 2x, & 0 \leq x < 1/2, \\ 2(1-x), & 1/2 \leq x \leq 1, \end{cases}$$

and for any positive integer m,

$$|x^2 - f_m(x)| \leq 2^{-2m-2},$$

where

$$f_m(x) := x - \sum_{s=1}^{m} \frac{g_s(x)}{2^{2s}}. \tag{4}$$

The approximation of x^2 by networks with the ReLU activation function $\sigma(x)$ then follows from the representation

$$g(x) = 2\sigma(x) - 4\sigma(x - 1/2). \tag{5}$$

Thus, in this case, we will obtain matrices containing weights 2 and 4, which will make the path norm of approximant networks big. Note that the same approach is also used in [3] for constructing ReLU network approximations of analytic functions. In [5], the approximation of the product

$$xy = h\left(\frac{x-y+1}{2}\right) - h\left(\frac{x+y}{2}\right) + \frac{x+y}{2} - \frac{1}{4}$$

is achieved by approximating the function $h(x) := x(1-x)$, which, in turn, is based on the observation that, for the triangle wave

$$R^k = T^k \circ T^{k-1} \circ \cdots \circ T^1,$$

where $T^k : [0, 2^{2-2k}] \to [0, 2^{-2k}]$ is defined by

$$T^k(x) := \sigma(x/2) - \sigma(x - 2^{1-2k}), \tag{6}$$

and for any positive integer m,

$$|h(x) - \sum_{k=1}^{m} R^k(x)| \leq 2^{-m}, \quad x \in [0,1].$$

Although in the representation (6), the coefficients (weights) are all in $[-1,1]$, the approximant $\sum_{k=1}^{m} R^k(x)$ in this case does not have the factors 2^{-2s} presented in the approximant $f_m(x)$ in (4), which, again, will result in big values of path norms. Therefore, in order to take advantage of the presence of those reducing weights, we would like to represent the function $g(x)$ in (5) by a linear combination of activation functions with smaller coefficients. This is possible if, instead of $\sigma(x)$, we deploy the absolute value activation function $a(x)$. Indeed, in this case, we have that $g(x)$ can be represented on $[0,1]$ as

$$g(x) = 1 - 2a(x - 1/2). \tag{7}$$

In the next section, we use the above representation (7) to show that analytic functions can be ε-approximated by networks from $\mathcal{F}_a(L, \mathbf{p}, B)$ with each of L, $\|\mathbf{p}\|_\infty$ and B, as well as the network weights having logarithmic dependence on $1/\varepsilon$. As all networks will have the same activation function $a(x) = |x|$, in the following, the subscript a will be omitted.

3. Results

We first construct a neural network with activation function $a(x)$, that, for the given $\gamma, m \in \mathbb{N}$, simultaneously approximates all d-dimensional monomials of a degree less than γ up to an error of $\gamma^2 4^{-m}$. The depth of this network has order $m \log_2 \gamma$ and its width is of order $m\gamma^{d+1}$. Moreover, the entries of the product of the absolute values of matrices of the network have an order of at most γ^5 (note the independence of m).

For $\gamma > 0$, let $C_{d,\gamma}$ denote the number of d-dimensional monomials $\mathbf{x}^{\mathbf{k}}$ with degree $\|\mathbf{k}\|_1 < \gamma$. Then, $C_{d,\gamma} < (\gamma+1)^d$ and the following holds:

Lemma 1. *There is a neural network* $\mathrm{Mon}^d_{m,\gamma} \in \mathcal{F}(L, \mathbf{p})$ *with* $L \leq \lceil \log_2 \gamma \rceil (2m+5) + 2$, $p_0 = d+1$, $p_{L+1} = C_{d,\gamma}$ *and* $\|\mathbf{p}\|_\infty \leq 6\gamma(m+2)C_{d,\gamma}$ *such that*

$$\left\| \mathrm{Mon}^d_{m,\gamma}(\mathbf{x}) - (\mathbf{x}^{\mathbf{k}})_{\|\mathbf{k}\|_1 < \gamma} \right\|_\infty \leq \gamma^2 4^{-m}, \quad \mathbf{x} \in [0,1]^d.$$

Moreover, the entries of the $C_{d,\gamma} \times (d+1)$*-dimensional matrix obtained by multiplying the absolute values of matrices presented in* $\mathrm{Mon}^d_{m,\gamma}$ *are all bounded by* $144(\gamma+1)^5$.

Taking in the above lemma $\gamma, m = \lceil \log_2 \frac{1}{\varepsilon} \rceil$, we obtain a neural network from $\mathcal{F}(L, \mathbf{p})$, with L and $\|\mathbf{p}\|_\infty$ having logarithmic dependence on $1/\varepsilon$, which simultaneously approximates the monomials of a degree at most of γ with error ε (up to a logarithmic factor). Moreover, the entries of the product of absolute values of matrices of this network will also have logarithmic dependence on $1/\varepsilon$. Below, we use this property to construct a neural network approximation of analytic and analytically continuable functions with an approximation error ε and with network parameters having logarithmic order.

Theorem 1. *Let* $f(\mathbf{x}) = \sum_{\mathbf{k} \in \mathbb{N}_0^d} a_{\mathbf{k}} \mathbf{x}^{\mathbf{k}}$ *be an analytic function on* $(0,1)^d$ *with* $\sum_{\mathbf{k} \in \mathbb{N}_0^d} |a_{\mathbf{k}}| \leq F$. *Then, for any* $\varepsilon, \delta \in (0,1)$, *there is a constant* $C = C(d, F)$ *and a neural network* $F_\varepsilon \in \mathcal{F}(L, \mathbf{p}, B)$ *with* $L \leq C(\log_2 \frac{1}{\delta})(\log_2^2 \frac{1}{\varepsilon})$, $\|\mathbf{p}\|_\infty \leq \frac{C}{\delta^{d+1}}(\log_2 \frac{1}{\varepsilon})^{d+2}$ *and*

$$B \leq 10^4 dF \left(\frac{\log_2((2F+16)/\varepsilon)}{\delta} \right)^5,$$

such that

$$|F_\varepsilon(\mathbf{x}) - f(\mathbf{x})| \le \frac{\varepsilon}{\delta^2}, \quad \textit{for all } \ \mathbf{x} \in (0, 1-\delta]^d.$$

Note that an exponential convergence rate of deep ReLU network approximants on subintervals $(0, 1-\delta]^d$ is also given in [3]. In our case, however, not only the depth and the width but also the path norm $\|F_\varepsilon\|_\times$ of the constructed network F_ε have logarithmic dependence on $1/\varepsilon$. Note that, in the above theorem, as δ approaches to 0, both $\|\mathbf{p}\|_\infty$ and B, as well as the approximation error, grow polynomially on $1/\delta$. In the next theorem, we use the properties of Chebyshev series to derive an exponential convergence rate on the whole hypercube $[0,1]^d$. Recall that the Chebyshev polynomials are defined as $T_0(x) = 1$, $T_1(x) = x$ and

$$T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x).$$

Chebyshev polynomials play an important role in the approximation theory ([16]), and, in particular, it is known ([17], Theorem 3.1) that if f is Lipschitz continuous on $[-1,1]$, then it has a unique representation as an absolutely and uniformly convergent Chebyshev series

$$f(x) = \sum_{k=0}^{\infty} a_k T_k(x).$$

Moreover, in case f can be analytically continued to an ellipse $E_\rho \subset \mathbb{C}$ with foci -1 and 1 and with the sum of semimajor and semiminor axes equal to $\rho > 1$, then the partial sums of the above Chebyshev series converge to f with a geometric rate and the coefficients a_k also decay with a geometric rate. This result was first derived by Bernstein in [18] and its extension to the multivariate case was given in [19]. Note that the condition $z \in E_\rho$ implies that $z^2 \in N_{1,h^2}$, where $h = (\rho - \rho^{-1})/2$ and, for $d, a > 0$, $N_{d,a} \subset \mathbb{C}$ denotes an open ellipse with foci 0 and d and the leftmost point $-a$. For $F > 0$, $\rho > 1$ and $h = (\rho - \rho^{-1})/2$, let $\mathcal{A}^d(\rho, F)$ be the space of functions $f : [0,1]^d \to \mathbb{R}$ that can be analytically continued to the region $\{\mathbf{z} \in \mathbb{C}^d : z_1^2 + \cdots + z_d^2 \in N_{d,h^2}\}$ and are bounded there by F. Using the extension of Bernstein's theorem to the multivariate case, we obtain

Lemma 2. *Let $\rho \ge 2^{\sqrt{d}}$. For $f \in \mathcal{A}^d(\rho, F)$, there is a constant $C = C(d, \rho, F)$ and a polynomial*

$$p(\mathbf{x}) = \sum_{\|\mathbf{k}\|_1 \le \gamma} b_{\mathbf{k}} \mathbf{x}^{\mathbf{k}}, \quad \mathbf{x} \in [0,1]^d,$$

with

$$|b_{\mathbf{k}}| \le C(\gamma+1)^d \tag{8}$$

and

$$|f(\mathbf{x}) - p(\mathbf{x})| \le C\rho^{-\gamma/\sqrt{d}}, \quad \textit{for all } \ \mathbf{x} \in [0,1]^d.$$

Combining Lemma 1 and Lemma 2, we obtain the following.

Theorem 2. *Let $\varepsilon \in (0,1)$ and let $\rho \ge 2^{\sqrt{d}}$. For $f \in \mathcal{A}^d(\rho, F)$, there is a constant $C = C(d, \rho, F)$ and a neural network $F_\varepsilon \in \mathcal{F}(L, \mathbf{p}, B)$ with $L \le C\log_2^2 \frac{1}{\varepsilon}$, $\|\mathbf{p}\|_\infty \le C(\log_2 \frac{1}{\varepsilon})^{d+2}$ and $B \le C(\log_2 \frac{1}{\varepsilon})^{2d+5}$ such that*

$$|F_\varepsilon(\mathbf{x}) - f(\mathbf{x})| \le \varepsilon, \quad \textit{for all } \ \mathbf{x} \in [0,1]^d.$$

We conclude this part by estimating the l_1 weight regularization of networks constructed in Theorem 2. First, the total number of weights in those networks is bounded by $(L+1)\|\mathbf{p}\|_\infty^2 = O(\log_2 \frac{1}{\varepsilon})^{2d+6}$. From (7), it follows that all of the weights of network $\mathrm{Mon}_{m,\gamma}^d$ from Lemma 1 are in $[-2,2]$. In Theorem 2, the network F_ε is obtained by adding to a network $\mathrm{Mon}_{m,\gamma}^d$, with $\gamma = m = O(\log_2 \frac{1}{\varepsilon})$, a layer with coefficients of partial sums of power series of an approximated function. Thus, using (8), we obtain that the l_1 weight norm of the network F_ε constructed in Theorem 2 has order $O(\log_2 \frac{1}{\varepsilon})^{4d+6}$.

4. Proofs

In the following proofs, I_k denotes an identity matrix of size $k \times k$ and all of the networks have activation $a(x) = |x|$. The proof of Lemma 1 is based on the following two lemmas.

Lemma 3. *For any positive integer m, there exists a neural network* $\mathrm{Mult}_m \in \mathcal{F}(2m+3, \mathbf{p})$, *with $p_0 = 3$, $p_{L+1} = 1$ and $\|\mathbf{p}\|_\infty = 3m+2$, such that*

$$|\mathrm{Mult}_m(x,y) - xy| \leq 3 \cdot 2^{-2m-3}, \quad \text{for all } x, y \in [0,1], \tag{9}$$

and the product of absolute values of the matrices presented in Mult_m *is equal to*

$$\left(3\sum_{k=1}^{m} \frac{2^k - 1}{2^{2k}}, 2 - 2^{-m}, 2 - 2^{-m}\right).$$

Proof. For $k \geq 2$, let R_k denote a row of length k with a first entry equal to $-1/2$, last entry equal to 1 and all other entries equal to 0. Let A_k be a matrix of size $(k+1) \times k$ obtained by adding the $(k+1)$-th row R_k to the identity matrix I_k. That is,

$$A_k = \begin{pmatrix} & & & I_k & & \\ -\frac{1}{2} & 0 & 0 & \dots & 0 & 1 \end{pmatrix}.$$

In addition, let B_k denote a matrix of size $k \times k$ given by

$$B_k = \begin{pmatrix} & & & & & & 0 \\ & & & & & & 0 \\ & & & I_{k-1} & & & \vdots \\ & & & & & & 0 \\ & & & & & & 0 \\ 1 & 0 & 0 & \dots & 0 & 0 & -2 \end{pmatrix}.$$

It then follows from (7) that

$$B_{m+2} \circ a \circ A_{m+1} \circ \dots \circ B_3 \circ a \circ A_2 \begin{pmatrix} 1 \\ x \end{pmatrix} = \begin{pmatrix} 1 \\ x \\ g_1(x) \\ g_2(x) \\ \cdot \\ \cdot \\ \cdot \\ g_m(x) \end{pmatrix},$$

where $g_s(x)$ is the function defined in (3), $s = 1, \dots, m$. Thus, if S_{m+2} is a row of length $m+2$ defined as

$$S_{m+2} = \left(0, 1, -\frac{1}{2^{2 \cdot 1}}, -\frac{1}{2^{2 \cdot 2}}, \dots, -\frac{1}{2^{2 \cdot m}}\right),$$

then

$$S_{m+2} \circ a \circ B_{m+2} \circ a \circ A_{m+1} \circ \dots \circ a \circ B_3 \circ a \circ A_2 \begin{pmatrix} 1 \\ x \end{pmatrix} = f_m(x),$$

where f_m is defined by (4). We have that

$$|S_{m+2}| \cdot |B_{m+2}| \cdot |A_{m+1}| \cdot \dots \cdot |B_3| \cdot |A_2| = \left(\sum_{k=1}^{m} \frac{2^{k+1}-2}{2^{2k}}, 2 - 2^{-m} \right).$$

As $xy = \frac{1}{2}\big((x+y)^2 - x^2 - y^2\big)$, then, in the first layer of Mult_m, we will obtain a vector

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ x \\ y \end{pmatrix} := C \begin{pmatrix} 1 \\ x \\ y \end{pmatrix} = \begin{pmatrix} 1 \\ x \\ 1 \\ y \\ 1 \\ x+y \end{pmatrix}$$

and will then apply the network in a parallel manner from the first part of the proof to each of the pairs $(1, x)$, $(1, y)$ and $(1, x+y)$. More precisely, for a given matrix M of size $p \times q$, let $\tilde{M}$ be a matrix of size $3p \times 3q$ defined as

$$\tilde{M} = \begin{pmatrix} M & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & M & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & M \end{pmatrix}.$$

Then, for the network

$$\mathrm{Mult}_m(x,y) = \left(-\frac{1}{2}, -\frac{1}{2}, \frac{1}{2} \right) \circ a \circ \tilde{S}_{m+2} \circ a \circ \tilde{B}_{m+2} \circ a \circ \tilde{A}_{m+1} \circ \dots \circ \tilde{B}_3 \circ a \circ \tilde{A}_2 \circ a \circ C \begin{pmatrix} 1 \\ x \\ y \end{pmatrix}$$

we have that

$$\mathrm{Mult}_m(x,y) = \frac{1}{2}(f_m(x+y) - f_m(x) - f_m(y)),$$

which, together with $|f_m(x) - x^2| < 2^{-2m-2}$ and the triangle inequality, implies (9). It remains to be noted that the product of absolute values of the matrices presented in Mult_m is equal to

$$\left(\frac{1}{2}, \frac{1}{2}, \frac{1}{2} \right) \cdot |\tilde{S}_{m+2}| \cdot |\tilde{B}_{m+2}| \cdot |\tilde{A}_{m+1}| \cdot \dots \cdot |\tilde{B}_3| \cdot |\tilde{A}_2| \cdot |C| = \left(3 \sum_{k=1}^{m} \frac{2^k - 1}{2^{2k}}, 2 - 2^{-m}, 2 - 2^{-m} \right),$$

which completes the proof of the lemma. □

Lemma 4. *For any positive integer m, there exists a neural network $\mathrm{Mult}_m^r \in \mathcal{F}(L, \mathbf{p})$, with $L = (2m+5)\lceil \log_2 r \rceil + 1$, $p_0 = r+1$, $p_{L+1} = 1$ and $\|\mathbf{p}\|_\infty \le 6r(m+2)+1$, such that*

$$|\mathrm{Mult}_m^r(\mathbf{x}) - \prod_{i=1}^{r} x_i| \le r^2 4^{-m} \quad \textit{for all} \quad \mathbf{x} = (x_1, \dots, x_r) \in [0,1]^r,$$

and, for the $(r+1)$-dimensional vector J_m^r obtained by multiplication of absolute values of matrices presented in Mult_m^r, we have that $\|J_m^r\|_\infty \le 144 r^4$.

Proof. First, for a given $k \in \mathbb{N}$, we construct a network $N_m^k \in \mathcal{F}(L, \mathbf{p})$ with $L = 2m+4$, $p_0 = 2k+1$ and $p_{L+1} = k+1$, such that

$$N_m^k(x_1, x_2, \dots, x_{2k-1}, x_{2k}) = (1, \mathrm{Mult}_m(x_1, x_2), \dots, \mathrm{Mult}_m(x_{2k-1}, x_{2k})).$$

In the first layer, we obtain a vector for which the first coordinate is 1 followed by triples $(1, x_{2l-1}, x_{2l})$ $l = 1, \dots, k$, that is, the vector $(1, 1, x_1, x_2, 1, x_3, x_4, \dots, 1, x_{2k-1}, x_{2k})$. N_m^k is then obtained by applying in parallel the network Mult_m to each triple $(1, x_{2l-1}, x_{2l})$ while keeping the first coordinate equal to 1. The product of absolute values of the matrices presented in this construction is a matrix of size $(k+1) \times (2k+1)$ having a form

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 \\ a_m & b_m & b_m & 0 & 0 & 0 & \dots & 0 & 0 & 0 \\ a_m & 0 & 0 & b_m & b_m & 0 & \dots & 0 & 0 & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ a_m & 0 & 0 & 0 & 0 & 0 & \dots & 0 & b_m & b_m \end{pmatrix},$$

where $a_m = 3\sum_{k=1}^{m} \frac{2^k-1}{2^{2k}}$ and $b_m = 2 - 2^{-m}$ are the coordinates obtained in the previous lemma. Let us now construct the network Mult_m^r. The first hidden layer of Mult_m^r computes

$$(1, x_1, \dots, x_r) \mapsto (1, x_1, \dots, x_r, \underbrace{1, 1, \dots, 1}_{2^q - r}),$$

where $q = \lceil \log_2 r \rceil$. We then subsequently apply the networks $N_m^{2^q}, N_m^{2^{q-1}}, \dots, N_m^2$ and, in the last layer, we multiply the outcome by $(0, 1)$. From Lemma 3 and triangle inequality, we have that $|\text{Mult}_m(x, y) - tz| \le 3 \cdot 2^{-2m-3} + |x - t| + |y - z|$, for $x, y, t, z \in [0, 1]$. Hence, by induction on q, we obtain that $|\text{Mult}_m^r(\mathbf{x}) - \prod_{i=1}^r x_i| \le 3^q 2^{-2m-3} \le 3r^2 2^{-2m-3} \le r^2 4^{-m}$.

Note that the product of absolute values of matrices in each network N_m^k has the above form, that is, in each row, it has at most three nonzero values, each of which is less than 2. As the matrices given in the first and the last layer of Mult_m^r also satisfy this property, then each entry of the product of absolute values of all matrices of Mult_m^r will not exceed $12^{q+2} \le 144r^4$. □

Proof of Lemma 1. We have that, if $\|\mathbf{k}\|_1 = 0$, then $\mathbf{x}^{\mathbf{k}} = 1$, and if $\|\mathbf{k}\|_1 = 1$, then $\mathbf{k}$ has only one non-zero coordinate, say, k_j, which is equal to 1 and $\mathbf{x}^{\mathbf{k}} = x_j$. Denote $N = C_{d,\gamma} - d - 1$ and let $\mathbf{k}^1, \dots, \mathbf{k}^N$ be the multi-indices satisfying $1 < \|\mathbf{k}^i\|_1 < \gamma$, $i = 1, \dots, N$. For $\mathbf{k} = (k_1, \dots, k_d)$ with $\|\mathbf{k}\|_1 > 1$, denote by $\mathbf{x}_{\mathbf{k}}$ the $(\|\mathbf{k}\|_1 + 1)$-dimensional vector of the form

$$\mathbf{x}_{\mathbf{k}} = (1, \underbrace{x_1, \dots, x_1}_{k_1}, \dots, \underbrace{x_d, \dots, x_d}_{k_d}).$$

The first layer of $\text{Mon}_{m,\gamma}^d$ computes the $\left(d + 1 + \sum_{i=1}^N (\|\mathbf{k}^i\|_1 + 1)\right)$-dimensional vector

$$(1, \mathbf{x}, \mathbf{x}_{\mathbf{k}^1}, \dots, \mathbf{x}_{\mathbf{k}^N})^\mathsf{T}$$

by multiplying the input vector by matrix Γ of size $\left(d + 1 + \sum_{i=1}^N (\|\mathbf{k}^i\|_1 + 1)\right) \times (r+1)$. In the following layers, we do not change the first $d+1$ coordinates (by multiplying them by I_{d+1}), and, to each $\mathbf{x}_{\mathbf{k}^i}$, we apply in parallel the network $\text{Mult}_m^{\|\mathbf{k}^i\|_1}$. Recall that, in Lemma 4, J_m^r denotes the $(r+1)$-dimensional vector obtained from the product of absolute values of the matrices of Mult_m^r. We then have that the product of the absolute values of the matrices of $\mathrm{Mon}_{m,\gamma}^d$ has the form

$$M = \begin{pmatrix} I_k & & & & \\ & J_m^{\|\mathbf{k}^1\|_1} & & \mathbf{0} & \\ & & J_m^{\|\mathbf{k}^2\|_1} & & \\ & \mathbf{0} & & \ddots & \\ & & & & J_m^{\|\mathbf{k}^N\|_1} \end{pmatrix} \cdot \Gamma.$$

As the matrix Γ only contains entries 0 and 1, then, applying Lemma 4, we obtain that the entries of M are bounded by

$$\max_{1\le i\le N} \left\| \left| J_m^{\|\mathbf{k}^i\|_1} \right| \right\|_1 \le 144(\gamma+1)^5.$$

□

Proof of Theorem 1. Let $\gamma = \left\lceil \frac{\log_2((2F+16)/\varepsilon)}{\log_2(1-\delta)^{-1}} \right\rceil$. Then, for $\mathbf{x} \in (0, 1-\delta]^d$, we have that

$$\left| f(\mathbf{x}) - \sum_{\|\mathbf{k}\|_1<\gamma} a_{\mathbf{k}} \mathbf{x}^{\mathbf{k}} \right| = \left| \sum_{\|\mathbf{k}\|_1\ge\gamma} a_{\mathbf{k}} \mathbf{x}^{\mathbf{k}} \right| \le (1-\delta)^\gamma F \le \frac{\varepsilon F}{2F+16} \le \frac{\varepsilon}{2} \le \frac{\varepsilon}{2\delta^2}. \tag{10}$$

Applying Lemma 1 with $m = \lceil \log_2 \frac{4F+16}{\varepsilon} \rceil$, we obtain that, for all $\mathbf{x} \in [0,1]^d$

$$\begin{aligned} &\left\| \mathrm{Mon}_{m,\gamma}^d(\mathbf{x}) - (\mathbf{x}^{\mathbf{k}})_{\|\mathbf{k}\|_1<\gamma} \right\|_\infty \le \gamma^2 4^{-m} \le \left(\frac{4}{\log_2^2(1-\delta)^{-1}} \right) \left(\log_2^2 \frac{2F+16}{\varepsilon} \right) \left(\frac{\varepsilon}{4F+16} \right)^2 \\ &\le \frac{4(2F+16)\varepsilon^2}{\delta^2 \varepsilon (4F+16)^2} \le \frac{\varepsilon}{2F\delta^2}, \end{aligned} \tag{11}$$

where we used the inequalities $\log_2(1-\delta)^{-1} \ge \delta, \delta \in (0,1)$, and $\log_2^2 r \le r$ for $r \ge 16$. In order to approximate the partial sum $\sum_{\|\mathbf{k}\|_1 \le \gamma} a_{\mathbf{k}} \mathbf{x}^{\mathbf{k}}$, we add one last layer with the coefficients of that partial sum to the network $\mathrm{Mon}_{m,\gamma+1}^d$. As the sum of absolute values of those coefficients is bounded by F, then, combining (10) and (11), for the obtained network F_ε we obtain

$$|F_\varepsilon(\mathbf{x}) - f(\mathbf{x})| \le \frac{\varepsilon}{\delta^2}, \quad \textit{for all } \; \mathbf{x} \in (0, 1-\delta]^d.$$

From Lemma 1 it follows that

$$\|F_\varepsilon\|_\times \le 144(d+1)F(\gamma+1)^5 \le 10^4 dF \left(\frac{\log_2((2F+16)/\varepsilon)}{\delta} \right)^5.$$

□

Let us now present the result from [19] that will be used to derive Lemma 2. First, if $f \in \mathcal{A}^d(\rho, F)$, then ([20], Theorem 4.1) f has a unique representation as an absolutely and uniformly convergent multivariate Chebyshev series

$$f(\mathbf{x}) = \sum_{k_1=0}^\infty \cdots \sum_{k_d=0}^\infty a_{k_1,\ldots,k_d} T_{k_1}(x_1) \ldots T_{k_d}(x_d), \quad \mathbf{x} \in [0,1]^d.$$

Note that, for $\mathbf{k} := (k_1, \ldots, k_d)$, the degree of a d-dimensional polynomial $T_{k_1}(x_1) \ldots T_{k_d}(x_d)$ is $\|\mathbf{k}\|_1 = k_1 + \cdots + k_d$. Then, for any non-negative integers $n_1, \ldots, n_d$, the partial sum

$$p(\mathbf{x}) = \sum_{k_1=0}^{n_1} \cdots \sum_{k_d=0}^{n_d} a_{\mathbf{k}} T_{k_1}(x_1) \ldots T_{k_d}(x_d) \tag{12}$$

is a polynomial truncation of the multivariate Chebyshev series of f of degree $d(p) = n_1 + \cdots + n_d$. It is shown in [19] that

Theorem 3. *For $f \in \mathcal{A}^d(\rho, F)$, there is a constant $C = C(d, \rho, F)$ such that the multivariate Chebyshev coefficients of f satisfy*

$$|a_{\mathbf{k}}| \le C\rho^{-\|\mathbf{k}\|_2} \tag{13}$$

and, for the polynomial truncations p of the multivariate Chebyshev series of f, we have that

$$\inf_{d(p) \le \gamma} \|f(\mathbf{x}) - p(\mathbf{x})\|_{[0,1]^d} \le C\rho^{-\gamma/\sqrt{d}}.$$

Proof of Lemma 2. Note that, from the recursive definition of the Chebyshev polynomials, it follows that, for any $k \ge 0$, the coefficients of the Chebyshev polynomial $T_k(x)$ are all bounded by 2^k. Let p now be a polynomial given by (12) with degree $d(p) \le \gamma$. As the number of summands in the right-hand side of (12) is bounded by $(\gamma+1)^d$, then, using (13), we obtain that p can be rewritten as

$$p(\mathbf{x}) = \sum_{\|\mathbf{k}\|_1 \le \gamma} b_{\mathbf{k}} \mathbf{x}^{\mathbf{k}},$$

with

$$|b_{\mathbf{k}}| \le C(\gamma+1)^d 2^{\|\mathbf{k}\|_1} \rho^{-\|\mathbf{k}\|_2} \le C(\gamma+1)^d 2^{\sqrt{d}\|\mathbf{k}\|_2} \rho^{-\|\mathbf{k}\|_2} \le C(\gamma+1)^d,$$

where the last inequality follows from the condition $\rho \ge 2^{\sqrt{d}}$. □

Proof of Theorem 2. The proof follows from Lemmas 1 and 2 by taking $\gamma = m = \lceil \log_2 \frac{1}{\varepsilon} \rceil$ and adding, to the network $\mathrm{Mon}^d_{m,\gamma+1}$, the last layer with the coefficients of the polynomial $p(\mathbf{x})$ from Lemma 2. For the obtained network F_ε we have that

$$\|F_\varepsilon\|_\times \le 144C(d+1)C_{d,\gamma+1}(\gamma+2)^d(\gamma+2)^5 \le 144C(d+1)(\gamma+2)^{2d+5},$$

where C is the constant from Lemma 2. □

5. Discussion

Although various activation functions, including the ReLU, sigmoid and the Gaussian function, have already been used in the literature for neural network approximations of smooth and analytic functions (see [3,8,21]), approximating properties of neural networks with an absolute value activation function, which is a built-in activation function of software-based neural network evolving methods (such as NEAT-Python, [11]), has been barely covered previously. Whereas the algorithms developed in the works [12,13] allow us to train neural networks with an absolute value activation function, in the present paper, we study the capabilities of those networks to approximate analytic functions. While popular types of constraints imposed on approximating neural networks are either controlling the l_p norms of network weights or adjusting their architectures, in the present work, we study approximating properties of neural networks with regularized path norms and show that networks with an absolute value activation function and with network path norms having logarithmic dependence on $1/\varepsilon$ can ε-approximate functions that are analytic on certain regions of $\mathbb{C}^d$. The sizes and the weights of constructed networks also have logarithmic dependence on $1/\varepsilon$.

Funding: This research was funded by NWO Vidi grant: "Statistical foundation for multilayer neural networks": VI.Vidi.192.021.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The author would like to thank Johannes Schmidt-Hieber for support and valuable suggestions. The author is also grateful to the referees for the evaluation of the paper and for constructive comments.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

References

1. Scarselli, F.; Tsoi, A.C. Universal approximation using feedforward neural networks: A survey of some existing methods, and some new results. *Neural Netw.* **1998**, *11*, 15–37.
2. Lu, Z.; Pu, H.; Wang, F.; Hu, Z.; Wang, L. The expressive power of neural networks: A view from the width. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 6231–6239.
3. E, W.; Wang, Q. Exponential convergence of the deep neural network approximation for analytic functions. *Sci. China Math.* **2018**, *61*, 1733–1740.
4. Neyshabur, B.; Tomioka, R.; Srebro, N. Norm-based capacity control in neural networks. In Proceeding of the 28th Conference on Learning Theory (COLT), Paris, France, 3–6 July 2015; pp. 1376–1401.
5. Schmidt-Hieber, J. Nonparametric regression using deep neural networks with ReLU activation function. *Ann. Stat.* **2020**, *48*, 1875–1897.
6. Taheri, M.; Xie, F.; ; Lederer, J. Statistical Guarantees for Regularized Neural Networks. *Neural Netw.* **2021**, *142*, 148–161.
7. Yarotsky, D. Error bounds for approximations with deep ReLU networks. *Neural Netw.* **2017**, *94*, 103–114.
8. Opschoor, J.A.A.; Schwab, C.; Zech, J. Exponential ReLU DNN Expression of Holomorphic Maps in High Dimension. *Constr. Approx.* **2021**, *55*, 537–582.
9. Barron, A.; Klusowski, J. Approximation and estimation for high-dimensional deep learning networks. *arXiv* **2018**, arXiv:1809.03090.
10. Zheng, S.; Meng, Q.; Zhang, H.; Chen, W.; Yu, N.; Liu, T. Capacity control of ReLU neural networks by basis-path norm. *arXiv* **2019**, arXiv:1809.07122.
11. Overview of Builtin Activation Functions. Available online: https://neat-python.readthedocs.io/en/latest/activation.html (accessed on 5 July 2022).
12. Batruni, R. A multilayer neural network with piecewise-linear structure and backpropagation learning. *IEEE Trans. Neural Netw.* **1991**, *2*, 395–403.
13. Lin, J.-N.; Unbehauen, R. Canonical piecewise-linear neural networks. *IEEE Trans. Neural Netw.* **1995**, *6*, 43–50.
14. Bartlett, P.L.; Harvey, N.; Liaw, C.; Mehrabian, A. Nearly-tight VC-dimension and pseudodimension bounds for piecewise linear neural networks. *J. Mach. Learn. Res.* **2019**, *20*, 1–17.
15. He, F.; Wang, B.; Tao, D. Piecewise linear activations substantially shape the loss surfaces of neural networks. In Proceedings of the International Conference on Learning Representations, Addis Ababa, Ethiopia, 26–30 April 2020.
16. Mason, J.C.; Handscomb, D.C. *Chebyshev Polynomials*; Chapman and Hall/CRC: New York, NY, USA, 2002.
17. Trefethen, L.N. *Approximation Theory and Approximation Practice*; SIAM: Philadelphia, PA, USA, 2013.
18. Bernstein, S. Sur la meilleure approximation de $|x|$ par des polynomes de degrés donnés. *Acta Math.* **1914**, *37*, 1–57.
19. Trefethen, L.N. Multivariate polynomial approximation in the hypercube. *Proc. Am. Math. Soc.* **2017**, *145*, 4837–4844.
20. Mason, J.C. Near-best multivariate approximation by Fourier series, Chebyshev series and Chebyshev interpolation. *J. Approx. Theory* **1980**, *28*, 349–358, .
21. Mhaskar, H.N. Neural networks for optimal approximation of smooth and analytic functions. *Neural Comput.* **1996**, *8*, 164–177.