On the approximation by single hidden layer feedforward neural networks with fixed weights

Namig J Guliyev, Vugar E Ismailov

ON THE APPROXIMATION BY SINGLE HIDDEN LAYER FEEDFORWARD NEURAL NETWORKS WITH FIXED WEIGHTS

NAMIG J. GULIYEV AND VUGAR E. ISMAILOV

Abstract. Feedforward neural networks have wide applicability in various disciplines of science due to their universal approximation property. Some authors have shown that single hidden layer feedforward neural networks (SLFNs) with fixed weights still possess the universal approximation property provided that approximated functions are univariate. But this phenomenon does not lay any restrictions on the number of neurons in the hidden layer. The more this number, the more the probability of the considered network to give precise results. In this note, we constructively prove that SLFNs with the fixed weight 1 and two neurons in the hidden layer can approximate any continuous function on a compact subset of the real line. The applicability of this result is demonstrated in various numerical examples. Finally, we show that SLFNs with fixed weights cannot approximate all continuous multivariate functions.

Contents

1. Introduction

Approximation capabilities of single hidden layer feedforward neural networks (SLFNs) have been investigated in many works over the past 30 years. Typical results show that SLFNs possess the universal approximation property; that is, they can approximate any continuous function on a compact set with arbitrary precision.

2010 *Mathematics Subject Classification.* 41A30, 41A63, 65D15, 68T05, 92B20.
Key words and phrases. feedforward neural network, approximation, hidden layer, sigmoidal function, activation function, weight.

An SLFN with r units in the hidden layer and input $\mathbf{x} = (x_1, \dots, x_d)$ evaluates a function of the form

$$\sum_{i=1}^{r} c_i \sigma(\mathbf{w}^i \cdot \mathbf{x} - \theta_i), \tag{1.1}$$

where the weights $\mathbf{w}^i$ are vectors in $\mathbb{R}^d$, the thresholds θ_i and the coefficients c_i are real numbers, and the activation function σ is a univariate function. Properties of this neural network model have been studied quite well. By choosing various activation functions, many authors proved that SLFNs with the chosen activation function possess the universal approximation property (see, e.g., [3, 4, 6, 7, 8, 10, 11, 14, 29]). That is, for any compact set $Q \subset \mathbb{R}^d$, the class of functions (1.1) is dense in $C(Q)$, the space of continuous functions on Q. The most general and complete result of this type was obtained by Leshno, Lin, Pinkus and Schocken [23]. They proved that a continuous activation function σ has the universal approximation property (or density property) if and only if it is not a polynomial. This result has shown the power of SLFNs within all possible choices of the activation function σ, provided that σ is continuous. For a detailed review of these and many other results, see [30].

In many applications, it is convenient to take the activation function σ as a *sigmoidal function* which is defined as

$$\lim_{t \to -\infty} \sigma(t) = 0 \quad \text{and} \quad \lim_{t \to +\infty} \sigma(t) = 1.$$

The literature on neural networks abounds with the use of such functions and their superpositions (see, e.g., [2, 4, 6, 8, 10, 11, 13, 15, 20, 22, 29]). The possibility of approximating a continuous function on a compact subset of the real line or d-dimensional space by SLFNs with a sigmoidal activation function has been well studied in a number of papers.

In recent years, the theory of neural networks has been developed further in this direction. For example, from the point of view of practical applications, neural networks with a restricted set of weights have gained a special interest (see, e.g., [9, 16, 17, 19, 21, 24]). It was proved that SLFNs with some restricted set of weights still possess the universal approximation property. For example, Stinchcombe and White [34] showed that SLFNs with a polygonal, polynomial spline or analytic activation function and a bounded set of weights have the universal approximation property. Ito [20] investigated this property of networks using monotone sigmoidal functions (tending to 0 at minus infinity and 1 at infinity), with only weights located on the unit sphere. In [16, 17, 19], one of the coauthors considered SLFNs with weights varying on a restricted set of directions and gave several necessary and sufficient conditions for good approximation by such networks. For a set W of weights consisting of two directions, he showed that there is a geometrically explicit solution to the problem. Hahm and Hong [13] went further in this direction, and showed that SLFNs with fixed weights can approximate arbitrarily well any univariate function. Since fixed weights reduce the computational expense and training time, this result is of particular interest. In a mathematical formulation, the result reads as follows.

Theorem 1.1 (Hahm and Hong [13]). *Assume f is a continuous function on a finite segment $[a, b]$ of $\mathbb{R}$. Assume σ is a bounded measurable sigmoidal function on $\mathbb{R}$. Then for any sufficiently small $\varepsilon > 0$ there exist constants c_i, $\theta_i \in \mathbb{R}$ and*

positive integers K and n such that

$$\left| f(x) - \sum_{i=1}^{n} c_i \sigma(Kx - \theta_i) \right| < \varepsilon$$

for all $x \in [a, b]$.

Note that in this theorem both K and n depend on ε. The smaller the ε, the more neurons in the hidden layer one should take to approximate with the required precision. This phenomenon is pointed out as necessary in many papers. For various activation functions σ, there are plenty of practical examples, diagrams, tables, etc. in the literature, showing how the number of neurons increases as the error of approximation gets smaller.

It is well known that one of the challenges of neural networks is the process of deciding optimal number of hidden neurons. The other challenge is understanding how to reduce the computational expense and training time. As usual, networks with fixed weights best fit this purpose. In this respect, Cao and Xie [2] strengthened the above result by specifying the number of hidden neurons to realize approximation to any continuous function. By implementing modulus of continuity, they established upper bound estimations for the approximation error. It was shown in [2] that for the class of Lipschitz functions $\mathrm{Lip}_M(\alpha)$ with a Lipschitz constant M and degree α, the approximation bound is $M(1 + \|\sigma\|)(b-a)n^{-\alpha}$, where $\|\sigma\|$ is the sup of $\sigma(x)$ on $[a, b]$.

Approximation capabilities of SLFNs with a fixed weight were also analyzed in Lin, Guo, Cao and Xu [26]. Taking the activation function σ as a continuous, even and 2π-periodic function, the authors of [26] showed that neural networks of the form

$$\sum_{i=1}^{r} c_i \sigma(x - x_i) \tag{1.2}$$

can approximate any continuous function on $[-\pi, \pi]$ with an arbitrary precision ε. Note that all the weights are fixed equal to 1, and consequently do not depend on ε. To prove this, they first gave an integral representation for trigonometric polynomials, and constructed explicitly a network formed as (1.2) that approximates this integral representation. Finally, the obtained result for trigonometric polynomials was used to prove a Jackson-type upper bound for the approximation error.

In this paper, we construct a special sigmoidal activation function which meets both the above mentioned challenges in the univariate setting. In mathematical terminology, we construct a sigmoidal function σ for which K and n in the above theorem do not depend on the error ε. Moreover, we can take $K = 1$ and $n = 2$. That is, only parameters c_i and θ_i depend on ε. Can we find these numbers? For a large class of functions f, especially for analytic functions, our answer to this question is positive. We give an algorithm and a computer program for computing these numbers in practice. Our results are illustrated by several examples. Finally, we show that SLFNs with fixed weights are not capable of approximating all multivariate functions with arbitrary precision.

2. Construction of a sigmoidal function

In this section, we construct algorithmically a sigmoidal function σ which we use in our main result in the following section. Besides sigmoidality, we take care

about smoothness and monotonicity of our σ in the weak sense. Here by "weak monotonicity" we understand behavior of a function whose difference in absolute value from a monotonic function is a sufficiently small number. In this regard, we say that a real function f defined on a set $X \subseteq \mathbb{R}$ is called λ-*increasing* (respectively, λ-*decreasing*) if there exists an increasing (respectively, decreasing) function $u\colon X \to \mathbb{R}$ such that $|f(x) - u(x)| \leq \lambda$ for all $x \in X$. Obviously, 0-monotonicity coincides with the usual concept of monotonicity, and a λ_1-increasing function is λ_2-increasing if $\lambda_1 \leq \lambda_2$.

To start with the construction of σ, assume that we are given a closed interval $[a, b]$ and a sufficiently small real number λ. We construct σ algorithmically, based on two numbers, namely λ and $d := b - a$. The following steps describe the algorithm.

1. Introduce the function

$$h(x) := 1 - \frac{\min\{1/2, \lambda\}}{1 + \log(x - d + 1)}.$$

Note that this function is strictly increasing on the real line and satisfies the following properties:

(1) $0 < h(x) < 1$ for all $x \in [d, +\infty)$;
(2) $1 - h(d) \leq \lambda$;
(3) $h(x) \to 1$, as $x \to +\infty$.

We want to construct σ satisfying the inequalities

$$h(x) < \sigma(x) < 1 \tag{2.1}$$

for $x \in [d, +\infty)$. Then our σ will tend to 1 as x tends to $+\infty$ and obey the inequality

$$|\sigma(x) - h(x)| \leq \lambda,$$

i.e., it will be a λ-increasing function.

2. Before proceeding to the construction of σ, we need to enumerate the monic polynomials with rational coefficients. Let q_n be the Calkin–Wilf sequence (see [1]). Then we can enumerate all the rational numbers by setting

$$r_0 := 0, \quad r_{2n} := q_n, \quad r_{2n-1} := -q_n, \ n = 1, 2, \ldots.$$

Note that each monic polynomial with rational coefficients can uniquely be written as $r_{k_0} + r_{k_1} x + \ldots + r_{k_{l-1}} x^{l-1} + x^l$, and each positive rational number determines a unique finite continued fraction

$$[m_0; m_1, \ldots, m_l] := m_0 + \cfrac{1}{m_1 + \cfrac{1}{m_2 + \cfrac{1}{\ddots + \cfrac{1}{m_l}}}}$$

with $m_0 \geq 0$, $m_1, \ldots, m_{l-1} \geq 1$ and $m_l \geq 2$. We now construct a bijection between the set of all monic polynomials with rational coefficients and the set of all positive rational numbers as follows. To the only zeroth-degree monic polynomial 1 we associate the rational number 1, to each first-degree monic polynomial of the form $r_{k_0} + x$ we associate the rational number $k_0 + 2$, to each second-degree monic

polynomial of the form $r_{k_0}+r_{k_1}x+x^2$ we associate the rational number $[k_0; k_1+2] = k_0 + 1/(k_1+2)$, and to each monic polynomial

$$r_{k_0} + r_{k_1}x + \ldots + r_{k_{l-2}}x^{l-2} + r_{k_{l-1}}x^{l-1} + x^l$$

of degree $l \geq 3$ we associate the rational number $[k_0; k_1+1, \ldots, k_{l-2}+1, k_{l-1}+2]$. In other words, we define $u_1(x) := 1$,

$$u_n(x) := r_{q_n-2} + x$$

if $q_n \in \mathbb{Z}$,

$$u_n(x) := r_{m_0} + r_{m_1-2}x + x^2$$

if $q_n = [m_0; m_1]$, and

$$u_n(x) := r_{m_0} + r_{m_1-1}x + \ldots + r_{m_{l-2}-1}x^{l-2} + r_{m_{l-1}-2}x^{l-1} + x^l$$

if $q_n = [m_0; m_1, \ldots, m_{l-2}, m_{l-1}]$ with $l \geq 3$. For example, the first few elements of this sequence are

$$1, \quad x^2, \quad x, \quad x^2 - x, \quad x^2 - 1, \quad x^3, \quad x - 1, \quad x^2 + x, \quad \ldots.$$

3. We start with constructing σ on the intervals $[(2n-1)d, 2nd]$, $n = 1, 2, \ldots$. For each monic polynomial $u_n(x) = \alpha_0 + \alpha_1 x + \ldots + \alpha_{l-1}x^{l-1} + x^l$, set

$$B_1 := \alpha_0 + \frac{\alpha_1 - |\alpha_1|}{2} + \ldots + \frac{\alpha_{l-1} - |\alpha_{l-1}|}{2}$$

and

$$B_2 := \alpha_0 + \frac{\alpha_1 + |\alpha_1|}{2} + \ldots + \frac{\alpha_{l-1} + |\alpha_{l-1}|}{2} + 1.$$

Note that the numbers B_1 and B_2 depend on n. To avoid complication of symbols, we do not indicate this in the notation.

Introduce the sequence

$$M_n := h((2n+1)d), \qquad n = 1, 2, \ldots.$$

Clearly, this sequence is strictly increasing and converges to 1.

Now we define σ as the function

$$\sigma(x) := a_n + b_n u_n\left(\frac{x}{d} - 2n + 1\right), \quad x \in [(2n-1)d, 2nd], \tag{2.2}$$

where

$$a_1 := \frac{1}{2}, \qquad b_1 := \frac{h(3d)}{2}, \tag{2.3}$$

and

$$a_n := \frac{(1+2M_n)B_2 - (2+M_n)B_1}{3(B_2 - B_1)}, \qquad b_n := \frac{1 - M_n}{3(B_2 - B_1)}, \qquad n = 2, 3, \ldots. \tag{2.4}$$

It is not difficult to notice that for $n > 2$ the numbers a_n, b_n are the coefficients of the linear function $y = a_n + b_n x$ mapping the closed interval $[B_1, B_2]$ onto the closed interval $[(1+2M_n)/3, (2+M_n)/3]$. Besides, for $n = 1$, i.e. on the interval $[d, 2d]$,

$$\sigma(x) = \frac{1 + M_1}{2}.$$

Therefore, we obtain that

$$h(x) < M_n < \frac{1+2M_n}{3} \leq \sigma(x) \leq \frac{2+M_n}{3} < 1, \tag{2.5}$$

for all $x \in [(2n-1)d, 2nd]$, $n = 1, 2, \ldots$.

4. In this step, we construct σ on the intervals $[2nd,(2n+1)d]$, $n=1,2,\ldots$. For this purpose we use the *smooth transition function*

$$\beta_{a,b}(x):=\frac{\widehat{\beta}(b-x)}{\widehat{\beta}(b-x)+\widehat{\beta}(x-a)},$$

where

$$\widehat{\beta}(x):=\begin{cases}e^{-1/x}, & x>0,\\ 0, & x\leq 0.\end{cases}$$

Obviously, $\beta_{a,b}(x)=1$ for $x\leq a$, $\beta_{a,b}(x)=0$ for $x\geq b$, and $0<\beta_{a,b}(x)<1$ for $a<x<b$.

Set

$$K_n:=\frac{\sigma(2nd)+\sigma((2n+1)d)}{2},\qquad n=1,2,\ldots.$$

Note that the numbers $\sigma(2nd)$ and $\sigma((2n+1)d)$ have already been defined in the previous step. Since both the numbers $\sigma(2nd)$ and $\sigma((2n+1)d)$ lie in the interval $(M_n,1)$, it follows that $K_n\in(M_n,1)$.

First we extend σ smoothly to the interval $[2nd,2nd+d/2]$. Take $\varepsilon:=(1-M_n)/6$ and choose $\delta\leq d/2$ such that

$$\left|a_n+b_nu_n\left(\frac{x}{d}-2n+1\right)-(a_n+b_nu_n(1))\right|\leq\varepsilon,\quad x\in[2nd,2nd+\delta]. \tag{2.6}$$

One can choose this δ as

$$\delta:=\min\left\{\frac{\varepsilon d}{b_nC},\frac{d}{2}\right\},$$

where $C>0$ is a number satisfying $|u_n'(x)|\leq C$ for $x\in(1,1.5)$. For example, for $n=1$, δ can be chosen as $d/2$. Now define σ on the first half of the interval $[2nd,(2n+1)d]$ as the function

$$\begin{aligned}\sigma(x):=&K_n-\beta_{2nd,2nd+\delta}(x)\\ &\times\left(K_n-a_n-b_nu_n\left(\frac{x}{d}-2n+1\right)\right),\quad x\in\left[2nd,2nd+\frac{d}{2}\right].\end{aligned} \tag{2.7}$$

Let us prove that $\sigma(x)$ satisfies the condition (2.1). Indeed, if $2nd+\delta\leq x\leq 2nd+d/2$, then there is nothing to prove, since $\sigma(x)=K_n\in(M_n,1)$. If $2nd\leq x<2nd+\delta$, then $0<\beta_{2nd,2nd+\delta}(x)\leq 1$ and hence from (2.7) it follows that for each $x\in[2nd,2nd+\delta)$, $\sigma(x)$ is between the numbers K_n and $A_n(x):=a_n+b_nu_n\left(\frac{x}{d}-2n+1\right)$. On the other hand, from (2.6) we obtain that

$$a_n+b_nu_n(1)-\varepsilon\leq A_n(x)\leq a_n+b_nu_n(1)+\varepsilon,$$

which together with (2.2) and (2.5) yields $A_n(x)\in\left[\frac{1+2M_n}{3}-\varepsilon,\frac{2+M_n}{3}+\varepsilon\right]$ for $x\in[2nd,2nd+\delta)$. Since $\varepsilon=(1-M_n)/6$, the inclusion $A_n(x)\in(M_n,1)$ is valid. Now since both K_n and $A_n(x)$ belong to $(M_n,1)$, we finally conclude that

$$h(x)<M_n<\sigma(x)<1,\quad \text{for } x\in\left[2nd,2nd+\frac{d}{2}\right].$$

We define σ on the second half of the interval in a similar way:

$$\begin{aligned}\sigma(x):=&K_n-(1-\beta_{(2n+1)d-\overline{\delta},(2n+1)d}(x))\\ &\times\left(K_n-a_{n+1}-b_{n+1}u_{n+1}\left(\frac{x}{d}-2n-1\right)\right),\quad x\in\left[2nd+\frac{d}{2},(2n+1)d\right],\end{aligned}$$

where

$$\overline{\delta} := \min\left\{\frac{\overline{\varepsilon} d}{b_{n+1}\overline{C}}, \frac{d}{2}\right\}, \qquad \overline{\varepsilon} := \frac{1 - M_{n+1}}{6}, \qquad \overline{C} \geq \sup_{[-0.5,0]} |u'_{n+1}(x)|.$$

One can easily verify, as above, that the constructed $\sigma(x)$ satisfies the condition (2.1) on $[2nd + d/2, 2nd + d]$ and

$$\sigma\left(2nd + \frac{d}{2}\right) = K_n, \qquad \sigma^{(i)}\left(2nd + \frac{d}{2}\right) = 0, \quad i = 1, 2, \ldots.$$

Steps 3 and 4 construct σ on the interval $[d, +\infty)$.

5. On the remaining interval $(-\infty, d)$, we define σ as

$$\sigma(x) := \left(1 - \widehat{\beta}(d - x)\right)\frac{1 + M_1}{2}, \quad x \in (-\infty, d).$$

It is not difficult to verify that σ is a strictly increasing, smooth function on $(-\infty, d)$. Note also that $\sigma(x) \to \sigma(d) = (1 + M_1)/2$, as x tends to d from the left and $\sigma^{(i)}(d) = 0$ for $i = 1, 2, \ldots$. This final step completes the construction of σ on the whole real line.

3. Practical computation and properties of the constructed sigmoidal function

It should be noted that the above algorithm allows one to compute the constructed σ at any point of the real axis instantly. The code of this algorithm is available at http://sites.google.com/site/njguliyev/papers/monic-sigmoidal. As a practical example, we give here the graph of σ (see Figure 3.1) and a numerical table (see Table 3.1) containing several computed values of this function on the interval $[0, 20]$. Figure 3.2 shows how the graph of λ-increasing function σ changes on the interval $[0, 100]$ as the parameter λ decreases.

The above σ obeys the following properties:

(1) σ is sigmoidal;
(2) $\sigma \in C^\infty(\mathbb{R})$;
(3) σ is strictly increasing on $(-\infty, d)$ and λ-strictly increasing on $[d, +\infty)$;
(4) σ is easily computable in practice.

All these properties are easily seen from the above exposition. But the essential property of our sigmoidal function is its ability to approximate an arbitrary continuous function using only a fixed number of translations and scalings of σ. More precisely, only two translations and scalings are sufficient. We formulate this important property as a theorem in the next section.

4. Main results

The main results of the paper are formulated in the following two theorems.

Theorem 4.1. *Assume that f is a continuous function on a finite segment $[a, b]$ of $\mathbb{R}$ and σ is the sigmoidal function constructed in Section 2. Then for any sufficiently small $\varepsilon > 0$ there exist constants c_1, c_2, θ_1 and θ_2 such that*

$$|f(x) - c_1\sigma(x - \theta_1) - c_2\sigma(x - \theta_2)| < \varepsilon$$

for all $x \in [a, b]$.

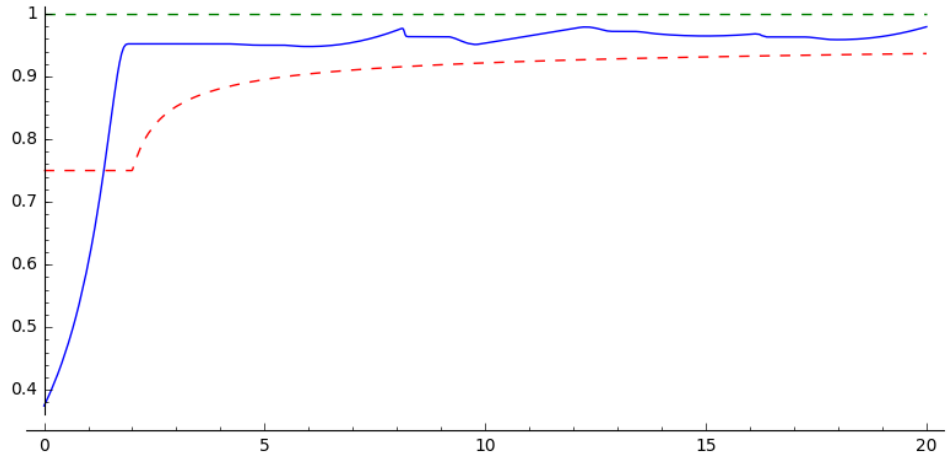

FIGURE 3.1. The graph of σ on $[0, 20]$ ($d = 2$, $\lambda = 1/4$)

TABLE 3.1. Some computed values of σ ($d = 2$, $\lambda = 1/4$)

t	σ	t	σ	t	σ	t	σ	t	σ
0.0	0.37462	4.0	0.95210	8.0	0.97394	12.0	0.97662	16.0	0.96739
0.4	0.44248	4.4	0.95146	8.4	0.96359	12.4	0.97848	16.4	0.96309
0.8	0.53832	4.8	0.95003	8.8	0.96359	12.8	0.97233	16.8	0.96309
1.2	0.67932	5.2	0.95003	9.2	0.96314	13.2	0.97204	17.2	0.96307
1.6	0.87394	5.6	0.94924	9.6	0.95312	13.6	0.97061	17.6	0.96067
2.0	0.95210	6.0	0.94787	10.0	0.95325	14.0	0.96739	18.0	0.95879
2.4	0.95210	6.4	0.94891	10.4	0.95792	14.4	0.96565	18.4	0.95962
2.8	0.95210	6.8	0.95204	10.8	0.96260	14.8	0.96478	18.8	0.96209
3.2	0.95210	7.2	0.95725	11.2	0.96727	15.2	0.96478	19.2	0.96621
3.6	0.95210	7.6	0.96455	11.6	0.97195	15.6	0.96565	19.6	0.97198

Proof. Set $d := b - a$ and divide the interval $[d, +\infty)$ into the segments $[d, 2d]$, $[2d, 3d]$, ... It follows from (2.2) that

$$\sigma(dx + (2n-1)d) = a_n + b_n u_n(x), \qquad x \in [0, 1] \tag{4.1}$$

for $n = 1, 2, \ldots$. Here a_n and b_n are computed by (2.3) and (2.4) for $n = 1$ and $n > 1$, respectively.

From (4.1) it follows that for each $n = 1, 2, \ldots$,

$$u_n(x) = \frac{1}{b_n}\sigma(dx + (2n-1)d) - \frac{a_n}{b_n}. \tag{4.2}$$

Let now g be any continuous function on the unit interval $[0, 1]$. By the density of polynomials with rational coefficients in the space of continuous functions over any compact subset of $\mathbb{R}$, for any $\varepsilon > 0$ there exists a polynomial $p(x)$ of the above form such that

$$|g(x) - p(x)| < \varepsilon$$

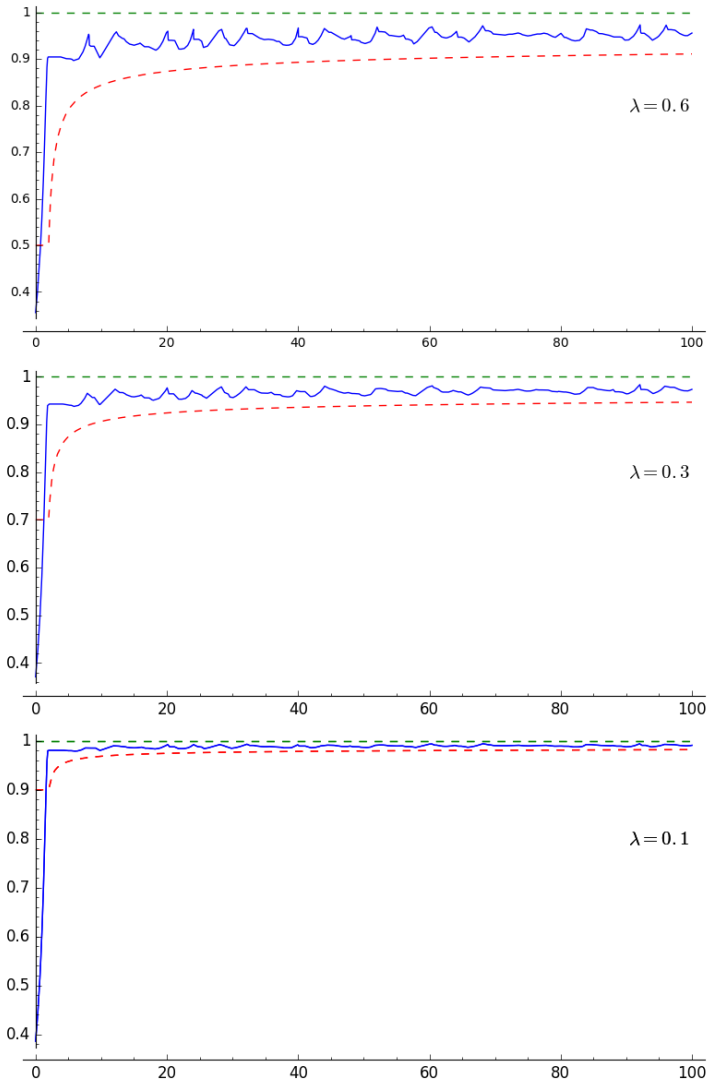

FIGURE 3.2. The graph of σ on $[0, 100]$ $(d = 2)$

for all $x \in [0, 1]$. Denote by p_0 the leading coefficient of p. If $p_0 \neq 0$ (i.e., $p \not\equiv 0$) then we define u_n as $u_n(x) := p(x)/p_0$, otherwise we just set $u_n(x) := 1$. In both cases

$$|g(x) - p_0 u_n(x)| < \varepsilon, \qquad x \in [0, 1].$$

This together with (4.2) means that

$$|g(x) - c_1\sigma(dx - s_1) - c_0| < \varepsilon$$

for some c_0, c_1, $s_1 \in \mathbb{R}$ and all $x \in [0, 1]$. Namely, $c_1 = p_0/b_n$, $s_1 = d - 2nd$ and $c_0 = p_0 a_n/b_n$. On the other hand, we can write $c_0 = c_2\sigma(dx - s_2)$, where

$c_2 := 2c_0/(1 + h(3d))$ and $s_2 := -d$. Hence,

$$|g(x) - c_1\sigma(dx - s_1) - c_2\sigma(dx - s_2)| < \varepsilon. \tag{4.3}$$

Note that (4.3) is valid for the unit interval $[0, 1]$. Using linear transformation it is not difficult to go from $[0, 1]$ to the interval $[a, b]$. Indeed, let $f \in C[a, b]$, σ be constructed as above, and ε be an arbitrarily small positive number. The transformed function $g(x) = f(a + (b - a)x)$ is well defined on $[0, 1]$ and we can apply the inequality (4.3). Now using the inverse transformation $x = (t-a)/(b-a)$, we can write

$$|f(t) - c_1\sigma(t - \theta_1) - c_2\sigma(t - \theta_2)| < \varepsilon$$

for all $t \in [a, b]$, where $\theta_1 = a + s_1$ and $\theta_2 = a + s_2$. The last inequality completes the proof. □

Since any compact subset of the real line is contained in a segment $[a, b]$, the following generalization of Theorem 4.1 holds.

Theorem 4.2. *Let Q be a compact subset of the real line and d be its diameter. Let λ be any positive number. Then one can algorithmically construct a computable sigmoidal activation function $\sigma\colon \mathbb{R} \to \mathbb{R}$, which is infinitely differentiable, strictly increasing on $(-\infty, d)$, λ-strictly increasing on $[d, +\infty)$, and satisfies the following property: For any $f \in C(Q)$ and $\varepsilon > 0$ there exist numbers c_1, c_2, θ_1 and θ_2 such that*

$$|f(x) - c_1\sigma(x - \theta_1) - c_2\sigma(x - \theta_2)| < \varepsilon$$

for all $x \in Q$.

Remark 4.1. The idea of using monic polynomials (see Section 2 and the proof above) is new in the numerical analysis of neural networks with limited number of hidden neurons. In fact, if one is interested more in a theoretical than in a practical result, then any countable dense subset of $C[0, 1]$ suffices. Maiorov and Pinkus [28] used such a subset to prove existence of a sigmoidal, monotonic and analytic activation function, and consequently a neural network with a fixed number of hidden neurons, which approximates arbitrarily well any continuous function. Note that the result is of theoretical value and the authors of [28] do not suggest constructing and using their sigmoidal function. In our previous work [12], we exploited a sequence of all polynomials with rational coefficients to construct a new universal sigmoidal function. Note that in [12] the problem of fixing weights in approximation by neural networks was not considered. Although the construction was efficient in the sense of computation of that sigmoidal function, some difficulties appeared while computing an approximating neural network parameters for some relatively simple approximated functions (see Remark 2 in [12]). This was a reason why we avoided giving practical numerical examples. The usage of monic polynomials in this instance turned out to be advantageous in reducing "running time" of the algorithm for computing the mentioned network parameters. This allows one to approximate various functions with sufficiently small precision and obtain all the required parameters (scaling coefficients and thresholds) in practice. We give corresponding numerical results in the next section.

5. Numerical results

We prove in Theorem 4.1 that any continuous function on $[a, b]$ can be approximated arbitrarily well by SLFNs with the fixed weight 1 and with only two neurons in the hidden layer. An activation function σ for such a network is constructed in Section 2. We have seen from the proof that our approach is totally constructive. One can evaluate the value of σ at any point of the real axis and draw its graph instantly using the programming interface at the URL shown at the beginning of Section 3. In the current section, we demonstrate our result in various examples. For different error bounds we find the parameters c_1, c_2, θ_1 and θ_2 in Theorem 4.1. All computations were done in SageMath [33]. For computations, we use the following algorithm, which works well for analytic functions. Assume f is a function, whose Taylor series around the point $(a+b)/2$ converges uniformly to f on $[a, b]$, and $\varepsilon > 0$.

(1) Consider the function $g(t) := f(a+(b-a)t)$, which is well-defined on $[0, 1]$;
(2) Find k such that the k-th Taylor polynomial

$$T_k(x) := \sum_{i=0}^{k} \frac{g^{(i)}(1/2)}{i!} \left(x - \frac{1}{2}\right)^i$$

satisfies the inequality $|T_k(x) - g(x)| \le \varepsilon/2$ for all $x \in [0, 1]$;
(3) Find a polynomial p with rational coefficients such that

$$|p(x) - T_k(x)| \le \frac{\varepsilon}{2}, \qquad x \in [0, 1],$$

and denote by p_0 the leading coefficient of this polynomial;
(4) If $p_0 \neq 0$, then find n such that $u_n(x) = p(x)/p_0$. Otherwise, set $n := 1$;
(5) For $n = 1$ and $n > 1$ evaluate a_n and b_n by (2.3) and (2.4), respectively;
(6) Calculate the parameters of the network as

$$c_1 := \frac{p_0}{b_n}, \qquad c_2 := \frac{2p_0 a_n}{b_n(1 + h(3d))}, \qquad \theta_1 := b - 2n(b-a), \qquad \theta_2 := 2a - b;$$

(7) Construct the network $\mathcal{N} = c_1\sigma(x - \theta_1) + c_2\sigma(x - \theta_2)$. Then $\mathcal{N}$ gives an ε-approximation to f.

In the sequel, we give four practical examples. To be able to make comparisons between these examples, all the considered functions are given on the same interval $[-1, 1]$. First we select the polynomial function $f(x) = x^3 + x^2 - 5x + 3$ as a target function. We investigate the sigmoidal neural network approximation to $f(x)$. This function was also considered in [13]. Note that in [13] the authors chose the sigmoidal function as

$$\sigma(x) = \begin{cases} 1, & \text{if } x \ge 0, \\ 0, & \text{if } x < 0, \end{cases}$$

and obtained the numerical results (see Table 5.1) for SLFNs with 8, 32, 128, 532 neurons in the hidden layer (see also [2] for an additional constructive result concerning the error of approximation in this example).

As it is seen from the table, the number of neurons in the hidden layer increases as the error bound decreases in value. This phenomenon is no longer true for our sigmoidal function (see Section 2). Using Theorem 4.1, we can construct explicitly an SLFN with only two neurons in the hidden layer, which approximates the above polynomial with arbitrarily given precision. Here by *explicit construction* we mean

TABLE 5.1. The Heaviside function as a sigmoidal function

N	Number of neurons ($2N^2$)	Maximum error
2	8	0.666016
4	32	0.165262
8	128	0.041331
16	512	0.010333

TABLE 5.2. Several ε-approximators of the function $1+x+x^2/2+x^3/6+x^4/24+x^5/120+x^6/720$

Number of neurons	Parameters of the network				Maximum error
	c_1	c_2	θ_1	θ_2	
2	2.0619×10^2	2.1131×10^2	-1979	-3	0.95
2	5.9326×10^2	6.1734×10^2	-1.4260×10^8	-3	0.60
2	1.4853×10^3	1.5546×10^3	-4.0140×10^{22}	-3	0.35
2	5.1231×10^2	5.3283×10^2	-3.2505×10^7	-3	0.10
2	4.2386×10^3	4.4466×10^3	-2.0403×10^{65}	-3	0.04
2	2.8744×10^4	3.0184×10^4	-1.7353×10^{442}	-3	0.01

that all the network parameters can be computed directly. Namely, the calculated values of these parameters are as follows: $c_1 \approx 2059.373597$, $c_2 \approx -2120.974727$, $\theta_1 = -467$, and $\theta_2 = -3$. It turns out that for the above polynomial we have an exact representation. That is, on the interval $[-1, 1]$ we have the identity

$$x^3 + x^2 - 5x + 3 \equiv c_1\sigma(x - \theta_1) + c_2\sigma(x - \theta_2).$$

Let us now consider the other polynomial function

$$f(x) = 1 + x + \frac{x^2}{2} + \frac{x^3}{6} + \frac{x^4}{24} + \frac{x^5}{120} + \frac{x^6}{720}.$$

For this function we do not have an exact representation as above. Nevertheless, one can easily construct a ε-approximating network with two neurons in the hidden layer for any sufficiently small approximation error ε. Table 5.2 displays numerical computations of the network parameters for six different approximation errors.

At the end we consider the nonpolynomial functions $f(x) = 4x/(4 + x^2)$ and $f(x) = \sin x - x\cos(x + 1)$. Tables 5.3 and 5.4 display all the parameters of the ε-approximating neural networks for the above six approximation error bounds. As it is seen from the tables, these bounds do not alter the number of hidden neurons. Figures 5.1, 5.2 and 5.3 show how graphs of some constructed networks $\mathcal{N}$ approximate the corresponding target functions f.

6. Analysis of the multivariate case

In this section, we want to draw the reader's attention to the following question. Do SLFNs with fixed weights preserve their universal approximation property in the multivariate setting? That is, if networks of the form

$$h(\mathbf{x}) = \sum_{i=1}^{r} c_i\sigma(\mathbf{w} \cdot \mathbf{x} - \theta_i), \tag{6.1}$$

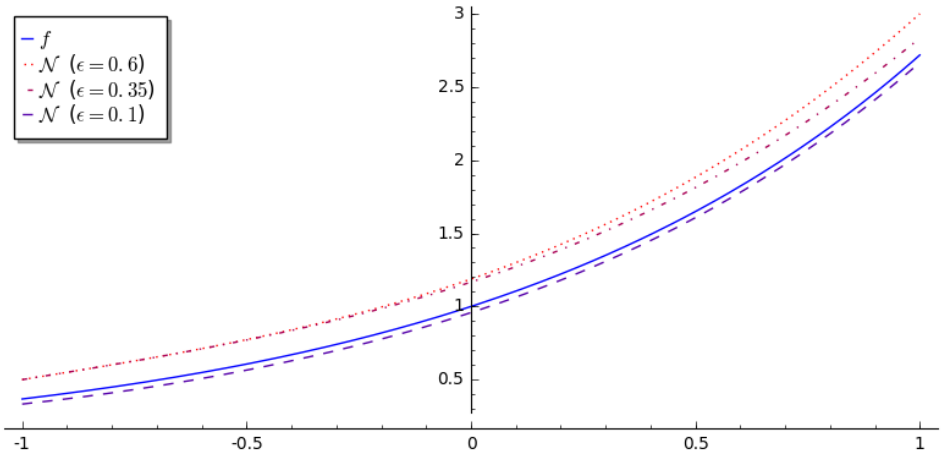

FIGURE 5.1. The graphs of $f(x) = 1 + x + x^2/2 + x^3/6 + x^4/24 + x^5/120 + x^6/720$ and some of its approximators ($\lambda = 1/4$)

TABLE 5.3. Several ε-approximators of the function $4x/(4 + x^2)$

Number of neurons	Parameters of the network				Maximum error
	c_1	c_2	θ_1	θ_2	
2	1.5965×10^2	1.6454×10^2	-283	-3	0.95
2	1.5965×10^2	1.6454×10^2	-283	-3	0.60
2	-1.8579×10^3	-1.9428×10^3	-6.1840×10^{11}	-3	0.35
2	1.1293×10^4	1.1842×10^4	-4.6730×10^{34}	-3	0.10
2	2.6746×10^4	2.8074×10^4	-6.8296×10^{82}	-3	0.04
2	-3.4218×10^6	-3.5939×10^6	-2.9305×10^{4885}	-3	0.01

TABLE 5.4. Several ε-approximators of the function $\sin x - x\cos(x + 1)$

Number of neurons	Parameters of the network				Maximum error
	c_1	c_2	θ_1	θ_2	
2	8.950×10^3	9.390×10^3	-3.591×10^{53}	-3	0.95
2	3.145×10^3	3.295×10^3	-3.397×10^{23}	-3	0.60
2	1.649×10^5	1.732×10^5	-9.532×10^{1264}	-3	0.35
2	-4.756×10^7	-4.995×10^7	$-1.308 \times 10^{180281}$	-3	0.10
2	-1.241×10^7	-1.303×10^7	-5.813×10^{61963}	-3	0.04
2	1.083×10^9	1.138×10^9	$-2.620 \times 10^{5556115}$	-3	0.01

where the weight $\mathbf{w} \in \mathbb{R}^d$ is fixed for all units of the hidden layer, but which may be different for different networks h, can approximate any continuous multivariate function $f(x_1, \ldots, x_d)$, $d > 1$, within arbitrarily small tolerance? Note that if $\mathbf{w}$ is fixed for all h, then it is obvious that there is a multivariate function which cannot be approximated by networks of the form (6.1). Indeed, the linear functional

$$F(f) = f(x_1) - f(x_2),$$

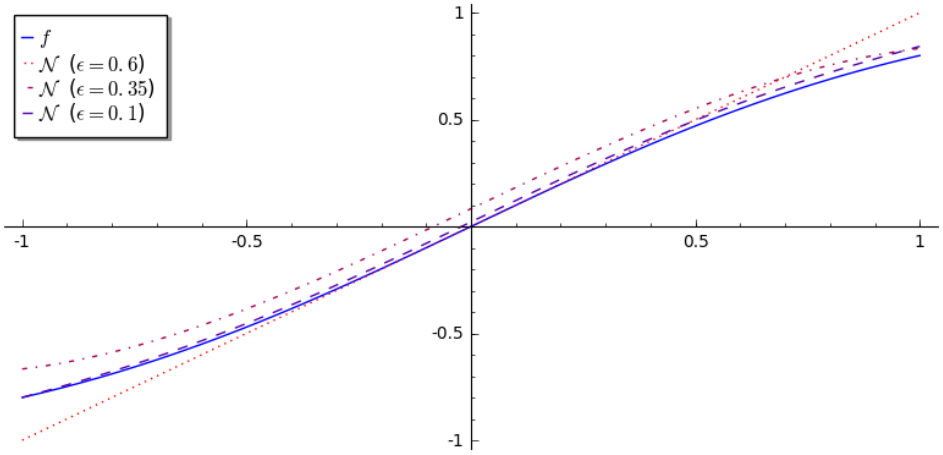

FIGURE 5.2. The graphs of $f(x) = 4x/(4+x^2)$ and some of its approximators ($\lambda = 1/4$)

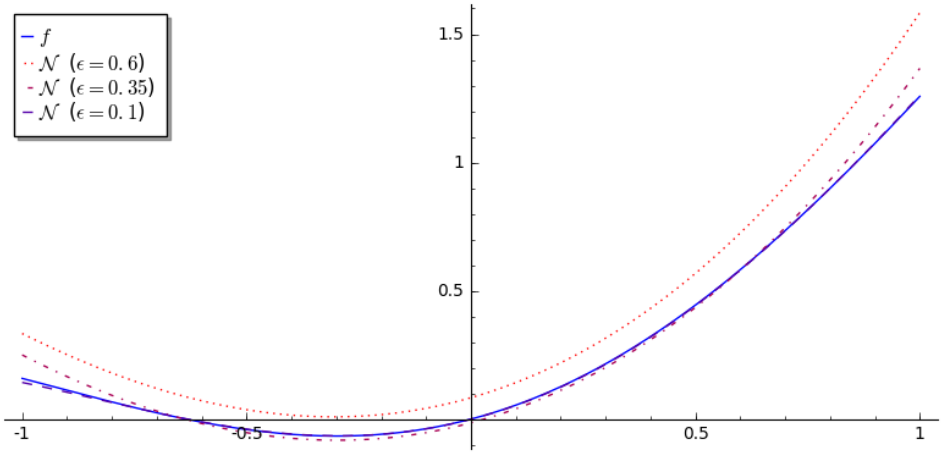

FIGURE 5.3. The graphs of $f(x) = \sin x - x\cos(x+1)$ and some of its approximators ($\lambda = 1/4$)

where $\mathbf{x}_1$ and $\mathbf{x}_2$ are selected so that $\mathbf{w}\cdot\mathbf{x}_1 = \mathbf{w}\cdot\mathbf{x}_2$, annihilates all functions h. Since the functional F is nontrivial, the set of all functions h, which we denote in the sequel by $\mathcal{H}$, is not dense in $C(Q)$ for an arbitrary compact set Q containing the points $\mathbf{x}_1$ and $\mathbf{x}_2$; hence approximation to all continuous functions cannot be possible on such compact sets Q. The above question, in the case where $\mathbf{w}$ is different for different networks h, is rather complicated. The positive answer to this question would mean, for example, that Theorem 1.1 admits a generalization to d-variable functions. Unfortunately, our answer to this question is negative. The details are as follows. Each summand in (6.1) is a function depending on the inner product $\mathbf{w}\cdot\mathbf{x}$. Thus, the whole sum itself, i.e. the function $h(\mathbf{x})$ is a function of the form $g(\mathbf{w}\cdot\mathbf{x})$. Note that functions of the form $g(\mathbf{w}\cdot\mathbf{x})$ are called *ridge functions*.

The literature abounds with the use of such functions and their linear combinations (see, e.g., [18, 31] and a great deal of references therein). We see that the set $\mathcal{H}$ is a subset of the set of ridge functions $\mathcal{R} := \left\{g(\mathbf{w}\cdot\mathbf{x})\colon \mathbf{w}\in\mathbb{R}^d\setminus\{\mathbf{0}\},\ g\in C(\mathbb{R})\right\}$. Along with $\mathcal{R}$, let us also consider the sets

$$\mathcal{R}_k := \left\{\sum_{i=1}^{k} g_i(\mathbf{w}^i\cdot\mathbf{x})\colon \mathbf{w}^i\in\mathbb{R}^d\setminus\{\mathbf{0}\},\ g_i\in C(\mathbb{R}),\ i=1,\dots,k\right\}.$$

Note that in $\mathcal{R}_k$ we vary over both the vectors $\mathbf{w}^i$ and the functions g_i, whilst k is fixed. Clearly, $\mathcal{R}=\mathcal{R}_1$. In [27], Lin and Pinkus proved that for any $k\in\mathbb{N}$, there exists a function $f\in C(\mathbb{R}^d)$ and a compact set $Q\subset\mathbb{R}^d$ such that

$$\inf_{g\in\mathcal{R}_k}\|f-g\|>0.$$

Here $\|\cdot\|$ denotes the uniform norm. It follows from this result that for each $k\in\mathbb{N}$ the set $\mathcal{R}_k$ (hence $\mathcal{R}$) is not dense in $C(\mathbb{R}^d)$ in the topology of uniform convergence on compacta. Since $\mathcal{H}\subset\mathcal{R}$, we obtain that the set $\mathcal{H}$ cannot be dense either. Thus there are always continuous multivariate functions which cannot be approximated arbitrarily well by SLFNs with fixed weights. This phenomenon justifies why we and the other researchers (see Introduction) investigate universal approximation property of such networks only in the univariate case.

The above analysis leads us to the following general negative result on the approximation by SLFNs with limited weights.

Theorem 6.1. *For any continuous function $\sigma\colon\mathbb{R}\to\mathbb{R}$, there is a multivariate continuous function which cannot be approximated arbitrarily well by neural networks of the form*

$$\sum_{i=1}^{r} c_i\sigma(\mathbf{w}^i\cdot\mathbf{x}-\theta_i), \tag{6.2}$$

where we vary over all $r\in\mathbb{N}$, $c_i,\theta_i\in\mathbb{R}$, $\mathbf{w}^i\in\mathbb{R}^d$, but the number of pairwise independent vectors (weights) $\mathbf{w}^i$ in each network (6.2) is uniformly bounded by some positive integer k (which is the same for all networks).

This theorem shows a particular limitation of neural networks with one hidden layer. We refer the reader to [5, 25] for interesting results and discussions around other limitations of such networks.

Acknowledgements

The research of the second author was supported by the Azerbaijan National Academy of Sciences under the program "Approximation by neural networks and some problems of frames".

References

[1] N. Calkin and H. S. Wilf, *Recounting the rationals*, Amer. Math. Monthly **107** (2000), 360–367.

[2] F. Cao and T. Xie, *The construction and approximation for feedforword neural networks with fixed weights*, Proceedings of the ninth international conference on machine learning and cybernetics, Qingdao, 2010, pp. 3164–3168.

[3] T. Chen and H. Chen, *Approximation of continuous functionals by neural networks with application to dynamic systems*, IEEE Trans. Neural Networks **4** (1993), 910–918.

[4] C. K. Chui and X. Li, *Approximation by ridge functions and neural networks with one hidden layer*, J. Approx. Theory **70** (1992), 131–141.
[5] C. K. Chui, X. Li and H. N. Mhaskar, *Limitations of the approximation capabilities of neural networks with one hidden layer*, Adv. Comput. Math. **5** (1996), no. 2-3, 233–243.
[6] D. Costarelli and R. Spigler, *Constructive approximation by superposition of sigmoidal functions*, Anal. Theory Appl. **29** (2013), 169–196.
[7] N. E. Cotter, *The Stone–Weierstrass theorem and its application to neural networks*, IEEE Trans. Neural Networks **1** (1990), 290–295.
[8] G. Cybenko, *Approximation by superpositions of a sigmoidal function*, Math. Control Signal Systems **2** (1989), 303–314.
[9] S. Draghici, *On the capabilities of neural networks using limited precision weights*, Neural Networks **15** (2002), 395–414.
[10] K. Funahashi, *On the approximate realization of continuous mapping by neural networks*, Neural Networks **2** (1989), 183–192.
[11] A. R. Gallant and H. White, *There exists a neural network that does not make avoidable mistakes*, Proceedings of the IEEE 1988 international conference on neural networks, vol. 1, IEEE Press, New York, 1988, pp. 657–664.
[12] N. J. Guliyev and V. E. Ismailov, *A single hidden layer feedforward network with only one neuron in the hidden layer can approximate any univariate function*, Neural Computation **28** (2016), no. 7, 1289–1304. arXiv:1601.00013
[13] N. Hahm and B.I. Hong, *An approximation by neural networks with a fixed weight*, Comput. Math. Appl. **47** (2004), no. 12, 1897–1903.
[14] K. Hornik, *Approximation capabilities of multilayer feedforward networks*, Neural Networks **4** (1991), 251–257.
[15] A. Iliev, N. Kyurkchiev and S. Markov, *On the approximation of the step function by some sigmoid functions*, Math. Comput. Simulation **133** (2017), 223–234.
[16] V. E. Ismailov, *Approximation by neural networks with weights varying on a finite set of directions*, J. Math. Anal. Appl. **389** (2012), no. 1, 72–83.
[17] ———, *Approximation by ridge functions and neural networks with a bounded number of neurons*, Appl. Anal. **94** (2015), no. 11, 2245–2260.
[18] ———, *Approximation by sums of ridge functions with fixed directions* (Russian), Algebra i Analiz, **28** (2016), no. 6, 20–69.
[19] V. E. Ismailov and E. Savas, *Measure theoretic results for approximation by neural networks with limited weights*, Numer. Funct. Anal. Optim. **38** (2017), no. 7, 819–830.
[20] Y. Ito, *Approximation of continuous functions on $\mathbb{R}^d$ by linear combinations of shifted rotations of a sigmoid function with and without scaling*, Neural Networks **5** (1992), 105–115.
[21] B. Jian, C. Yu and Y. Jinshou, *Neural networks with limited precision weights and its application in embedded systems*, Proceedings of the the second international workshop on education technology and computer science, Wuhan, 2010, pp. 86–91.
[22] V. Kůrková, *Kolmogorov's theorem and multilayer neural networks*, Neural Networks **5** (1992), 501–506.
[23] M. Leshno, V. Ya. Lin, A. Pinkus and S. Schocken, *Multilayer feedforward networks with a non-polynomial activation function can approximate any function*, Neural Networks **6** (1993), 861–867.
[24] Y. Liao, S.-C. Fang and H. L. W. Nuttle, *A neural network model with bounded-weights for pattern classification*, Comput. Oper. Res. **31** (2004), 1411–1426.
[25] S. Lin, *Limitations of shallow nets approximation*, Neural Networks **94** (2017), 96–102.
[26] S. Lin, X. Guo, F. Cao and Z. Xu, *Approximation by neural networks with scattered data* Appl. Math. Comput. **224** (2013), 29–35.
[27] V. Ya. Lin and A. Pinkus, *Fundamentality of ridge functions*, J. Approx. Theory **75** (1993), 295–311.
[28] V. Maiorov and A. Pinkus, *Lower bounds for approximation by MLP neural networks*, Neurocomputing **25** (1999), 81–91.
[29] H. N. Mhaskar and C. A. Micchelli, *Approximation by superposition of a sigmoidal function and radial basis functions*, Adv. Appl. Math. **13** (1992), 350–373.
[30] A. Pinkus, *Approximation theory of the MLP model in neural networks*, Acta numerica, 1999, Cambridge Univ. Press, Cambridge, 1999, pp. 143–195.
[31] ———, *Ridge functions*, Cambridge University Press, Cambridge, 2015.

[32] P. C. Sikkema, *Der Wert einiger Konstanten in der Theorie der Approximation mit Bernstein-Polynomen*, Numer. Math. **3** (1961), 107–116.
[33] W. A. Stein et al., *Sage Mathematics Software (Version 7.6)*, The Sage Developers, 2017, http://www.sagemath.org.
[34] M. Stinchcombe and H. White, *Approximating and learning unknown mappings using multilayer feedforward networks with bounded weights*, Proceedings of the 1990 IEEE international joint conference on neural networks, vol. 3, IEEE, New York, 1990, pp. 7–16.

Institute of Mathematics and Mechanics, Azerbaijan National Academy of Sciences, 9 B. Vahabzadeh str., AZ1141, Baku, Azerbaijan.
E-mail address: njguliyev@gmail.com

Institute of Mathematics and Mechanics, Azerbaijan National Academy of Sciences, 9 B. Vahabzadeh str., AZ1141, Baku, Azerbaijan.
E-mail address: vugaris@mail.ru