A SINGLE HIDDEN LAYER FEEDFORWARD NETWORK WITH ONLY ONE NEURON IN THE HIDDEN LAYER CAN APPROXIMATE ANY UNIVARIATE FUNCTION

NAMIG J. GULIYEV AND VUGAR E. ISMAILOV

ABSTRACT. The possibility of approximating a continuous function on a compact subset of the real line by a feedforward single hidden layer neural network with a sigmoidal activation function has been studied in many papers. Such networks can approximate an arbitrary continuous function provided that an unlimited number of neurons in a hidden layer is permitted. In this paper, we consider constructive approximation on any finite interval of $\mathbb{R}$ by neural networks with only one neuron in the hidden layer. We construct algorithmically a smooth, sigmoidal, almost monotone activation function σ providing approximation to an arbitrary continuous function within any degree of accuracy.

1. INTRODUCTION

Neural networks are being successfully applied across an extraordinary range of problem domains, in fields as diverse as computer science, finance, medicine, engineering, physics, etc. The main reason for such popularity is their ability to approximate arbitrary functions. For the last 30 years a number of results have been published showing that the artificial neural network called a *feedforward network with one hidden layer* can approximate arbitrarily well any continuous function of several real variables. These results play an important role in determining boundaries of efficacy of the considered networks. But the proofs are usually do not state how many neurons should be used in the hidden layer. The purpose of this paper is to prove constructively that a neural network having only one neuron in its single hidden layer can approximate arbitrarily well all continuous functions defined on any compact subset of the real axis.

The building blocks for neural networks are called *neurons*. An artificial neuron is a device with n real inputs and an output. This output is generally a superposition of a univariate function with an affine function in the n-dimensional Euclidean space, that is a function of the form $\sigma(w_1 x_1 + \cdots + w_n x_n - \theta)$. The neurons are organized in layers. Each neuron of each layer is connected to each neuron of the subsequent (and thus previous) layer. Information flows from one layer to the subsequent layer (thus the term feedforward). A feedforward neural network with one hidden layer has three layers: input layer, hidden layer, and output layer. A feedforward network with one hidden layer consisting of r neurons computes functions of the form

$$c_0 + \sum_{i=1}^{r} c_i \sigma(w_i \cdot x - \theta_i). \tag{1.1}$$

1

Here the vectors w_i, called *weights*, are vectors in $\mathbb{R}^n$; the *thresholds* θ_i and the coefficients c_i are real numbers and σ is a univariate activation function. The following are common examples of activation functions:

$$\sigma(t) = \frac{1}{1 + e^{-t}} \qquad \text{(the squashing function)},$$

$$\sigma(t) = \begin{cases} 0, & t \leq -1, \\ \dfrac{t+1}{2}, & -1 \leq t \leq 1, \\ 1, & t \geq 1 \end{cases} \qquad \text{(the piecewise linear function)},$$

$$\sigma(t) = \frac{1}{\pi} \arctan t + \frac{1}{2} \qquad \text{(the arctan sigmoid function)},$$

$$\sigma(t) = \frac{1}{\sqrt{2\pi}} \int\limits_{-\infty}^{t} e^{-x^2/2} dx \qquad \text{(the Gaussian function)}.$$

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. Note that all the above activation functions are sigmoidal.

In approximation by neural networks, there are two main problems. The first is the *density* problem of determining the conditions under which an arbitrary target function can be approximated arbitrarily well by neural networks. The second problem, called the *complexity* problem, is to determine how many neurons in hidden layers are necessary to give a prescribed degree of approximation. This problem is almost the same as the problem of degree of approximation (see [1, 3, 11]). The possibility of approximating a continuous function on a compact subset of the real line (or n-dimensional space) by a single hidden layer neural network with a sigmoidal activation function has been well studied in a number of papers. Different methods were used. Carroll and Dickinson [4] used the inverse Radon transformation to prove the universal approximation property of single hidden layer neural networks. Gallant and White [10] constructed a specific continuous, nondecreasing sigmoidal function, called a cosine squasher, from which it was possible to obtain any Fourier series. Thus their activation function had the density property. Cybenko [8] and Funahashi, [9] independently from each other, established that feedforward neural networks with a continuous sigmoidal activation function can approximate any continuous function within any degree of accuracy on compact subsets of $\mathbb{R}^n$. Cybenko's proof uses the functional analysis method, combining the Hahn–Banach Theorem and the Riesz Representation Theorem, whiles Funahashi's proof applies the result of Irie and Miyake [14] on the integral representation of $f \in L_1(\mathbb{R}^n)$ functions, using a kernel which can be expressed as a difference of two sigmoidal functions. Hornik, Stinchcombe and White [13] applied the Stone–Weierstrass theorem, using trigonometric functions.

Chen, Chen and Liu [5] extended the result of Cybenko by proving that any continuous function on a compact subset of $\mathbb{R}^n$ can be approximated by a single

hidden layer feedforward network with a bounded (not necessarily continuous) sigmoidal activation function. Almost the same result was independently obtained by Jones [18].

Chui and Li [6] proved that a single hidden layer network with a continuous sigmoidal activation function having integer weights and thresholds can approximate an arbitrary continuous function on a compact subset of $\mathbb{R}$. Ito [17] established a density result for continuous functions on a compact subset of $\mathbb{R}$ by neural networks with a sigmoidal function having only unit weights. Density properties of a single hidden layer network with a restricted set of weights were studied also in other papers (for a detailed discussion see [15]).

In many subsequent papers, which dealt with the density problem, nonsigmoidal activation functions were allowed. Among them are the papers by Stinchcombe and White [26], Cotter [7], Hornik [12], Mhaskar and Micchelli [21], and other researchers. The more general result in this direction belongs to Leshno, Lin, Pinkus and Schocken [19]. They proved that the necessary and sufficient condition for any continuous activation function to have the density property is that it not be a polynomial.

It should be remarked that in all the above mentioned works the number of neurons r in the hidden layer is not fixed. As such to achieve a desired precision one may take an excessive number of neurons. This, in turn, gives rise to the problem of complexity (see above).

Our approach to the problem of approximation by single hidden layer feedforward networks is different and quite simple. We consider networks (1.1) defined on $\mathbb{R}$ with a limited number of neurons (r is fixed!) in a hidden layer and ask the following fair question: is it possible to construct a well behaved (that is, sigmoidal, smooth, monotone, etc.) universal activation function providing approximation to arbitrary continuous functions on any compact set in $\mathbb{R}$ within any degree of precision? We show that this is possible even in the case of a feedforward network with only one neuron in its hidden layer (that is, in the case $r = 1$). The basic form of our theorem claims that there exists a smooth, sigmoidal, almost monotone activation function σ with the property: for each univariate continuous function f on the unit interval and any positive ε one can chose three numbers c_0, c_1 and θ such that the function $c_0 + c_1\sigma(t - \theta)$ gives ε-approximation to f. It should be remarked that we prove not only the existence result but also give an algorithm for constructing the mentioned universal sigmoidal function. For a wide class of Lipschitz continuous functions we also give an algorithm for evaluating the numbers c_0, c_1 and θ.

For numerical experiments we used SageMath [24]. We wrote a code for creating the graph of σ and computing $\sigma(t)$ at any reasonable $t \in \mathbb{R}$. The code is open-source and available at `http://sites.google.com/site/njguliyev/papers/sigmoidal`.

2. The theoretical result

We begin this section with the definition of a λ-increasing (λ-decreasing) function. Let λ be any nonnegative number. A real function f defined on (a, b) is called λ-increasing (λ-decreasing) if there exists an increasing (decreasing) function $u\colon (a, b) \to \mathbb{R}$ such that $|f(x) - u(x)| \leq \lambda$, for all $x \in (a, b)$. If u is strictly increasing (or strictly decreasing), then the above function f is called a λ-strictly increasing (or λ-strictly decreasing) function. Clearly, 0-monotonicity coincides with the usual concept of monotonicity and a λ_1-increasing function is λ_2-increasing if $\lambda_1 \leq \lambda_2$.

The following theorem is valid.

Theorem 2.1. *For any positive numbers α and λ, there exists a $C^\infty(\mathbb{R})$, sigmoidal activation function $\sigma\colon \mathbb{R} \to \mathbb{R}$ which is strictly increasing on $(-\infty, \alpha)$, λ-strictly increasing on $[\alpha, +\infty)$, and satisfies the following property: For any finite closed interval $[a, b]$ of $\mathbb{R}$ and any $f \in C[a, b]$ and $\varepsilon > 0$ there exist three real numbers c_0, c_1 and θ for which*

$$\left| f(t) - c_1 \sigma\left(\frac{\alpha}{b-a}t - \theta\right) - c_0 \right| < \varepsilon$$

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

Proof. Let α be any positive number. Divide the interval $[\alpha, +\infty)$ into the segments $[\alpha, 2\alpha]$, $[2\alpha, 3\alpha]$, ... Let $h(t)$ be any strictly increasing, infinitely differentiable function on $[\alpha, +\infty)$ with the properties

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

The existence of a strictly increasing smooth function satisfying these properties is easy to verify. Note that from conditions (1)–(3) it follows that any function $w(t)$ satisfying the inequality $h(t) < w(t) < 1$ for all $t \in [\alpha, +\infty)$, is λ-strictly increasing and $w(t) \to 1$, as $t \to +\infty$.

We are going to construct σ obeying the required properties in stages. Let $\{u_n(t)\}_{n=1}^\infty$ be the sequence of all polynomials with rational coefficients defined on $[0, 1]$. First, we define σ on the closed intervals $[(2m-1)\alpha, 2m\alpha]$, $m = 1, 2, \ldots$, as the function

$$\sigma(t) = a_m + b_m u_m\left(\frac{t}{\alpha} - 2m + 1\right), \quad t \in [(2m-1)\alpha, 2m\alpha],$$

or equivalently,

$$\sigma(\alpha t + (2m-1)\alpha) = a_m + b_m u_m(t), \quad t \in [0, 1], \tag{2.1}$$

where a_m and $b_m \neq 0$ are chosen in such a way that the condition

$$h(t) < \sigma(t) < 1 \tag{2.2}$$

holds for all $t \in [(2m-1)\alpha, 2m\alpha]$.

At the second stage we define σ on the intervals $[2m\alpha, (2m+1)\alpha]$, $m = 1, 2, \ldots$, so that it is in $C^\infty(\mathbb{R})$ and satisfies the inequality (2.2). Finally, in all of $(-\infty, \alpha)$ we define σ while maintaining the C^∞ strict monotonicity property, and also in such a way that $\lim_{t\to-\infty} \sigma(t) = 0$. We obtain from the properties of h and the condition (2.2) that $\sigma(t)$ is a λ-strictly increasing function on the interval $[\alpha, +\infty)$ and $\sigma(t) \to 1$, as $t \to +\infty$. Note that the construction of a σ obeying all the above conditions is feasible. We show this in the next section.

From (2.1) it follows that for each $m = 1, 2, \ldots$,

$$u_m(t) = \frac{1}{b_m}\sigma(\alpha t + (2m-1)\alpha) - \frac{a_m}{b_m}. \tag{2.3}$$

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 $u_m(t)$ of the above form such that

$$|g(t) - u_m(t)| < \varepsilon,$$

for all $t \in [0, 1]$. This together with (2.3) means that

$$|g(t) - c_1 \sigma(\alpha t - s) - c_0| < \varepsilon, \tag{2.4}$$

for some c_0, c_1, $s \in \mathbb{R}$ and all $t \in [0, 1]$.

Note that (2.4) proves our theorem for the unit interval $[0, 1]$. Using linear transformation it is not difficult to go from $[0, 1]$ to any finite closed 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 $g(t) = f(a + (b - a)t)$ is well defined on $[0, 1]$ and we can apply the inequality (2.4). Now using the inverse transformation $t = \frac{x-a}{b-a}$, we can write that

$$|f(x) - c_1 \sigma(wx - \theta) - c_0| < \varepsilon, \tag{2.5}$$

where $w = \frac{\alpha}{b-a}$ and $\theta = \frac{\alpha a}{b-a} + s$. The last inequality (2.5) completes the proof. $\square$

Remark 2.1. The idea of using a limited number of neurons in hidden layers of a feedforward network was first implemented by Maiorov and Pinkus [20]. They proved the existence of a sigmoidal, strictly increasing, analytic activation function such that two hidden layer neural networks with this activation function and a fixed number of neurons in each hidden layer can approximate any continuous multivariate function over the unit cube in $\mathbb{R}^n$. Note that the result is of theoretical value and the authors do not suggest constructing and using their sigmoidal function. Using the techniques developed in [20], we showed theoretically that if replace the demand of analyticity by smoothness, then the number of neurons in hidden layers can be reduced substantially (see [16]). We stress again that in both papers the algorithmic implementation of the obtained results is not discussed nor illustrated by numerical examples.

In the next section, we propose an algorithm for computing the above sigmoidal function σ at any point of the real axis. The code of this algorithm is available at http://sites.google.com/site/njguliyev/papers/sigmoidal. As examples, we include in the paper the graph of σ (see Figure 1) and a numerical table (see Table 1) containing several computed values of this function.

3. Algorithmic construction of the universal sigmoidal function

Step 1. *Definition of $h(t)$.*
Set

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

Note that this function satisfies the conditions 1)–3) in the proof of Theorem 2.1.

Step 2. *Enumerating the rationals.*
Let a_n be Stern's diatomic sequence:

$$a_1 = 1, \quad a_{2n} = a_n, \ a_{2n+1} = a_n + a_{n+1}, \ n = 1, 2, \dots .$$

It should be remarked that this sequence first appeared in print in 1858 [25] and has been the subject of many papers (see, e.g., [22] and the references therein).

The Calkin–Wilf [2] sequence $q_n := a_n/a_{n+1}$ contains every positive rational number exactly once and hence the sequence

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

is the enumeration of all the rational numbers. It is possible to calculate q_n and r_n directly. Let

$$(\underbrace{11\ldots1}_{f_k}\underbrace{00\ldots0}_{f_{k-1}}\underbrace{11\ldots1}_{f_{k-2}}\ldots\underbrace{00\ldots0}_{f_1}\underbrace{11\ldots1}_{f_0})_2$$

be the binary code of n. Here, $f_0, f_1, \ldots, f_k$ show the number of 1-digits, 0-digits, 1-digits, etc., respectively, starting from the end of the binary code. Note that f_0 can be zero. Then q_n equals the continued fraction

$$[f_0; f_1, \ldots, f_k] := f_0 + \cfrac{1}{f_1 + \cfrac{1}{f_2 + \cfrac{1}{\ddots + \cfrac{1}{f_k}}}}. \tag{3.1}$$

The calculation of r_n is reduced to the calculation of $q_{n/2}$, if n is even and $q_{(n+1)/2}$, if n is odd.

Step 3. *Enumerating the polynomials with rational coefficients.*

It is clear that every positive rational number determines a unique finite continued fraction $[n_0; n_1, \ldots, n_l]$ with $n_0 \geq 0$, $n_1, \ldots, n_{l-1} \geq 1$ and $n_l \geq 2$.

Since each non-zero polynomial with rational coefficients can uniquely be written as $r_{k_0} + r_{k_1}t + \ldots + r_{k_d}t^d$, where $r_{k_d} \neq 0$ (i.e. $k_d > 0$), we have the following bijection between the set of all non-zero polynomials with rational coefficients and the set of all positive rational numbers:

$$r_{k_0} + r_{k_1}t + \ldots + r_{k_d}t^d \mapsto [k_0; k_1 + 1, \ldots, k_d + 1]$$

We define $u_1(t) := 0$ and

$$u_m(t) := r_{n_0} + r_{n_1-1}t + \ldots + r_{n_d-1}t^d, \quad m = 2, 3, \ldots,$$

where $q_{m-1} = [n_0; n_1, \ldots, n_d]$.

Step 4. *Construction of σ on $[(2m-1)\alpha, 2m\alpha]$.*

Set $M := h((2m+1)\alpha)$. Besides, for each polynomial $u_m(t) = d_0 + d_1t + \ldots + d_kt^k$, set

$$A_1 := d_0 + \frac{d_1 - |d_1|}{2} + \ldots + \frac{d_k - |d_k|}{2} \tag{3.2}$$

and

$$A_2 := d_0 + \frac{d_1 + |d_1|}{2} + \ldots + \frac{d_k + |d_k|}{2}. \tag{3.3}$$

It is not difficult to verify that

$$A_1 \leq u_m(t) \leq A_2, \quad t \in [0, 1].$$
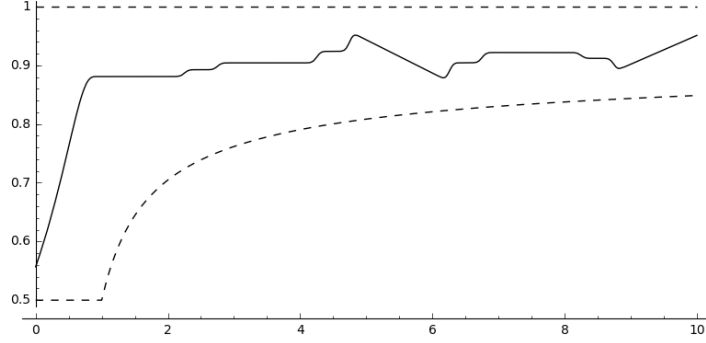
If u_m is constant, then we put

$$\sigma(t) := \frac{1 + M}{2}.$$

Otherwise, we define $\sigma(t)$ as the function

$$\sigma(t) = a_m + b_m u_m\left(\frac{t}{\alpha} - 2m + 1\right), \quad t \in [(2m-1)\alpha, 2m\alpha], \tag{3.4}$$

where

$$a_m := \frac{(1 + 2M)A_2 - (2 + M)A_1}{3(A_2 - A_1)}, \qquad b_m := \frac{1 - M}{3(A_2 - A_1)}. \tag{3.5}$$

FIGURE 1. The graph of σ on $[0, 10]$ ($\alpha = 1$, $\lambda = 1/2$)

t	σ	t	σ	t	σ	t	σ	t	σ
α	0.88087	100	0.95548	200	0.96034	0	0.55682	-100	0.00868
10	0.95095	110	0.94162	210	0.96064	-10	0.07655	-110	0.00790
20	0.95879	120	0.97124	220	0.94790	-20	0.04096	-120	0.00725
30	0.96241	130	0.97163	230	0.96119	-30	0.02796	-130	0.00670
40	0.96464	140	0.94397	240	0.97430	-40	0.02122	-140	0.00623
50	0.94931	150	0.97230	250	0.95743	-50	0.01710	-150	0.00581
60	0.96739	160	0.97259	260	0.97461	-60	0.01432	-160	0.00545
70	0.93666	170	0.94573	270	0.95793	-70	0.01232	-170	0.00514
80	0.96910	180	0.94622	280	0.94979	-80	0.01081	-180	0.00485
90	0.93951	190	0.94669	290	0.96670	-90	0.00963	-190	0.00460

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

Note that a_m, b_m are the coefficients of the linear function $y = a_m + b_m x$ mapping the closed interval $[A_1, A_2]$ onto the closed interval $[(1 + 2M)/3, (2 + M)/3]$. Thus,

$$h(t) < M < \frac{1 + 2M}{3} \leq \sigma(t) \leq \frac{2 + M}{3} < 1, \qquad (3.6)$$

for all $t \in [(2m - 1)\alpha, 2m\alpha]$.

Step 5. *Construction of σ on $[2m\alpha, (2m + 1)\alpha]$.*

To define σ on the intervals $[2m\alpha, (2m + 1)\alpha]$ we will use the *smooth transition function*

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

where

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

It is easy to see that $\beta_{a,b}(t) = 1$ for $t \leq a$, $\beta_{a,b}(t) = 0$ for $t \geq b$ and $0 < \beta_{a,b}(t) < 1$ for $a < t < b$. Set

$$K := \frac{\sigma(2m\alpha) + \sigma((2m + 1)\alpha)}{2}$$

Since both $\sigma(2m\alpha)$ and $\sigma((2m+1)\alpha)$ belong to the interval $(M, 1)$, we obtain that $K \in (M, 1)$.

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

$$\left| a_m + b_m u_m \left(\frac{t}{\alpha} - 2m + 1 \right) - (a_m + b_m u_m(1)) \right| \leq \varepsilon, \quad t \in [2m\alpha, 2m\alpha + \delta]. \quad (3.7)$$

Let us show how one can choose this δ. If u_m is constant, it is sufficient to take $\delta := \alpha/2$. If u_m is not constant, we take

$$\delta := \min \left\{ \frac{\varepsilon\alpha}{b_m C}, \frac{\alpha}{2} \right\},$$

where C is a number satisfying $|u'_m(t)| \leq C$ for $t \in (1, 1.5)$. Now define σ on the first half of the interval $[2m\alpha, (2m+1)\alpha]$ as the function

$$\begin{aligned} \sigma(t) := K &- \beta_{2m\alpha, 2m\alpha+\delta}(t) \\ &\times \left(K - a_m - b_m u_m \left(\frac{t}{\alpha} - 2m + 1 \right) \right), \quad t \in \left[2m\alpha, 2m\alpha + \frac{\alpha}{2} \right]. \end{aligned} \quad (3.8)$$

Let us verify that $\sigma(t)$ satisfies the condition (2.2). Indeed, if $2m\alpha + \delta \leq t \leq 2m\alpha + \frac{\alpha}{2}$, then there is nothing to prove, since $\sigma(t) = K \in (M, 1)$. If $2m\alpha \leq t < 2m\alpha + \delta$, then $0 < \beta_{2m\alpha, 2m\alpha+\delta}(t) \leq 1$ and hence from (3.8) it follows that for each $t \in [2m\alpha, 2m\alpha + \delta)$, $\sigma(t)$ is between the numbers K and $A(t) = a_m + b_m u_m \left(\frac{t}{\alpha} - 2m + 1 \right)$. On the other hand, from (3.7) we obtain that

$$a_m + b_m u_m(1) - \varepsilon \leq A(t) \leq a_m + b_m u_m(1) + \varepsilon,$$

which together with (3.4) and (3.6) yields that $A(t) \in \left[\frac{1+2M}{3} - \varepsilon, \frac{2+M}{3} + \varepsilon \right]$, for $t \in [2m\alpha, 2m\alpha + \delta)$. Since $\varepsilon = (1-M)/6$, the inclusion $A(t) \in (M, 1)$ is valid. Now since both K and $A(t)$ belong to $(M, 1)$, we finally conclude that

$$h(t) < M < \sigma(t) < 1, \quad \text{for } t \in \left[2m\alpha, 2m\alpha + \frac{\alpha}{2} \right].$$

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

$$\begin{aligned} \sigma(t) := K &- (1 - \beta_{(2m+1)\alpha-\bar{\delta},(2m+1)\alpha}(t)) \\ &\times \left(K - a_{m+1} - b_{m+1} u_{m+1} \left(\frac{t}{\alpha} - 2m - 1 \right) \right), \quad t \in \left[2m\alpha + \frac{\alpha}{2}, (2m+1)\alpha \right], \end{aligned}$$

where

$$\bar{\delta} := \min \left\{ \frac{\bar{\varepsilon}\alpha}{b_{m+1}\overline{C}}, \frac{\alpha}{2} \right\}, \qquad \bar{\varepsilon} := \frac{1 - h((2m+3)\alpha)}{6}, \qquad \overline{C} \geq \sup_{[-0.5, 0]} |u'_{m+1}(t)|.$$

One can easily verify, as above, that the constructed $\sigma(t)$ satisfies the condition (2.2) on $[2m\alpha, 2m\alpha + \alpha/2]$ and

$$\sigma \left(2m\alpha + \frac{\alpha}{2} \right) = K, \qquad \sigma^{(i)} \left(2m\alpha + \frac{\alpha}{2} \right) = 0, \quad i = 1, 2, \dots.$$

Step 6. *Construction of σ on $(-\infty, \alpha)$.*
Finally, we put

$$\sigma(t) := \left(1 - \widehat{\beta}(\alpha - t) \right) \frac{1 + h(3\alpha)}{2}, \quad t \in (-\infty, \alpha).$$

It is not difficult to verify that σ is a strictly increasing, smooth function on $(-\infty, \alpha)$. Note also that $\sigma(t) \to \sigma(\alpha) = (1 + h(3\alpha))/2$ (see Step 4), as t tends to α from the left and $\sigma^{(i)}(\alpha) = 0$, for $i = 1, 2, \ldots$.

Step 6 completes the construction of the universal activation function σ, which satisfies Theorem 2.1.

4. The algorithm for evaluating the numbers c_0, c_1 and θ

Although Theorem 2.1 is valid for all continuous functions, in practice it is quite difficult to calculate algorithmically c_0, c_1 and θ in Theorem 2.1 for badly behaved continuous functions. The main difficulty arises while attempting to design an efficient algorithm for the construction of a best approximating polynomial within any given degree of accuracy. But for certain large classes of well behaved functions, the computation of the above numbers is doable. In this section, we show this for the class of Lipschitz continuous functions.

Assume that f is a Lipschitz continuous function on $[a, b]$ with a Lipschitz constant L. In order to find the parameters c_0, c_1 and θ algorithmically, it is sufficient to perform the following steps.

Step 1. *Going to the unit interval.*

Consider the function $g(t) = f(a + (b - a)t)$, which is Lipschitz continuous on $[0, 1]$ with a Lipschitz constant $L_1 = L(b - a)$. Denote by

$$B_n(x) := \sum_{k=0}^{n} g\left(\frac{k}{n}\right) \binom{n}{k} x^k (1 - x)^{n-k}$$

the n-th Bernstein polynomial of the function g. Let $\varepsilon > 0$ be given.

Step 2. *Finding the position of a given rational number.*

Define the functions

$$p_q \colon \mathbb{Q}_+ \to \mathbb{N}, \qquad p_r \colon \mathbb{Q} \to \mathbb{N} \cup \{0\},$$

which return the positions of a positive rational number and a rational number in the sequences $\{q_n\}$ and $\{r_n\}$, respectively (see Section 3). We start with the computation of p_q. Let q be a positive rational number. If (3.1) is the continued fraction representation of q with k even (we may always consider $[f_0; f_1, \ldots, f_k - 1, 1]$ instead of $[f_0; f_1, \ldots, f_k]$ if needed), then the binary representation of the position $p_q(q)$ of q in the Calkin–Wilf sequence is

$$p_q(q) = (\underbrace{11 \ldots 1}_{f_k} \underbrace{00 \ldots 0}_{f_{k-1}} \underbrace{11 \ldots 1}_{f_{k-2}} \ldots \underbrace{00 \ldots 0}_{f_1} \underbrace{11 \ldots 1}_{f_0})_2.$$

Now we can easily find $p_r(r)$ by the formula

$$p_r(r) = \begin{cases} 0, & r = 0, \\ 2p_q(r), & r > 0, \\ 2p_q(-r) - 1, & r < 0. \end{cases}$$

Step 3. *Finding $n \in \mathbb{N}$ such that $|B_n(x) - g(x)| \leq \varepsilon/2$, $x \in [0, 1]$.*

We use the inequality (see [23])

$$|B_n(x) - g(x)| \leq \chi \omega \left(\frac{1}{\sqrt{n}}\right), \quad x \in [0, 1],$$

where
$$\chi = \frac{4306 + 837\sqrt{6}}{5832} = 1.089887\ldots.$$
Since
$$\omega\left(\frac{1}{\sqrt{n}}\right) \leq \frac{L_1}{\sqrt{n}}$$
it is sufficient to take
$$n := \left\lceil \left(\frac{2\chi L_1}{\varepsilon}\right)^2 \right\rceil,$$
where $\lceil \cdot \rceil$ is the ceiling function defined as $\lceil x \rceil := \min\{k \in \mathbb{Z} \mid k \geq x\}$.

Step 4. *Finding a polynomial p with rational coefficients such that $|p(x) - B_n(x)| \leq \varepsilon/2$, $x \in [0,1]$.*

If $B_n(x) = a_0 + a_1 x + \ldots + a_k x^k$ then it is sufficient to choose $d_0, d_1, \ldots, d_k \in \mathbb{Q}$ such that
$$|a_0 - d_0| + |a_1 - d_1| + \ldots + |a_k - d_k| \leq \frac{\varepsilon}{2}$$
and put $p(x) := d_0 + d_1 x + \ldots + d_k x^k$.

Step 5. *Finding $m \in \mathbb{N}$ such that $u_m \equiv p$.*

For the definition of u_m see Section 3. If $p \equiv 0$ then clearly $m = 1$. Otherwise let $n_i := p_r(d_i)$ be the positions of the numbers d_i in the sequence $\{r_n\}$. Then
$$m = p_q([n_0; n_1 + 1, \ldots, n_k + 1]) + 1.$$

Step 6. *Evaluating the numbers c_0, c_1 and s in (2.4).*

Set $s := (1 - 2m)\alpha$. If $u_m = p$ is constant then we put
$$c_1 := 1, \qquad c_0 := d_0 - \frac{1 + h((2m+1)\alpha)}{2}.$$
If $u_m = p$ is not constant then we put
$$c_1 := \frac{1}{b_m}, \qquad c_0 := -\frac{a_m}{b_m},$$
where a_m and b_m are computed using the formulas (3.2), (3.3) and (3.5).

Step 7. *Evaluating the numbers c_0, c_1 and θ.*

In this step, we return to our original function f and calculate the numbers c_0, c_1 and θ (see Theorem 2.1). The numbers c_0 and c_1 have been calculated above (they are the same for both g and f). In order to find θ, we use the formula $\theta = \frac{\alpha a}{b-a} + s$.

Remark 4.1. Note that some computational difficulties may arise while implementing the above algorithm in standard computers. For some functions, the index m of a polynomial u_m in Step 5 may be extraordinarily large. In this case, a computer is not capable of producing this number, hence the numbers c_0, c_1 and θ.

REFERENCES

[1] A. R. Barron, Universal approximation bounds for superposition of a sigmoidal function, *IEEE Trans. Information Theory* **39** (1993), 930–945.

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

[3] F. Cao, T. Xie, and Z. Xu, The estimate for approximation error of neural networks: A constructive approach, *Neurocomputing* **71** (2008), no. 4–6, 626–630.

[4] S. M. Carroll and B. W. Dickinson, Construction of neural nets using the Radon transform, in *Proceedings of the IEEE 1989 International Joint Conference on Neural Networks*, 1989, Vol. 1, IEEE, New York, pp. 607–611.

[5] T. Chen, H. Chen, and R. Liu, A constructive proof of Cybenko's approximation theorem and its extensions, In *Computing Science and Statistics*, Springer-Verlag, 1992, pp. 163–168.

[6] C. K. Chui and X. Li, Approximation by ridge functions and neural networks with one hidden layer, *J. Approx. Theory* **70** (1992), 131–141.

[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, Signals, and Systems* **2** (1989), 303–314.

[9] K. Funahashi, On the approximate realization of continuous mapping by neural networks, *Neural Networks* **2** (1989), 183–192.

[10] A. R. Gallant and H. White, There exists a neural network that does not make avoidable mistakes, in *Proceedings of the IEEE 1988 International Conference on Neural Networks*, 1988, Vol. 1, IEEE, New York, pp. 657–664.

[11] N. Hahm and B.I. Hong, Extension of localized approximation by neural networks, *Bull. Austral. Math. Soc.* **59** (1999), 121–131.

[12] K. Hornik, Approximation capabilities of multilayer feedforward networks, *Neural Networks* **4** (1991), 251–257.

[13] K. Hornik, M. Stinchcombe and H. White, Multilayer feedforward networks are universal approximators, *Neural Networks* **2** (1989), 359–366.

[14] B. Irie and S. Miyake, Capability of three-layered perceptrons, in *Proceedings of the IEEE 1988 International Conference on Neural Networks*, 1988, Vol. 1, IEEE, New York, pp. 641–648.

[15] 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.

[16] V. E. Ismailov, On the approximation by neural networks with bounded number of neurons in hidden layers, *J. Math. Anal. Appl.* **417** (2014), no. 2, 963–969.

[17] Y. Ito, Representation of functions by superpositions of a step or sigmoid function and their applications to neural network theory, *Neural Networks* **4** (1991), 385–394.

[18] L. K. Jones, Constructive approximations for neural networks by sigmoidal functions, *Proc. IEEE* **78** (1990), 1586–1589. Correction and addition, *Proc. IEEE* **79** (1991), 243.

[19] 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.

[20] V. Maiorov and A. Pinkus, Lower bounds for approximation by MLP neural networks, *Neurocomputing* **25** (1999), 81–91.

[21] 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.

[22] S. Northshield, Stern's diatomic sequence $0, 1, 1, 2, 1, 3, 2, 3, 1, 4, \ldots$, *Amer. Math. Monthly* **117** (2010), no. 7, 581–598.

[23] P. C. Sikkema, Der Wert einiger Konstanten in der Theorie der Approximation mit Bernstein-Polynomen, *Numer. Math.* **3** (1961), 107–116.

[24] W. A. Stein et al., *Sage Mathematics Software (Version 6.10)*, The Sage Developers, 2015, `http://www.sagemath.org`.

[25] M. A. Stern, Über eine zahlentheoretische Funktion, *J. Reine Agnew. Math.* **55** (1858) 193–220.

[26] M. Stinchcombe and H. White, Approximating and learning unknown mappings using multilayer feedforward networks with bounded weights, in *Proceedings of the IEEE 1990 International Joint Conference on Neural Networks*, 1990, Vol. 3, IEEE, New York, 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`