Neural Networks 98 (2018) 296–304

Contents lists available at ScienceDirect

Neural Networks

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

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

Namig J. Guliyev *, Vugar E. Ismailov

*Institute of Mathematics and Mechanics, Azerbaijan National Academy of Sciences, 9 B. Vahabzadeh str., AZ1141, Baku, Azerbaijan*

A R T I C L E   I N F O

*Article history:*
Received 22 August 2017
Received in revised form 15 November 2017
Accepted 12 December 2017
Available online 18 December 2017

*Keywords:*
Feedforward neural network
Approximation
Hidden layer
Sigmoidal function
Activation function
Weight

A B S T R A C T

Single hidden layer feedforward neural networks (SLFNs) with fixed weights 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 proof is implemented by a step by step construction of a universal sigmoidal activation function. This function has nice properties such as computability, smoothness and weak monotonicity. The applicability of the obtained result is demonstrated in various numerical examples. Finally, we show that SLFNs with fixed weights cannot approximate all continuous multivariate functions.



## 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.

An SLFN with $r$ units in the hidden layer and input $\mathbf{x} = (x_1, \ldots, x_d)$ 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 $\theta_i$ and the coefficients $c_i$ are real numbers, and the activation function $\sigma$ 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., Chen and Chen, 1993; Chui and Li, 1992; Costarelli and Spigler, 2013; Cotter, 1990; Cybenko, 1989; Funahashi, 1989; Gallant and White, 1988; Hornik, 1991; Mhaskar and Micchelli, 1992). 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 (1993). They proved that a continuous activation function $\sigma$ 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 $\sigma$, provided that $\sigma$ is continuous. For a detailed review of these and many other results, see Pinkus (1999).

In many applications, it is convenient to take the activation function $\sigma$ 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., Cao and Xie, 2010; Chen and Cao, 2009; Chui and Li, 1992; Costarelli, 2015; Costarelli and Spigler, 2013; Costarelli and Vinti, 2016a, b, c, 2017; Cybenko, 1989; Funahashi, 1989; Gallant and White, 1988; Hahm and Hong, 2004; Iliev, Kyurkchiev, and Markov, 2017; Ito, 1992; Kůrková, 1992; Mhaskar and Micchelli, 1992). 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., Draghici, 2002; Ismailov, 2012, 2015; Ismailov and Savas, 2017; Jian, Yu, and Jinshou, 2010; Liao, Fang, and Nuttle, 2004). It was proved that SLFNs with some restricted set of weights still possess the universal

* Corresponding author.
*E-mail addresses:* njguliyev@gmail.com (N.J. Guliyev), vugaris@mail.ru (V.E. Ismailov).

https://doi.org/10.1016/j.neunet.2017.12.007
0893-6080/

approximation property. For example, Stinchcombe and White (1990) showed that SLFNs with a polygonal, polynomial spline or analytic activation function and a bounded set of weights have the universal approximation property. Using monotone sigmoidal functions, Ito (1992) investigated this property for networks with weights located only on the unit sphere. In Ismailov (2012, 2015) and Ismailov and Savas (2017), 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 (2004) 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, 2004*). *Assume $f$ is a continuous function on a finite segment $[a, b]$ of $\mathbb{R}$. Assume $\sigma$ 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 $\varepsilon$. The smaller the $\varepsilon$, 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 $\sigma$, 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, 2010 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 Cao and Xie (2010) that for the class of Lipschitz functions $\mathrm{Lip}_M(\alpha)$ with a Lipschitz constant $M$ and degree $\alpha$, 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 (2013). Taking the activation function $\sigma$ as a continuous, even and $2\pi$-periodic function, the authors of Lin et al. (2013) 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 $\varepsilon$. Note that all the weights are fixed equal to 1, and consequently do not depend on $\varepsilon$. 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 $\sigma$ for which $K$ and $n$ in the above theorem do not depend on the error $\varepsilon$. Moreover, we can take $K = 1$ and $n = 2$. That is, only parameters $c_i$ and $\theta_i$ depend on $\varepsilon$. 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.

To construct our sigmoidal function $\sigma$ and prove the main theorem (see Theorem 4.1), we extensively use so called *monic polynomials* with rational coefficients. To the best of our knowledge, the idea of using monic polynomials 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 (1999) 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 of Maiorov and Pinkus (1999) is of theoretical value and the authors of Maiorov and Pinkus (1999) did not suggest constructing and using their sigmoidal function. In our previous work Guliyev and Ismailov (2016), we exploited a sequence of all polynomials with rational coefficients to construct a new universal sigmoidal function. Note that in Guliyev and Ismailov (2016) the problem of fixing weights in approximation by neural networks was not considered. Although the construction in Guliyev and Ismailov (2016) was efficient in the sense of computation of that sigmoidal function, some serious difficulties appeared while computing an approximating neural network parameters for some relatively simple approximated functions (see Remark 2 in Guliyev and Ismailov, 2016). 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 us to approximate various functions with sufficiently small precision and obtain all the required parameters in practice (see the numerical results in Section 5).

## 2. Construction of a sigmoidal function

In this section, we construct algorithmically a sigmoidal function $\sigma$ which we use in our main result in the following section. Besides sigmoidality, we take care about smoothness and monotonicity of our $\sigma$ 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 $\lambda$-*increasing* (respectively, $\lambda$-*decreasing*) if there exists an increasing (respectively, decreasing) function $u : X \to \mathbb{R}$ such that $|f(x) - u(x)| \le \lambda$ for all $x \in X$ (see Guliyev and Ismailov, 2016). Obviously, 0-monotonicity coincides with the usual concept of monotonicity, and a $\lambda_1$-increasing function is $\lambda_2$-increasing if $\lambda_1 \le \lambda_2$.

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

1. Introduce the function

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

Note that this function is strictly increasing on $[d, +\infty)$ and satisfies the following properties: