

Universal Approximation by Ridge Computational Models and Neural Networks: A Survey *

Marcello Sanguineti

Department of Communications, Computer and System Sciences (DIST)

University of Genova – Via Opera Pia 13, 16145 Genova, Italy

marcello@dist.unige.it

Abstract

Computational models made up of linear combinations of ridge basis functions, widely used in machine learning and artificial intelligence, are considered. For such models, the literature on the so-called “universal approximation property” is surveyed. Different approaches, proof techniques, and tools are examined.

Keywords: nonlinear computational models, ridge computational units, universal approximation, complexity.

AMS subject classifications: 68-02, 68T05, 41-02, 41A30.

1 Introduction

A basic goal of computational learning is to approximate within a desired accuracy a functional relationship between inputs and outputs by using a simple model. Among the models most widely used in applications, there are those made up of linear combinations of computational units represented by ridge functions, i.e., functions that are constant along hyperplanes; these models include many widespread neural networks [1].

Ridge models have been successfully applied in a variety of areas, such as time-series forecasting, system identification, data mining, approximation of decision strategies, financial and business applications,

*This work was supported in part by a PRIN grant from the Italian Ministry for University and Research, Project “Models and Algorithms for Robust Network Optimization.”

pattern recognition, optimal traffic control, routing in telecommunications, etc. (see, e.g., [2, 3, 4, 5, 6, 7], the references therein, and the bibliographies in [8, 9]). All these problems share a common aspect: a multivariable input/output mapping has to be approximated. Experimental results and theoretical investigations have shown that computational models having the form of linear combinations of simple ridge functions with adjustable parameters (so-called “computational units”) can achieve surprisingly good performances (see [2, 5, 6, 7, 10, 11] and the references therein).

A basic question is the following: for what kind of ridge computational units is it possible to guarantee that there exists an arbitrarily close approximation for every function belonging to the family of interest in the application at hand (e.g., continuous or square-integrable functions)? Such a property is often called the “universal approximation property.” In mathematical terms, it corresponds to “density” in suitable function spaces. Although the density of a computational model does not imply efficiency, its lack with respect to spaces of functions commonly used in applications is a sign of limited capabilities.

There exists a vast literature on theoretical investigations of the universal approximation property for ridge computational models. This paper is a survey of this topic.

The paper is organized as follows. In Section 2, some basic differences between the structure of classical linear computational models and the structure of ridge computational models are discussed. Section 3 is devoted to the perceptron model, its fall, and its renaissance. Section 4 describes in detail five main phases of investigations of the universal approximation property for ridge computational models. Different proof techniques are examined in Section 5. Section 6 provides a final discussion of the price of universality and warns about the “curse of dimensionality.”

2 Computational models and approximation

2.1 The “universal approximation property”

The first question concerning a computational model is whether a sufficiently large number of computational units allow one to approximate up to every desired degree of accuracy all “reasonable” functions encountered in applications.

To compute or estimate the accuracy of approximation, it is natural to choose an “ambient” function space \mathcal{H} endowed with a norm $\|\cdot\|$ and to measure the distance between two functions $f, g \in \mathcal{H}$ by the quantity $\|f - g\|$. Mathematically, the capability of approximating up to every desired degree of accuracy all functions in a space \mathcal{H} with a norm $\|\cdot\|$ is called *density*. So, a set \mathcal{Y} in a normed linear space \mathcal{H} is dense in \mathcal{H} if, for every $f \in \mathcal{H}$ and every $\varepsilon > 0$, there exists $y \in \mathcal{Y}$ such that $\|f - y\| \leq \varepsilon$. An equivalent way to state density is in terms of closure. Given a subset \mathcal{Y} of a normed linear space \mathcal{H} , the *closure* of \mathcal{H} in the norm of \mathcal{H} is defined as $\text{cl}_{\mathcal{H}} \mathcal{Y} = \{f \in \mathcal{H} \mid \forall \varepsilon > 0 \exists g \in \mathcal{Y} \text{ such that } \|f - g\| < \varepsilon\}$. Then, the set \mathcal{Y} is dense in \mathcal{H} with the norm $\|\cdot\|$ if $\text{cl}_{\mathcal{H}} \mathcal{Y} = \mathcal{H}$, i.e., if the closure of \mathcal{Y} w.r.t. the norm $\|\cdot\|$ on \mathcal{H} is the whole space \mathcal{H} .

The choices of the space \mathcal{H} , its norm, and the type of computational model with the corresponding density property depend on the application at hand. In most cases, one is interested in the \mathcal{C} -density property and the \mathcal{L}_2 -density property. In this paper, we consider the density property in the spaces $\mathcal{L}_p(K, \mathbb{R}^m)$ and $\mathcal{C}(K, \mathbb{R}^m)$, where K is a compact subset of \mathbb{R}^d . A set \mathcal{Y} of functions has the $\mathcal{C}(K)$ - or $\mathcal{L}_p(K)$ - density property if and only if $\text{cl}_{\mathcal{C}(K)} \mathcal{Y} = \mathcal{C}(K)$ and $\text{cl}_{\mathcal{L}_p(K)} \mathcal{Y} = \mathcal{L}_p(K)$, respectively. When the normed linear space is the space of continuous functions with the supremum norm, in neural network terminology the density property is also called the *universal approximation property*. The closure in $\mathcal{C}(K)$ (i.e., with respect to the supremum norm) is also called *uniform closure* [12, p. 149].

2.2 Linear and nonlinear computational models

Linear computational models implement linear combinations of a set of n fixed computational units. So, they correspond to sets $\{A_n\}$ that are finite-dimensional subspaces of a linear space \mathcal{H} . If the computational units are linearly independent,¹ then n is the dimension of the subspace that they span, otherwise the subspace generated by the computational units has a dimension less than n . For example, if algebraic polynomials of order at most $n - 1$ are considered, an n -dimensional subspace is generated by the first n elements of the set $\{x^{i-1} : i \in \mathbb{N}_+\}$. Summing up, linear computational models correspond to linear subspaces A_n , i.e., linear combinations of elements of a fixed set of functions.

$$A_n = \left\{ \sum_{i=1}^n c_i \phi_i(\cdot) : c_i \in \mathbb{R} \right\}.$$

Hence, the number of parameters in linear computational models with linearly independent computational units is equal to the number n of such functions.

In this survey, we define as *nonlinear computational models* the linear combinations of functions with a fixed structure, in which there is a certain number of “free” parameters to be adjusted; sometimes we shall call such functions “parametrized computational units.” To clarify this point, we give the following example, which, despite its simplicity, contains all the main features of interest.

The approximation of one-variable real-valued functions by sine trigonometric polynomials of degree at most n corresponds to the approximation scheme obtained by linear combinations of the first n elements of the set $\{\sin(2\pi i x) : i \in \mathbb{N}\}$. The linear combinations of such first n elements represent a linear computational model with n fixed computational units equal to sines with frequencies multiple of 2π .

Instead of considering only frequencies multiple of 2π , suppose to take all possible frequencies into account, i.e., let us define the set of functions $\{\sin(2\pi i x) : i \in \mathbb{R}_+\}$. For each choice of $i \in \mathbb{R}_+$,

¹Recall that n elements f_1, \dots, f_n of a real linear space H are called *linearly independent* if there exist no numbers $c_1, \dots, c_n \in \mathbb{R}$ such that $c_1 f_1 + \dots + c_n f_n = 0$ with $\sum_{j=1}^n |c_j| > 0$. If there exist $c_1, \dots, c_n \in \mathbb{R}$ such that the above two conditions hold, then the elements f_1, \dots, f_n are called *linearly dependent*; in such a case, every $f_j, j = 1, \dots, n$, can be expressed as a linear combination of the other $n - 1$ elements.

a sine with a different frequency is generated. Such a sine can be thought of as an element of a set of computational units, obtained by varying the free parameter $i \in \mathbb{R}$ to which a frequency $2\pi i$ corresponds. In other words, approximating functions are constructed as linear combinations of all possible n -tuples of the set of sines with arbitrary frequencies, each corresponding to a choice of the free parameter $i \in \mathbb{R}_+$. The total number of parameters is not equal to the number n of computational units any more, as in the case of linear approximation by sines, but is given by $2n$ (n coefficients of the linear combination, plus n frequencies). In the case of functions of d variables, there are $n(d+1)$ parameters, d being the number of free parameters in the inner product $i^\top x$, where $i \in \mathbb{N}_+^d$, $x \in \mathbb{R}^d$, for each sine with the frequency vector $2\pi i \in \mathbb{R}^d$, and $^\top$ denotes transposition.

Taking the hint from the above example, one can easily understand the structure of nonlinear computational models with n computational units $\varphi(\cdot, \cdot) : \mathbb{R}^d \times A \rightarrow \mathbb{R}$, a parameter set $A \subseteq \mathbb{R}^k$, and a single linear output unit: they generate all functions that can be written as $\sum_{i=1}^n c_i \varphi(\cdot, \kappa_i)$, where $\kappa_i \in A \subseteq \mathbb{R}^k$, i.e., all functions belonging to the set

$$A_n = \left\{ \sum_{i=1}^n c_i \varphi(\cdot, \kappa_i) : c_i \in \mathbb{R}, \kappa_i \in A \subseteq \mathbb{R}^k \right\}. \quad (1)$$

2.3 Relationships between the \mathcal{C} - and \mathcal{L}_p -density properties

As, for every compact set $K \subset \mathbb{R}^d$ and every $1 \leq p < \infty$, $\mathcal{C}(K)$ is dense in $\mathcal{L}_p(K)$ ([13, pp. 28-31] and [14, p. 151]), by the following two-step density argument one concludes that the density in $\mathcal{C}(K)$ implies the density in $\mathcal{L}_p(K)$.

Let A_n be the set of functions that can be computed by a model with a given type of computational units, and suppose that A_∞ is dense in $\mathcal{C}(K)$. Since, for every $p \in [1, \infty)$, $\mathcal{C}(K)$ is dense in $\mathcal{L}_p(K)$, for every $f \in \mathcal{L}_p(K)$ and $\varepsilon > 0$ there exists $\zeta \in \mathcal{C}(K)$ such that $\|f - \zeta\|_p \leq \varepsilon/2$. For such a ζ , there exist $n \in \mathbb{N}_+$ and a function $f_n \in A_n$ with n computational units such that $\|\zeta - f_n\|_\infty \leq \varepsilon/(2\mu(K))$, where $\mu(K)$ denotes the Lebesgue measure of the set K , and so $\|\zeta - f_n\|_p \leq \|\zeta - f_n\|_\infty \mu(K) \leq \varepsilon/2$. Thus, $\|f - f_n\|_p \leq \|f - \zeta\|_p + \|\zeta - f_n\|_p \leq \varepsilon/2 + \varepsilon/2 = \varepsilon$. Hence, A_n is dense in $\mathcal{L}_p(K)$. So, conditions guaranteeing the \mathcal{C} -density property also guarantee the \mathcal{L}_p one. However, without resorting to the two-step density argument outlined above, proof techniques developed “ad hoc” might guarantee the \mathcal{L}_p -density property under weaker assumptions on the computational units.

2.4 Ridge computational models

Ridge computational models correspond to the case in which the d -variable computational unit φ is obtained from a one-variable “mother function” h , composed with the inner product in \mathbb{R}^d . So, the ridge construction “shrinks” the d -dimensional vector x into a one-dimensional variable by the inner product, i.e.,

$$\varphi(x, \kappa_i) = h(x^\top \alpha_i + \beta_i), \tag{2}$$

where $\kappa_i \triangleq \text{col}(\alpha_i, \beta_i) \in \mathbb{R}^{d+1}$ and $h : \mathbb{R} \rightarrow \mathbb{R}$ is fixed. Each function of the form (2) is constant along the parallel hyperplanes $x^\top \alpha_i + \beta_i = c_i$, where $c_i \in \mathbb{R}$. Functions constant along hyperplanes are known as *ridge functions*; the vectors $\alpha_i \in \mathbb{R}^d \setminus \{0\}$ are called *directions*. Thus, each ridge function results from the composition of a multivariable function having a particularly simple form, i.e., the inner product $x^\top \alpha$ on \mathbb{R}^d , with an arbitrary function dependent on a unique variable. Simple examples of widespread ridge functions are $e^{x^\top \alpha}$ and $(x^\top \alpha)^k$.

As noted in [15], the name “ridge function” is quite recent, having being coined by Logan and Shepp in 1975 [16]. The reason for introducing such functions in [16], which is a seminal paper in computerized tomography, was the reconstruction of a multivariable function from the values of its integrals along certain planes or lines. If such planes or lines are parallel to one another, then each of the above-mentioned integrals can be regarded as a ridge function in a certain direction. However, ridge functions have been studied for a long time under the name of *plane waves* ([17], [18]), due to problems from physics. It should also be noted that ridge functions are studied in statistics, where they are often called *projection pursuit*.² Among papers on approximation by ridge functions and consequences on neural-network approximation, see [21, 15, 14, 22, 23, 24, 25, 26].

I denote by

$$\mathcal{P}^d(h) = \left\{ f : \mathbb{R}^d \rightarrow \mathbb{R} \mid f(x) = \sum_{i=1}^n c_i h(x^\top \alpha_i + \beta_i), \alpha_i \in \mathbb{R}^d, c_i, \beta_i \in \mathbb{R}, n \in \mathbb{N}_+ \right\} \tag{3}$$

the set of functions computed by linear combinations of computational units based on the ridge construction with a mother function h .

As regards the mother function h used to construct the ridge functions (2), a typical choice is a *sigmoid*, i.e., a bounded measurable function σ on the real line such that $\lim_{z \rightarrow +\infty} \sigma(z) = 1, \lim_{z \rightarrow -\infty} \sigma(z) = 0$ ³. Sigmoidal functions widely used in applications are:

- the *Heaviside function* (also known as *threshold function* in the neural-network literature)

$$h(t) = \begin{cases} 0, & \text{if } t < 0 \\ 1, & \text{if } t \geq 0; \end{cases}$$

- the *logistic sigmoid* $h(t) = \frac{1}{1+e^{-t}}$;

²Recall that projection pursuit algorithms investigate the approximation of a d -variable function by functions of the form $\sum_{i=1}^k g_i(x^\top \alpha_i)$, where $\alpha_i \in \mathbb{R}^d$ and $g_i : \mathbb{R} \rightarrow \mathbb{R}$ have to be suitably chosen; see, e.g., [19] and [20]).

³Here we use perhaps the most widespread definition of sigmoidal function. However, in the literature, there is a certain lack of consistency in the terminology; for example, some authors require also the continuity and/or monotonicity (or even strict monotonicity) of σ on the whole \mathbb{R} .

- $h(t) = \tanh(t/2)$, obtained from the logistic sigmoid by a shift;
- the piecewise-linear function
$$h(t) = \begin{cases} 0, & \text{if } t \leq -1 \\ \frac{t+1}{2}, & \text{if } -1 \leq t \leq 1 \\ 1, & \text{if } t \geq 1 \end{cases};$$
- the *Gaussian sigmoid* $h(t) = \frac{1}{(2\pi)^{1/2}} \int_{-\infty}^t e^{-s^2/2} ds$;
- the arctan sigmoid $h(t) = \frac{1}{\pi} \arctan(t) + \frac{1}{2}$.

2.5 Single-output models

Since for all positive integers d and m , a function $f : \mathbb{R}^d \rightarrow \mathbb{R}^m$ can be implemented by m mappings $f_j : \mathbb{R}^d \rightarrow \mathbb{R}$, $j = 1, \dots, m$, in this paper we consider only single-output models, i.e., $m = 1$. To simplify the notation, we also write $\mathcal{C}(K)$ and $\mathcal{L}_p(K)$ instead of $\mathcal{C}(K, \mathbb{R})$ and $\mathcal{L}_p(K, \mathbb{R})$, respectively. For brevity, sometimes we write \mathcal{C} - or \mathcal{L}_p - “density property on compacta,” meaning that the corresponding density property holds for real-valued functions defined on any compact subset of \mathbb{R}^d .

Although the extension to the case $m > 1$ is formally straightforward, as it simply requires using m computational models in parallel, one for each component of the output, this way of proceeding takes into account only the point of view of density, i.e., of an arbitrarily good approximation. As each component of the output is treated as an independent scalar one, in this way one gives up considering how different computational units can be involved in approximating the same component of a vector function, how a practical learning algorithm for finding the optimal values of the parameters can take advantage of this, and how the minimal number of parameters necessary to guarantee the desired approximation accuracy can be made smaller by taking into account the mutual dependence of the various components of the output.

2.6 A look at terminology

Let us discuss the connections between the terminology “ridge computational model” and the neural-network parlance.

Since the representation (1) can be considered as a neural network with d inputs, n computational units in the so-called “hidden layer”, and one linear output unit, the parametrized-basis computational models (1) are also called *one-hidden-layer networks (OHL networks)*. So, (3) is a d -variable single-output *ridge OHL network*.

The reason why only one hidden layer is considered is pragmatic: we shall see that one hidden layer is sufficient to achieve \mathcal{C} - and \mathcal{L}_p -densities. Relatively little is known about the approximation properties and the advantages of ridge computational models using more hidden layers. We refer the reader to [14, Section 7].

The model known in the neural-network community as “multilayer feedforward perceptron” consists of a finite sequence of layers, each containing a finite number of computational units. In each layer, every unit is connected to each unit of the subsequent layer. The computational units are also called *activation functions* or *neurons*, and the connections between neurons are known as *synapses*. The elements of the parameter vector α_i (see (2)) are called *weights*; the parameters β_i are called *thresholds* or *biases*. The term “feedforward” is motivated by the fact that in such a model the information flows from each layer to the subsequent one. The first layer, called the *input layer*, consists of the inputs to the network, and the last layer, providing the output signals, is called the *output layer*. In between there are the so-called *hidden layers*, whose computational units are the *hidden neurons*. Then, the output $x_j^{(l+1)}$ of the j -th unit of the $(l + 1)$ -th layer is given by

$$x_j^{(l+1)} = h(x^{(l)\top} \alpha_j^{(l)} + \beta_j^{(l)}).$$

Of course, many generalizations of the multilayer feedforward perceptron are possible. For example, the activation functions might be different in each layer (although the use of the same unit is a common choice) and the architecture might be changed to allow different links between the various units.

Multilayer feedforward perceptrons are usually classified on the basis of the number of hidden neurons. Unfortunately, in this respect, the neural network terminology sometimes is not consistent. For example, the term “multilayer perceptron” was introduced by Rosenblatt [27] to indicate the “no-hidden-layer model” with the Heaviside activation function, by analogy to biological models. However, sometimes such a model is confusingly called the “single layer feedforward network.”

From a mathematical perspective, applying an activation function to the output layer, especially if such an activation function is bounded, might be unnecessarily restrictive. Indeed, a widespread choice in the neural-network literature and applications is to use no activation function at the output layer but just to perform a weighted sum of the outputs of the hidden layer. This choice leads one naturally to ridge OHL networks. Thus, when a ridge construction is used, such networks correspond to multilayer perceptrons with one hidden layer, with which the reader with a neural-network background is familiar. Ridge computational models are sometimes called “one-hidden-layer feedforward perceptrons with linear output units,” as the output is obtained by a linear combination of the outputs of the hidden layer (i.e., the activation functions of the output layer are linear). Further, some authors call them “three-layer feedforward networks with linear output units” [28] or “three-layered perceptrons” [29], as in the terminology they account also for the input and output layers. Also the term “two-layer feedforward networks with linear output units” can be found, accounting for the hidden layer and the output one. For other variations of the terminology, see [30, footnote 6, p. 1421].

In the following, we use the terms “OHL network” and “ridge OHL network” to refer to computational models of the forms (1) and (3), respectively.

3 The perceptron: fall and renaissance

It was the very lack of density to determine the failure of the so-called “no-hidden-layer perceptron model” (indeed, such a model is no longer used, except when linear separation problems are dealt with). Following [14], this can be easily explained as follows. Let us consider the particular case in which a no-hidden-layer perceptron is used for classification, i.e., when the inputs and outputs take on discrete values. If one has a mother function h , d inputs $x = (x_1, \dots, x_d)$, and m outputs $y = (y_1, \dots, y_m)$, then each output is given by

$$y_j = h(x^\top \alpha_j + \beta_j), \quad j = 1, \dots, m. \quad (4)$$

The limitations on the approximation capabilities of the no-hidden-layer perceptron derive from the fact that the functions (4) are constant along certain hyperplanes. For example, let us consider the case in which $d = 2$, $m = 1$, and h is an increasing function. Then,

$$y = h(x_1 \alpha_1 + x_2 \alpha_2 + \beta).$$

If four inputs x^1, x^2, x^3, x^4 are given, such that no three of them lie on a straight line, then there are output values that cannot be interpolated or approximated arbitrarily well. To see this, consider two inputs, x^1 and x^2 , that lie on the opposite sides of the line joining the inputs x^3 and x^4 , and let $y^1 = y^2 = 1$, $y^3 = y^4 = 0$. Then there exist no $(\alpha, \beta) \in \mathbb{R}^2 \times \mathbb{R}$ such that

$$y^i = h(x_1^i \alpha_1 + x_2^i \alpha_2 + \beta), \quad i = 1, \dots, 4.$$

Moreover, for any choice of α and β there must exist one y^i such that the difference in the desired output value and the associate one is at least $1/2$.

The behavior illustrated by the above example represents a major limitation on the no-hidden-layer model and prevents one from building a network able to classify points on the basis of different criteria or to approximate arbitrarily well “all reasonable” functions encountered in applications. In general, if the Heaviside computational unit is used in a no-hidden-layer architecture, two sets of points in \mathbb{R}^d can be separated (i.e., classified) if and only if they are linearly separable⁴.

The limitations on the no-hidden-layer perceptron motivated the study of more complex models, having at least one hidden layer. For example, in [31] it was proved that any p points in \mathbb{R}^d can be arbitrarily separated into two sets by a ridge OHL network with the Heaviside computational unit and

⁴Two sets of points in \mathbb{R}^d are *linearly separable* if there exists a hyperplane (called the *separating hyperplane*) such that all the points of one set are on one side of the hyperplane and all the points of the other set are on the other side.

one output if at least $\lceil p/d \rceil$ functions in the hidden layer are used (for every $x \in \mathbb{R}$, $\lceil x \rceil$ denotes the smallest integer larger than or equal to x).

In 1969, it was proved (see the well-known book [32] by Minsky and Papert) that the simple perceptron with no hidden layer can represent or approximate only functions belonging to quite a narrow class. However, this left open the possibility that network architectures containing one or more hidden layers might achieve better performances. Only some 20 years later did the first results in this direction appear: at the end of the 1980s, almost thirty years after the publication in 1960 of the two early rules for training adaptive elements in network architectures (the Perceptron Learning Rule by Rosenblatt and the Least Mean Square algorithm [33] by Widrow and Hoff), there began a certain “renaissance” of neural network theory. Many researchers started to investigate the density property (a review of the developments in feedforward neural networks in the 1960-1990 period is given in [30]) and, due to the above-discussed limits of the no-hidden-layer perceptron, investigations focused on networks with at least one hidden layer. The model with one hidden layer corresponds to OHL networks.

Starting from the late eighties, plenty of works appeared, proving that ridge OHL networks, under mild conditions on the computational units, are capable of approximating arbitrarily well wide classes of functions commonly used in applications, such as continuous and square-integrable ones. These works answered the following question, raised in [34]:

“The apparent ability of sufficiently elaborate feedforward networks to approximate quite well nearly any function encountered in applications leads one to wonder about the ultimate capabilities of such networks. Are the successes observed to date reflective of some deep and fundamental approximation capability, or are they merely flukes, resulting from selective reporting and a fortuitous choice of problems?”

4 Five phases

The investigation on density properties of ridge OHL networks can be divided into five main phases. As the literature on neural networks’ density is extremely wide, for each of such phases we shall shortly review some of the most meaningful papers.

4.1 A bunch of seminal papers

The interest in studying the density properties of ridge OHL networks was stimulated by the work [35], which appeared in 1987 and was based on the so-called “Kolmogorov Superposition Theorem” [36]. This theorem (later improved by [37] and [38]) answers the 13th Hilbert’s problem, showing that any continuous d -variable function can be represented by one-dimensional functions (for an outline of the interesting history of the Kolmogorov Superposition Theorem, see [39, pp. 168-169]). For the reader’s convenience, we report here a simplified form of the improved version provided in [39, p. 168].

Theorem 1 (Arnold-Kolmogorov-Lorentz-Sprecher). *There exist d constants $0 \leq \lambda_p \leq 1$, $p = 1, \dots, d$ and $2d + 1$ continuous functions $\varphi_q : [0, 1] \rightarrow \mathbb{R}$, $q = 1, \dots, 2d + 1$, with the following property. For every continuous function $f : [0, 1]^d \rightarrow \mathbb{R}$, one can find a continuous function g such that*

$$f(x_1, \dots, x_d) = \sum_{q=1}^{2d+1} g \left(\sum_{p=1}^d \lambda_p \varphi_q(x_p) \right). \quad (5)$$

In [35], Hecht-Nielsen called attention to the fact that this theorem can be read as an existence theorem for neural networks with two hidden layers, used to represent any continuous function. More precisely, he remarked that Theorem 1 has the following interpretation in terms of neural networks. For each $q = 1, \dots, 2d + 1$, let us take d neurons, each with an activation function φ_q . This represents the first hidden layer of the network. The weighted sums of the outputs $\varphi_q(x_p)$, $q = 1, \dots, 2d + 1$, $p = 1, \dots, d$ of these neurons are the inputs to the second hidden layer, having $2d + 1$ neurons, each of them with the activation function g . Finally, the output layer performs the sum of the outputs of the second hidden layer. Note that Theorem 1 guarantees the *exact representation of every* real-valued continuous function with d variables; moreover, the network is "almost universal," in the sense that everything but the activation functions in the second hidden layer is independent of the function to be represented.

After the seminal work [35], the importance of the Kolmogorov Superposition Theorem for the theory of representation and approximation of functions by neural networks was argued in both ways. For instance, in [40] it was remarked that the one-dimensional functions involved in Theorem 1 depend on the desired d -dimensional function to be represented and are far from those currently used in applications, where computational units are fixed in advance and have a simple form (e.g., they are sigmoidal functions). However, in [41] and [42], it was shown that, by replacing the exact representation guaranteed by the Kolmogorov Superposition Theorem with the search of an arbitrarily good approximation, the difficulties pointed out in [40] can be overcome, and the \mathcal{C} -density property can be proved by means on Theorem 1 for networks with two hidden layers. (Also an estimate of the number of hidden units was provided in [41] and [42].) For further work on the computational aspects of the Kolmogorov Superposition Theorem and its connections with neural networks, we refer the reader to [43] and [44]. An innovative neural-network architecture based on this theorem was later proposed in [45].

Going back to the seminal work [35], the author remarked that

"no constructive method for developing the" basis "functions is known" ... "the direct usefulness of this result is doubtful, at least in the nearest term"

Hence, the work [35] established an existence result that gave a hint into the capabilities of ridge OHL networks but that did not explain the successful performances of such networks in applications. A question naturally arose: is it possible to guarantee, if not an exact representation, at least an arbitrarily good approximation (in a suitable norm; e.g., the \mathcal{C} -density property) by using only computational units

with the same fixed and simple structure, such as those successfully used in applications? In subsequent years, this question received various answers, many of which will be reviewed in the following.

4.2 1988-1989: starting with sigmoidals

A first answer was given in 1988 by the author of [35] himself: in [46] he proved that ridge OHL networks with the so-called *logistic sigmoid* (i.e., $h(t) = \frac{1}{1+e^{-t}}$) enjoy for every compact set $K \subset \mathbb{R}^d$ the density property in $\mathcal{L}_2(K)$. In the same year, it was proved in [47] that the density property in $\mathcal{L}_2(K)$ is enjoyed also by ridge OHL networks having, as computational unit, the so-called *cosine squasher*, i.e., a nondecreasing sigmoidal function defined as

$$h(t) = \begin{cases} 0, & \text{if } t \leq -\pi/2, \\ (\cos(t + 3\pi/2) + 1) / 2, & \text{if } -\pi/2 \leq t \leq \pi/2, \\ 1, & \text{if } t \geq \pi/2. \end{cases}$$

However, the works [46] and [47] still deal with particular computational units, i.e., the logistic sigmoid and the cosine squasher. Hence, they leave open the question whether other computational units enjoy similar density properties.

Another paper that motivated the study of density properties for ridge OHL networks was [29], published in 1988. It showed the possibility of *exactly representing* every function in $\mathcal{L}_2(\mathbb{R}^d)$ by an integral representation that can be considered as a one-hidden-layer network with a “continuum” of certain hidden units belonging to $\mathcal{L}_1(\mathbb{R}^d)$ (note that sigmoidal functions do not satisfy this condition). Although this result is of almost no practical utility, as it involves a continuum of computational units, it is theoretically important and offered a basis for subsequent developments (e.g., [48]), which investigated approximations instead of exact representations. Note that most of the subsequent works focused on \mathcal{C} - and \mathcal{L}_p -density properties for functions defined on compact subsets of \mathbb{R}^d and not on all \mathbb{R}^d . The density property in the whole \mathbb{R}^d was considered again in [49, 50, 51, 52].

In 1988, the work [53] regarded as computational units the so-called *squashing functions*, i.e., non-decreasing sigmoidals [34, Definition 2.3] (a slightly different definition of squashing function is given in [47]). Examples of squashing functions are the Heaviside function, the *ramp function*, defined as $h(t) = 0$ if $t \leq 0$, $h(t) = t$ if $0 \leq t \leq 1$, and $h(t) = 1$ if $t \geq 1$, and the cosine squasher. It was proved in [53] that the \mathcal{C} -density property can be achieved by monotone squashing functions and two hidden layers.

The year 1989 is a milestone for the theory of ridge computational models and, more generally, for neurocomputing: for wide classes of sigmoidal functions the \mathcal{C} -density property on compacta was proved independently and almost simultaneously in the works [54, 48, 34], using different proof techniques. In the same year 1989, the \mathcal{L}_2 -density property on compacta was proved in [55] for continuous sigmoidals, using different tools. For an extension of the proof technique described in [48] and [34] to networks with the squashing function in the output layer, see [56].

The results in [48] and [54] regard continuous sigmoidals and continuous strictly increasing sigmoidals, respectively. In [57, p. 254], it was noted that the proof technique used in [54] can be easily generalized to the case where the computational unit is not necessarily a continuous sigmoidal but is continuous and has distinct finite limits at $\pm\infty$. The density result in [34] allows a noncontinuous sigmoidal, too: the only requirement is that the sigmoidal be nondecreasing (hence bounded). Moreover, it was noted in [57, p. 253] that the methods used in [34] can be easily modified to extend the \mathcal{C} -density property whenever the computational unit has distinct finite limits at $-\infty$ and $+\infty$.

4.3 Being sigmoidal is not substantial

After the density proofs for sigmoidals, obtained in [34], [48], [54], and [55], many papers improved and extended density results to wider classes of computational units. It is obvious that OHL networks with certain non-sigmoidal computational units enjoy suitable density properties: this is the case, for example, with the sine and cosine functions [47]. Let us consider some non-trivial extensions.

In Table 5.4 and the references cited therein, the reader can find other details on conditions under which the \mathcal{C} - and \mathcal{L}_p -density properties were proved in various papers (in reading the table, one should recall that conditions guaranteeing density in $\mathcal{C}(K)$ implies density in $\mathcal{L}_p(K)$).

The first result extending the density property to a class of non-necessarily sigmoidal computational units proved that every $h \in \mathcal{L}_1(\mathbb{R})$ such that $\int_{-\infty}^{+\infty} h(t) dt \neq 0$ has the \mathcal{C} -density property on compacta (note that sigmoidal functions do not belong to $\mathcal{L}_1(\mathbb{R})$). This justified theoretically the successful applications of ridge OHL networks with computational units different from “biologically motivated” sigmoidals [58].

Subsequently, various authors proved density results for non-necessarily sigmoidal computational units (e.g., the exponential $h(t) = e^t$ [59], piecewise linear functions, etc.). In [57], the \mathcal{C} -density property was proved for any continuous bounded and nonconstant computational unit, and the \mathcal{L}_p -density property was proved for any bounded and nonconstant computational unit. The \mathcal{C} -density property for functions on the whole \mathbb{R}^d was investigated in [50] and [51].

In [60], it was shown that, if $h : \mathbb{R} \rightarrow \mathbb{R}$ is a continuous function, bounded by a polynomial of degree k on all of \mathbb{R} , $\lim_{t \rightarrow -\infty} \frac{h(t)}{t^k} = 0$, $\lim_{t \rightarrow +\infty} \frac{h(t)}{t^k} = 1$ (such a function h is called *k-th degree sigmoidal*) and, for some constant $C > 0$ and all $t \in \mathbb{R}$, $|h(t)| \leq C(1 + |t|)^k$, then the \mathcal{C} -density properties holds if and only if h is not a polynomial. Thus, in this terminology, a zeroth degree sigmoidal function is what is usually called a sigmoidal function.

Other extensions were described in [52], [61], [62], [63], [64], [65], [66], [67], [68] and the references therein. In the papers [69] and [70], the investigations of the \mathcal{C} - and \mathcal{L}_p -density properties were given an exhaustive answer. In [70], it was proved that, for a wide class of non-necessarily continuous computational units, the necessary and sufficient condition for the \mathcal{C} - and \mathcal{L}_p -density properties is non-polynomiality. Similar conclusions were drawn in [69].

Among related results, it is worth mentioning that every real-valued function defined on a finite subset of \mathbb{R}^d (in particular, every Boolean function $f : \{0,1\}^d \rightarrow \mathbb{R}$) can be *implemented* (so, there is no approximation error) by a ridge OHL network with certain sigmoidal computational units (see the references in [14, Section 5]).

4.4 Approximating also derivatives

A successive step lay in proving that ridge OHL networks are able to approximate up to any degree of accuracy functions together with their derivatives. This can be modeled as a density property in suitable Sobolev spaces ⁵. We shall not discuss this density issue here, as our focus is on the density properties in $\mathcal{C}(K)$ and $\mathcal{L}_p(K)$ spaces, in relation with the problem of rates of approximation); the reader can look up in [14, Section 4] and the references therein.

4.5 Restricting the parameter set

A successive step was to prove density under restriction on the parameter set $\mathcal{A} \subset \mathbb{R}^k$ (see (1)). Various authors showed that, for a certain class of computational units, there exists a constant (dependent on certain characteristics of the computational units) such that the \mathcal{C} - density property holds also if the values of the weights and thresholds are bounded from above by a value not smaller than such a constant. In [69], the results were extended to an arbitrarily small upper bound on the values of the weights and thresholds.

The effects of constraining the sizes of parameters were further investigated in [71]. In [72], it was noted that inspection of the proofs reveals that some density results in [70] hold also when parameters are bounded by an arbitrarily small upper bound.

In the papers [51] and [61], various density properties were proved for monotone sigmoidal functions, using only weights with a norm equal to 1. The case of continuous sigmoidal computational units and of weights and thresholds taking only integer values was addressed in [73].

In [69], conditions on the computational units were given such that a single threshold in the hidden layer suffices for density (in writing “a single threshold,” we mean that all the thresholds of the computational units have the same value).

⁵If for the multi-integer $k = (k_1, \dots, k_d)$ one lets $|k| = \sum_{j=1}^d k_j$ and $D^k f(x) = \frac{\partial^{|k|}}{\partial x_1^{k_1} \dots \partial x_d^{k_d}} f(x)$, then, for an open set $\Omega \subseteq \mathbb{R}^d$, the Sobolev space $W_p^s(\Omega)$ consists of all the functions $f : \Omega \rightarrow \mathbb{R}$ such that there exists almost everywhere in Ω all partial derivatives $D^k f$ with $|k| \leq s$, and all of these derivatives are in $\mathcal{L}_p(\Omega)$, i.e., $W_p^s(\Omega) = \{f : \Omega \rightarrow \mathbb{R} \text{ such that } f^{(i)} \in \mathcal{L}_p(\Omega), i = 1, \dots, s\}$, where $f^{(i)}$ denotes the partial derivative of order i of f . The Sobolev norm of $f \in W_p^s(\Omega)$ is defined as $\|f\|_{W_p^s(\Omega)} = \sum_{|k| \leq s} \|D^k f\|_{\mathcal{L}_p(\Omega)}$. Thus, the necessity for approximating arbitrarily well not only a function but also certain derivatives corresponds to the density property in a suitable Sobolev space.

5 Proof techniques

Various methods have been used to prove the \mathcal{C} - and \mathcal{L}_p -density properties for ridge OHL networks. Although all methods yield similar or equivalent results, the proof techniques employed often have important consequences on computational aspects. For example, the proof can be merely existential or constructive; in the latter case, it gives useful information on the implementation of the network (e.g., by suggesting an algorithm to construct the network). Sometimes the proof can provide, as an important by-product, upper or lower bounds on the number n of computational units necessary to obtain the desired approximation accuracy, etc.

An intuitive way of proving the \mathcal{C} - and \mathcal{L}_2 -density properties of $\mathcal{P}^d(h)$ for suitable computational unit functions h is the following, qualitatively described in [74, p. 511]. First, one can decide to approximate any given function in \mathcal{C} or \mathcal{L}_2 by a classical multivariable finite trigonometric sum, with an arbitrarily small error (in the sup or \mathcal{L}_2 norm, respectively). Let us focus on the case $d = 2$. In such a case, the terms of the sum are of the form $a_{mn} \cos mx \cos ny$. By substituting into the sum the relationship $2 \cos mx \cos ny = \cos(mx + ny) + \cos(mx - ny)$, one obtains a linear combination of terms of the form $\cos(z_i)$, where each z_i is a linear function of x and y . Now, it is easy to show (see, e.g., [75]) that every function f of a single real variable can be approximated arbitrarily well by a ridge OHL network in the sup norm if f is continuous and in the \mathcal{L}_2 norm if it is square-integrable on a compact set. Hence, if one approximates the function $\cos(z_i)$ in such a way, also the original trigonometric sum is approximated by a ridge OHL network.

As regards the three works [34], [48], and [54], in which the \mathcal{C} -density property on compacta was first proved independently and almost simultaneously for wide classes of sigmoidal functions, it should be noted that three different proof techniques were used. Another technique was used in [55], where the first proof of the \mathcal{L}_2 -density property of ridge OHL networks was given.

In the remaining of this section, we shall first shortly review the techniques used in the four papers cited above, then we shall focus on one of them, namely, the proof technique used in [34]. Then we shall discuss in detail the quite general density results later obtained in [70] by means of an improved application of the same technique.

5.1 Proofs based on Fourier Analysis

The proof technique adopted in [48] is based on the integral representation developed in [29], combined with tools from Fourier Analysis. Such an integral representation is approximated by a finite sum that, in turn, can be expressed as a ridge OHL network with continuous strictly increasing sigmoidals. It is worth noting that, for the case of networks with two hidden layers, in [48] also an alternative proof of the \mathcal{C} -density property based on Kolmogorov Superposition Theorem (see Theorem 1) is given.

5.2 Proofs based on the Hahn-Banach Theorem

In [54], the \mathcal{C} -density property is proved via the Hahn-Banach Theorem [76, Section 4.8], which is a standard tool to conclude about the densities of sets of functions. Such a theorem implies that, if a linear subspace \mathcal{Y} of a normed linear space \mathcal{H} is not dense in \mathcal{H} , then there exists a nonzero continuous functional on \mathcal{H} that is equal to zero for all elements of \mathcal{Y} [76, p. 153].

As, for any continuous function h on K , $\mathcal{P}^d(h)$ is a linear subspace of $\mathcal{C}(K)$, to verify the density of $\mathcal{P}^d(h)$ by the Hahn-Banach Theorem it is sufficient to show that every linear functional vanishing on $\mathcal{P}^d(h)$ must be equal to zero on the whole of $\mathcal{C}(K)$. This is proved in [54] by exploiting the representation of continuous functionals on $\mathcal{C}(K)$ (see [76, Theorem 4.14.8]).

The same proof technique was used in [57] to prove the \mathcal{L}_p -density property. More precisely, as, for any bounded function h on K , $\mathcal{P}^d(h)$ is a linear subspace of $\mathcal{L}_p(K)$, to verify the density of $\mathcal{P}^d(h)$ it is sufficient to show that every linear functional vanishing on $\mathcal{P}^d(h)$ must be equal to zero on the whole of $\mathcal{L}_p(K)$. To prove this, [54] exploited the representation of continuous functionals on $\mathcal{L}_p(K)$ spaces [76, Theorem 4.14.1 and 4.14.6]. This elegant proof technique was later used in other papers (see Table 5.4).

5.3 Proofs based on the Radon Transform

The proof technique followed by [55] to prove the \mathcal{L}_2 -density property is based on the Radon transform. Without going into details (we refer the reader to [77] and [78]), here we just recall that the Radon transform and its inverse are basic theoretical tools in medical and geographical imaging and computerized tomography. Loosely speaking, they allow one to represent exactly a function by all its integrals over hyperplanes of \mathbb{R}^d . Each hyperplane is identified by its unit normal vector and its distance from the origin.

In [55], the integral formula obtained by the Radon transform and its inverse is approximated by a finite sum of terms, which, in turn, are approximated by a ridge OHL network with continuous sigmoids. Proof techniques based on the Radon transform were later exploited also in [50] and [52].

5.4 Proofs based on the Stone-Weierstrass Theorem

Another tool widely used in the study of density properties of ridge OHL networks, and first exploited for this purpose in [34], is the Stone-Weierstrass Theorem (e.g. [79, p. 190] and [12, pp. 146-153]), which is Stone's extension of the classical Weierstrass theorem on density of algebraic polynomials in $\mathcal{C}([a, b])$ (e.g. [79, p. 66] and [12, p. 146]). Such extension is obtained by "isolating" the properties of polynomials that make Weierstrass theorem possible and by generalizing it to a more general context in which these properties hold. To explain the Stone-Weierstrass Theorem, let us first recall some concepts.

According to the usual definition of multiplication between two functions, i.e., $(fg)(x) \triangleq f(x)g(x)$, the linear space $\mathcal{C}([a, b])$ of continuous functions over the interval $[a, b]$ becomes an algebra⁶. The poly-

⁶An algebra of real-valued functions is simply a linear space H of functions endowed with a multiplication that satisfies,

nomials defined over $[a, b]$ are a subset of $\mathcal{C}([a, b])$ and, as they are closed under multiplication (a product of polynomials is a polynomial), they are also an algebra; hence, they form a subalgebra of $\mathcal{C}([a, b])$. The Weierstrass Theorem can be rephrased by saying that *the subalgebra of real-valued algebraic polynomials defined over $[a, b]$ is dense in $\mathcal{C}([a, b])$ with the sup norm.*

Two more concepts have to be introduced at this point. A set A of functions defined on K *separates points* on K if, for any two distinct points $x, y \in K$, there exists a function $f \in A$ such that $f(x) \neq f(y)$. Clearly, the algebra of all polynomials with one real variable satisfies this condition in \mathbb{R} . On the other hand, the set of all even polynomials, say over the compact interval $[-1, 1]$, provides an example of an algebra that does not separate points, since $f(-x) = f(x)$ for every even function f . A set A of functions defined on K *vanishes at no point* of K if, for any $x \in K$, there exists a function $f \in A$ such that $f(x) \neq 0$. We are now ready to state Stone's generalization of the Weierstrass Theorem: it says that the same density conclusion with respect to the sup norm as in the Weierstrass theorem holds in any algebra $C(K)$ of continuous, real-valued functions on a compact set⁷ $K \subset \mathbb{R}^d$, for *every* algebra A that vanishes at no point of K and A separates points of K .⁸ For the reader's convenience, we report here the Stone-Weierstrass Theorem [12, p. 150 and p. 153].

Theorem 2 (Stone-Weierstrass). *Let $K \subset \mathbb{R}^d$ be compact and A an algebra of real-valued continuous functions defined on K (i.e., let A be a subalgebra of $C(K)$). If A separates points of K and vanishes at no point of K , then it is dense in $C(K)$.*

Note that the theorem does not extend to complex-valued functions (see [76, p. 118] and [12, pp. 152-153] for additional conditions guaranteeing the \mathcal{C} -density property in the case of algebras of complex-valued functions). Of course, the Weierstrass Theorem on the density of algebraic polynomials in $C(K)$, for every compact set $K \subset \mathbb{R}^d$, immediately follows from Theorem 2.

As a first application of the Stone-Weierstrass Theorem to the study of the density properties of ridge OHL networks, we give the following theorem (see [70, Proof of Theorem 1, Step 2]), which can be considered a "dimension-reduction" result for ridge OHL networks.

Theorem 3 ("Dimension-reduction" for ridge OHL networks). *Let $h : \mathbb{R} \rightarrow \mathbb{R}$. If $\mathcal{P}^1(h)$ is dense in $\mathcal{C}(I)$ for some nonempty compact interval $I \subset \mathbb{R}$, then, for every positive integer d and every compact set $K \subset \mathbb{R}^d$, $\mathcal{P}^d(h)$ is dense in $C(K)$.*

According to Theorem 3, one can restrict the study of the \mathcal{C} -density property for ridge OHL networks to networks with a single input, i.e., to the case in which functions to be approximated depend only

for all $f, g, h \in H$ and all $\alpha \in \mathbb{R}$, the following properties: i) $f(g+h) = fg + fh$, ii) $(f+g)h = fh + gh$, iii) $f(gh) = (fg)h$, iv) $\alpha(fg) = (\alpha f)g = f(\alpha g)$, v) H is closed under such a multiplication. Thus, an algebra of functions is closed under addition, multiplication and scalar multiplication. Any subset of an algebra is called a *subalgebra* if itself is an algebra.

⁷Note that a closed interval in \mathbb{R} , as the one to which Weierstrass theorem makes reference, is of course a compact set: according to the Heine-Borel theorem, every bounded and closed subset of \mathbb{R}^d is compact (see, e.g., [76, p. 92]).

⁸Sometimes, instead of requiring that A should vanish at no point of K , the more restrictive condition that the constant function 1 should belong to A is used in the statement of the theorem; e.g. [76, Theorem 3.7.1]).

on one variable. A similar dimension-reduction theorem was proved in [80]. Moreover, using a proof technique based on the Radon transform, a result was proved in [52, Theorems 2 and 3], regarding the reductions from $\mathcal{L}_p(K)$ to $\mathcal{L}_p(I)$ and from $\mathcal{C}(\bar{\mathbb{R}}^d)$ to $\mathcal{C}(\bar{\mathbb{R}})$, where, for all $d \geq 1$, $\mathcal{C}(\bar{\mathbb{R}}^d) \triangleq \{f \in \mathcal{C}(\mathbb{R}^d) : \lim_{\|x\| \rightarrow \infty} f(x) \text{ exists}\}$. An analogous dimension-reduction theorem was also proved in [65] and [61].

To describe the idea behind the proof of Theorem 3, let us investigate the applicability of the Stone-Weierstrass Theorem to the sets $\mathcal{P}^d(h)$ of ridge OHL networks defined on the compact set $K \subset \mathbb{R}^d$ with a ridge computational unit h . Such sets are closed with respect to addition and, for suitable functions h , they vanish at no point of K and separate points on K but generally they are not closed up to multiplication (for example, they are not closed if h is the widespread logistic sigmoidal function or the hyperbolic tangent). However, for every d , the set $\mathcal{P}^d(\exp)$ of d -variable functions on K computable by ridge OHL networks with the exponential as a computational unit is an algebra that separate points of K and contains the constant functions, so it vanishes at no point of K . Hence, for all d , the set of functions $\mathcal{P}^d(\exp)$ satisfies the assumptions of the Stone-Weierstrass Theorem, so it is dense in $\mathcal{C}(K)$. On the other hand, the density of $\mathcal{P}^1(h)$ in $\mathcal{C}(I)$, where I is a nonempty compact interval of \mathbb{R} , allows one to approximate up to any desired accuracy all continuous functions on I , in particular, the function $\exp(\cdot)$. Thus one can compose the two approximations, first approximating $f \in \mathcal{C}(K)$ by an element of $\mathcal{P}^d(\exp)$ and then approximating by an element of $\mathcal{P}^1(h)$ the one-variable function $\exp[g((x))]$, where $g(x) = x^\top \alpha + \beta$ (once a suitable re-scaling of the variables has been made, so that the range of such a composed function is contained in the interval I). Thus, the density of $\mathcal{P}^d(h)$ in $\mathcal{C}(K)$ is implied by the density of $\mathcal{P}^1(h)$ in $\mathcal{C}(I)$.

In [59], a variety of computational units were considered, that satisfy the hypotheses of the Stone-Weierstrass Theorem. So, the latter can be applied directly to OHL networks with such computational units to prove the \mathcal{C} -density property without the two-step approximation argument mentioned above. Possible advantages of using computational units that satisfy the hypotheses of the Stone-Weierstrass Theorem are discussed in [81, p. 30] and [59].

The Stone-Weierstrass Theorem is a widespread tool for verifying the density of sets of continuous functions. After the work [34], in which it has been used to prove the \mathcal{C} -density property of ridge OHL networks with sigmoidal activation functions in $\mathcal{C}(K)$, proof techniques based on the Stone-Weierstrass Theorem have been applied to prove the same property with more general computational units [70]. Also [54], which, as discussed above, exploits an argument based on the Hahn-Banach Theorem to prove the \mathcal{C} -density property for sigmoidal computational units, applies the Stone-Weierstrass Theorem to prove the same property for certain nonsigmoidal activation functions. For the relationship between the Stone-Weierstrass Theorem and neural networks, see also [81] and [59].

As a final remark on the different density-proof techniques mentioned so far, it is worth noting that some of them are simply existential (e.g., those based on the Hahn-Banach Theorem and the Stone-

Weierstrass Theorem), whereas others (e.g., those based on the Radon transform) are constructive, i.e., they provide a (typically not easy) way to construct ridge OHL networks with the desired density property and, sometimes, they give upper bounds on the number n of computational units of a certain type that guarantees a given degree of approximation accuracy (see also Table 5.4).

Table 5.4, which is by no means exhaustive, summarizes a variety of density results, listed in chronological order (in the case of papers that appeared in the same year, the order is alphabetical starting from the first author's name) for many types of computational units.

5.5 A case study

In the following, we present and discuss a slightly simplified form of the \mathcal{C} - and \mathcal{L}_p -density theorems in [70], which improve and generalize the results of [60]. This choice is motivated by the fact that they guarantee density in $\mathcal{C}(K)$ under quite mild hypotheses on the computational units and their proof can be sketched in a way that gives useful information. The next result is a slight simplification of [70, Theorem 1].

Theorem 4. *Let $h : \mathbb{R} \rightarrow \mathbb{R}$ be locally essentially bounded⁹ and piecewise continuous and d be a positive integer. Then for every compact set $K \subset \mathbb{R}^d$, $\mathcal{P}^d(h)$ is dense in $\mathcal{C}(K)$ if and only if h is not a polynomial.*

The necessity for nonpolynomiality in Theorem 4 can be easily seen. Indeed, if h is a polynomial of degree k , then $h(x^\top \alpha + \beta)$ is also a polynomial of degree k for every α and β . Hence, $\mathcal{P}^d(h)$ is the set of algebraic polynomials of degree at most k , thus it cannot be dense in $\mathcal{C}(K)$. The proof of the “if” part of Theorem 4 is the following.

Step 1. First one proves that, for ridge OHL networks, it is sufficient to prove the density of $\mathcal{P}^1(I)$ in $\mathcal{C}(I)$, where I is a compact interval of \mathbb{R} . This is done using Theorem 3.

Step 2. Then one considers the one-dimensional case and proves the density of $\mathcal{P}^1(h)$ in $\mathcal{C}(I)$ for every compact interval $I \subset \mathbb{R}$.

Step 2 is subdivided into four substeps, which we are am going to outline in the following. Let $\mathcal{C}^\infty(\mathbb{R})$ denote the space of functions $f : \mathbb{R} \rightarrow \mathbb{R}$ having derivatives of every order. Recall that the support of a continuous function $f : \mathbb{R}^d \rightarrow \mathbb{R}$, $d \geq 1$, is the complement of the largest open set in which f is equal

⁹Recall that a real-valued function f defined almost everywhere with respect to the Lebesgue measure μ on a measurable set $\Omega \subseteq \mathbb{R}^d$ $d \geq 1$ is said to be essentially bounded on Ω if $|f(x)|$ is bounded almost everywhere on Ω . We denote $f \in \mathcal{L}_\infty(\Omega)$, with the norm $\|f\|_{\mathcal{L}_\infty(\Omega)} = \inf\{c \mid \mu(\{x : |f(x)| \geq c\}) = 0\} = \text{ess sup}_{x \in \Omega} |f(x)|$. A locally essentially bounded function $f : \Omega \rightarrow \mathbb{R}$ is a function that is in $\mathcal{L}_\infty(K)$ for every compact subset K of Ω ; the notation $f \in \mathcal{L}_\infty^{\text{loc}}(\Omega)$ is used to denote such functions.

Space \mathcal{H}	Basis function $\varphi(\cdot)$	Proof technique (main tool)	Constructive approach	Bound on the number of units	Ref.
$\mathcal{L}_2(K)$	continuous sigmoidal	Radon Transform	Yes	$O\left(\left(\frac{1}{n}\right)^{\frac{1}{d-1}}\right)$	[55]
$\mathcal{C}(K)$	continuous sigmoidal	Hahn-Banach Th.	No	-	[54]
$\mathcal{L}_1(K)$	bounded sigmoidal	Hahn-Banach Th.	No	-	[54]
$\mathcal{L}_1(K)$	$\varphi \in \mathcal{L}_1(\mathbb{R}), \int \varphi(t) dt \neq 0$	Hahn-Banach Th.	No	-	[54]
$\mathcal{C}(K)$	strictly increasing continuous	Fourier Analysis	No	-	[48]
$\mathcal{C}(K)$	non-decreasing sigmoidal	Stone-Weierstr. Th.	No	-	[34]
$\mathcal{L}_p(K)$	non-decreasing sigmoidal	Stone-Weierstr. Th.	No	-	[34]
$\mathcal{C}(K)$	$\varphi \in \mathcal{L}_1(\mathbb{R}), \int \varphi(t) dt \neq 0,$ and continuous φ	Stone-Weierstr. Th.	No	-	[82]
$\mathcal{C}(K)$	bounded sigmoidal	staircase functions	Yes	-	[80]
$\mathcal{C}(K)$	continuous, bounded, and nonconstant	Hahn-Banach Th.	No	-	[57]
$\mathcal{L}_p(K)$	bounded and nonconstant	Hahn-Banach Th.	No	-	[57]
$\mathcal{C}(K)$	increasing sigmoidal	Hahn-Banach Th.	No	-	[61]
$\mathcal{C}(K)$	increasing sigmoidal	polyn. approx.	Yes	-	[61]
$\mathcal{C}(K)$	continuous and φ/P bounded for a polyn. P but φ φ not a polyn. itself	Hahn-Banach Th. and Fourier Analysis	No	-	[60]
$\mathcal{C}(K)$	k -th degree sigmoidal	spline approximation	Yes	$O\left(\left(\frac{1}{n}\right)^{d+1+\frac{d}{k+1}}\right)$	[60]
$\mathcal{C}(K)$	locally bounded, piecewise continuous, and not an algebraic polyn.	Stone-Weierstr. Th.	No	-	[70]
$\mathcal{C}(K)$	locally Riemann integrable and not an algebraic polyn.	Hahn-Banach Th.	No	-	[69]
$\mathcal{L}_p(K)$	locally bounded and not an algebraic polyn.	Hahn-Banach Th.	No	-	[69]
$\mathcal{C}(K)$	bounded sigmoidal	Radon Transform	Yes	$O\left(\left(\frac{1}{n}\right)^{\frac{1}{d-1}}\right)$	[52, 65]
$\mathcal{L}_p(K)$	sigmoidal and $\varphi \in L_p^{loc}(\mathbb{R})$	Fourier Analysis	Yes	-	[65]
$\mathcal{C}(K)$	bounded s.t. $\exists \lim_{t \rightarrow \pm\infty} \varphi(t)$	Radon transform	Yes	$O\left(\left(\frac{1}{n}\right)^{\frac{1}{d-1}}\right)$	[70]

Table 1: Density properties in $\mathcal{C}(K)$ and $\mathcal{L}_p(K)$.

to zero. In other words, the support of f is the closure¹⁰ in \mathbb{R}^d of the set $\{x \in \mathbb{R}^d : f(x) \neq 0\}$. In the following, $\mathcal{C}_c(\mathbb{R})$ denotes the space of real-valued continuous functions with compact support (i.e., they are equal to zero outside a compact set of \mathbb{R}) and $\mathcal{C}_c^\infty(\mathbb{R}) \triangleq \mathcal{C}^\infty(\mathbb{R}) \cap \mathcal{C}_c(\mathbb{R})$, i.e., it denotes the space of real-valued continuous functions with derivatives of every order and compact support.

Step 2.1. If h is not a polynomial and $h \in \mathcal{C}^\infty(\mathbb{R})$, then $\mathcal{P}^1(h)$ is dense in $\mathcal{C}(I)$ for every compact interval $I \subset \mathbb{R}$.

From now on, let the function h satisfy the hypotheses of Theorem 4, i.e., let it be locally bounded and piecewise continuous.

Step 2.2. For every $\xi \in \mathcal{C}_c^\infty(\mathbb{R})$ and every compact interval $I \subset \mathbb{R}$, $h * \xi \in \text{cl}_{\mathcal{C}(I)} \mathcal{P}^1(h)$, where $(h * \xi)(x) \triangleq \int h(x-y)\xi(y)dy$ denotes the convolution between h and ξ ¹¹ and $\text{cl}_{\mathcal{C}(I)} \mathcal{P}^d(h)$ denotes the closure of $\mathcal{P}^d(h)$ with respect to the sup norm in the space $\mathcal{C}(I)$.¹²

Step 2.3. If there exists $\xi \in \mathcal{C}_c^\infty(\mathbb{R})$ such that $h * \xi$ is not a polynomial, then $\mathcal{P}^1(h)$ is dense in $\mathcal{C}(I)$ for every compact interval $I \subset \mathbb{R}$. This is shown as follows. By Step 2.2, for every compact interval $I \subset \mathbb{R}$, one has $(h * \xi)(\cdot) \in \text{cl}_{\mathcal{C}(I)} \mathcal{P}^1(h)$; hence, for every $\alpha, \beta \in \mathbb{R}$, $(h * \xi)(\cdot^\top \alpha + \beta) \in \text{cl}_{\mathcal{C}(I)} \mathcal{P}^d(h)$. On the other hand, $h * \xi \in \mathcal{C}^\infty(\mathbb{R})$.¹³ Thus, by Step 2.2, if $h * \xi$ is not a polynomial, then $\mathcal{P}^1(h * \xi)$ is dense in $\mathcal{C}(I)$ for every compact set $I \subset \mathbb{R}$. Finally, $(h * \xi)(\cdot^\top \alpha + \beta) \in \text{cl}_{\mathcal{C}(I)} \mathcal{P}^d(h)$ and $\text{cl}_{\mathcal{C}(I)} \mathcal{P}^1(h * \xi) = \mathcal{C}(I)$ for every compact set $I \subset \mathbb{R}$ imply $\text{cl}_{\mathcal{C}(I)} \mathcal{P}^1(h) = \mathcal{C}(I)$.

Step 2.4. If $h * \xi$ is a polynomial for all $\xi \in \mathcal{C}_c^\infty(\mathbb{R})$, then h itself is a polynomial.

Finally, the density of $\mathcal{P}^1(h)$ in $\mathcal{C}(I)$ is obtained by combining as follows the above-summarized substeps: by 2.4, if the function h is locally bounded, piecewise continuous and not a polynomial, then there must exist $\xi \in \mathcal{C}_c^\infty(\mathbb{R})$ such that the function $h * \xi$ is not a polynomial and so, by 2.3 (whose proof exploits 2.2 and 2.1), $\mathcal{P}^1(h)$ is dense in $\mathcal{C}(I)$.

A more detailed inspection of the arguments used in [70] reveals that Theorem 5 extends to the case where all network parameters are bounded by an arbitrarily small upper bound. This is important as, in practice, the values of the parameters are always bounded.

As regards the spaces $\mathcal{L}_p(K)$, $p \in [1, \infty)$, the following theorem is a slightly simplified form of [70,

¹⁰If $\mathcal{M} \subset \mathbb{R}^d$, then $\text{cl}_{\mathbb{R}^d} \mathcal{M} = \{x \in \mathbb{R}^d : (\forall \varepsilon > 0) (\exists y \in \mathcal{M}) (\|x - y\| < \varepsilon)\}$, where $\|\cdot\|$ denotes the Euclidean norm in \mathbb{R}^d .

¹¹Note that such a convolution is well-defined. In general, given $f \in \mathcal{L}_1(\mathbb{R}^d)$ and $g \in \mathcal{L}_p(\mathbb{R}^d)$, the function $f * g$ is integrable on \mathbb{R}^d ; moreover, $f * g \in \mathcal{L}_p(\mathbb{R}^d)$ and $\|f * g\|_p \leq \|f\|_1 \|g\|_p$ (e.g., [83, Section IV.4]).

¹²I.e., $h * \xi \in \text{cl}_{\mathcal{C}(I)} \mathcal{P}^1(h)$ means that, for every $\varepsilon > 0$, there exist $n \in \mathbb{N}_+$ and $f_n \in \mathcal{P}^1(h)$ such that $\sup_{x \in I} |h * \xi - f_n| < \varepsilon$.

¹³More generally, if $f \in \mathcal{C}_c^\infty(\mathbb{R}^d)$ and $g \in L_1^{\text{loc}}(\mathbb{R}^d)$, where $L_1^{\text{loc}}(\mathbb{R}^d)$ denotes the space of functions in $L_1^{\text{loc}}(K)$ for every compact set $K \subset \mathbb{R}^d$, then $f * g \in \mathcal{C}^\infty(\mathbb{R}^d)$ (e.g. [83, Section IV.4]).

Proposition 1].

Theorem 5. *Let $h : \mathbb{R} \rightarrow \mathbb{R}$ be locally absolutely bounded and piecewise continuous and d be any positive integer. Then, for every compact set $K \subset \mathbb{R}^d$, $\mathcal{P}^d(h)$ is dense in $\mathcal{L}_p(K)$, $p \in [1, \infty)$, if and only if h is not an algebraic polynomial.*

The guidelines of the proof of Theorem 5 are quite plain. The necessity for nonpolynomiality immediately follows: note that, if h is a polynomial of degree k , then $\mathcal{P}^d(h)$ is a subset of the set of polynomials of degree at most k , hence it cannot be dense in $\mathcal{L}_p(K)$. As regards the sufficiency of nonpolynomiality, from Theorem 4 $\mathcal{P}^d(h)$ is dense in $\mathcal{C}(K)$. So one concludes using standard results of functional analysis and the argument of Section 2.3.

Theorems 4 and 5 deserve some further remarks.

First, note that, according to them, the \mathcal{C} - and \mathcal{L}_p -density properties are not restricted to “biologically motivated” sigmoidals but, with the exception of polynomials, they are satisfied by any “reasonable” computational unit.

Second, the computational units need not be continuous or smooth to guarantee \mathcal{C} - and \mathcal{L}_p -density properties: the only requirement is nonpolynomiality. In such a way, the results from [70], together with those in [69], answer the basic question raised in [57, Discussion, p. 253]:

“Whether or not the continuity assumption can entirely be dropped is still an open (and quite challenging) problem.”

Third, it is worth investigating the role played by the thresholds, i.e., by the parameters β_i in the computational units $h(x^\top \alpha_i + \beta_i)$ (see (2)). Toward this end, following [70] let us consider the continuous, bounded, non-constant and non-polynomial computational unit $h(x) = \sin(x)$ and the compact interval $[-1, 1] \subset \mathbb{R}$. As the family $\{\sin(wx), w \in \mathbb{R}\}$ consists only of odd functions, functions like $\cos(x)$ cannot be approximated by using such a family. Hence, the set of functions $\{\sin(wx), w \in \mathbb{R}\}$ cannot be dense in $\mathcal{C}([-1, 1])$. However, density can be recovered by adding to such a set functions with a threshold element (corresponding, in this case, to a phase shift), e.g., $\sin(x + \pi/2) = \cos(x)$. On the other hand, there exist ridge computational units on which the threshold has no influence: for example, this is the case with the exponential function. As previously stated in this section, in [69] conditions on the computational units are given such that a single threshold suffices for \mathcal{C} - and \mathcal{L}_p -density properties (i.e., the density properties can be guaranteed also when all the thresholds of the computational units have the same value).

6 The price of universality

We have seen that quite mild assumptions on the computational units allow one to prove that ridge computational models enjoy the density properties in $\mathcal{C}(K)$ and $\mathcal{L}_p(K)$. Various techniques can be used

to prove the \mathcal{C} - and \mathcal{L}_p - density properties for ridge and radial networks: the Kolmogorov Theorem and its variations, the Hahn-Banach Theorem, the Stone-Weierstrass Theorem, Fourier Analysis, and the Radon Transform.

Some proof techniques are merely existential: neither provide an algorithm to construct a model nor estimate the number of computational units necessary to guarantee the desired approximation accuracy.

Other proofs, instead, are constructive; among these, some provide an upper bound on the number of computational units. However, typically, these upper bounds grow “very fast” with the number d of variables of the functions to be approximated. This is illustrated in the fifth column of Table 5.4: either no estimate is given of the rate of growth of the number n of computational units in consequence of the number d of variables or upper bounds on this rate are provided that exhibit an exponential growth with d . Hence, they suffer from the so-called “*curse of dimensionality*” [84]. Loosely speaking, to achieve density *in the whole spaces* $\mathcal{C}(K)$ and $\mathcal{L}_p(K)$, an exponential growth of the number of computational units may occur.

Using the number n of computational units as a measure of model complexity and extending to a general context the expression “universal approximator” employed in neural network-parlance, the above-described behavior can be summarized in the following qualitative but deep and intriguing sentence: “universality can be obtained for arbitrarily large dimensions at the price of an exponentially growing complexity.” So we can say, more shortly: “*the price of universality is complexity.*”

Of course, computational models based on ridge functions are not always the best choice; depending on the application at hand, other approximators (based, e.g., on radial functions or tensor-product construction) may be better suited (see the discussion in [5], particularly Section 4 therein). For example, when one has to guarantee some directional homogeneity property, radial computational units should be preferred, as ridge functions are constant along certain hyperplanes. However, when ridge computational models have to be used, the recipe to cope with the curse of dimensionality lies in giving up proving density in all the spaces $\mathcal{C}(K)$ and $\mathcal{L}_p(K)$. In other words, if one confines oneself to using suitable subsets of functions, then one can obtain arbitrarily accurate approximations by ridge computational models with a number of computational units that, for a fixed approximation accuracy, grows “slowly” with d (see, e.g., [11, 85, 86, 87, 88, 89, 90] and the references therein).

References

- [1] Haykin S. *Neural Networks: A Comprehensive Foundation*. Prentice Hall; 1998 (2nd Edition).
- [2] Sejnowski TJ, Rosenberg CR. Parallel networks that learn to pronounce English text. *Complex Systems* 1987;1:145–168.

- [3] Zoppoli R, Parisini T. Learning techniques and neural networks for the solution of N-stage nonlinear nonquadratic optimal control problems. In: Isidori A, Tarn TJ, editors. *Systems, Models and Feedback: Theory and Applications*. Birkhäuser, Boston; 1992. p. 193–210.
- [4] Alessandri A, Parisini T, Sanguineti M, Zoppoli R. Neural strategies for nonlinear optimal filtering. In: *Proc. IEEE Int. Conf. Syst. Eng. Kobe (Japan)*; 1992. p. 44–49.
- [5] Sjöberg J, Zhang Q, Ljung L, Benveniste A, Glorennec PY, Delyon B, et al. Nonlinear black-box modeling in system identification: A unified overview. *Automatica* 1995;31:1691–1724.
- [6] Burrell PR, Folarin BO. The impact of neural networks in finance. *Journal Neural Computing & Applications* 1997;6:193–200.
- [7] Zoppoli R, Sanguineti M, Parisini T. Approximating networks and extended Ritz method for the solution of functional optimization problems. *Journal of Optimization Theory and Applications* 2002;112:403–440.
- [8] Sharda R, Rampal R. Neural networks and management science/operations research: A bibliographic essay. In: *Encyclopedia of Library and Information Science*, vol. 61, supp. 24. Marcel Dekker, Inc.; 1998. p. 247–259.
- [9] Smith KA. Neural networks for combinatorial optimization: A review of more than a decade of research. *INFORMS Journal on Computing* 1999;11(1):15–34.
- [10] Kurková V, Sanguineti M. Error estimates for approximate optimization by the extended Ritz method. *SIAM Journal on Optimization* 2005;15:461–487.
- [11] Giulini S, Sanguineti M. Approximation schemes for functional optimization problems. *Journal of Optimization Theory and Applications*; forthcoming.
- [12] Rudin W. *Principles of Mathematical Analysis*. McGraw-Hill Book Company; 1964.
- [13] Adams RA. *Sobolev Spaces*. Academic Press, New York; 1975.
- [14] Pinkus A. Approximation theory of the mlp model in neural networks. *Acta Numerica* 1999;8:143–195.
- [15] Pinkus A. Approximation by ridge functions. In: Méhauté ALe, Rabut C, Schumaker LL, editors. *Surface Fitting and Multiresolution Methods*. Vanderbilt University Press, Nashville, TN; 1997. p. 1–14.
- [16] Logan BF, Shepp LA. Optimal reconstruction of a function from its projections. *Duke Math Journal* 1975;42:645–659.

- [17] John F. Plane Waves and Spherical Means Applied to Partial Differential Equations. Interscience Publishers, Inc., New York; 1955.
- [18] Courant R, Hilbert D. Methods of Mathematical Physics. vol. II. Interscience Publishers, Inc., New York; 1962.
- [19] Donoho DL, Johnstone IM. Projection-based approximation and a duality method with kernel methods. *Annals of Statistics* 1989;17:58–106.
- [20] Friedman JH, Stuetzle W. Projection pursuit regression. *Journal of the American Statistics Association* 1981;76:817–823.
- [21] Lin VY, Pinkus A. Fundamentality of ridge functions. *Journal of Approximation Theory* 1993; 75:295–311.
- [22] Petrushev PP. Approximation by ridge functions and neural networks. *SIAM Journal on Mathematical Analysis* 1999;30:155–189.
- [23] Maiorov VE. On best approximation by ridge functions. *Journal of Approximation Theory* 1999; 99:68–94.
- [24] Gordon Y, Maiorov V, Meyer M, Reisner S. On the best approximation by ridge functions in the uniform norm. *Constructive Approximation* 2001;18:61–85.
- [25] Candes EJ. Ridgelets: Estimating with ridge functions. *Ann Statist* 2003;31:1561–1599.
- [26] Ismailov VugarE. A note on the best l_2 approximation by ridge functions. *Applied Mathematics E-Notes* 2007;7:71–76.
- [27] Rosenblatt F. The perceptron: A probabilistic model for information storage and organization of the brain. *Psychological Review* 1958;65:386–408.
- [28] Scarselli F, Tsoi AC. Universal approximation using feedforward neural networks: A survey of some existing methods and some new results. *Neural Networks* 1998;11:15–37.
- [29] Irie B, Miyake S. Capability of three-layered perceptrons. In: *Proc. International Joint Conference on Neural Networks*. New York; 1988. p. 641–648.
- [30] Widrow B, Lehr MA. 30 years of adaptive neural networks: Perceptron, Madaline, and Backpropagation. *Proceedings of the IEEE* 1990;78:1415–1442.
- [31] Baum EB. On the capabilities of multilayer perceptrons. *Journal of Complexity* 1988;4:193–215.
- [32] Minsky M, Papert S. *Perceptrons*. MIT press; 1969.

- [33] Widrow B, Hoff MEJr. Adaptive switching circuits. In: 1960 IRE Western Electric Show and Convention Record; 1960. p. 96–104.
- [34] Hornik K, Stinchcombe M, White H. Multilayer feedforward networks are universal approximators. *Neural Networks* 1989;2:359–366.
- [35] Hecht-Nielsen R. Kolmogorov's mapping neural network existence theorem. In: Proc. International Joint Conference on Neural Networks. San Diego, CA; 1987. p. 11–14.
- [36] Kolmogorov AN. On the representation of continuous functions of many variables by superposition of continuous functions of one variable and addition. *Doklady Akademii Nauk SSSR* 1957;114:953–956.
- [37] Sprecher DA. On the structure of continuous functions of several variables. *Transactions of the American Mathematical Society* 1965;115:340–355.
- [38] Lorentz GG. The thirteen problem of Hilbert. In: Proc. of Symposia in Pure Mathematics. American Mathematical Society, Providence, RI; 1976. .
- [39] Lorentz GG. *Approximation of Functions*. Chelsea Publishing Company, New York; 1966.
- [40] Girosi F, Poggio T. Representation properties of networks: Kolmogorov's theorem is irrelevant. *Neural Computation* 1989;1:465–469.
- [41] Kůrková V. Kolmogorov's theorem is relevant. *Neural Computation* 1991;3:617–622.
- [42] Kůrková V. Kolmogorov's theorem and multilayer neural networks. *Neural Networks* 1992;5:501–506.
- [43] Sprecher DA. A universal mapping for Kolmogorov's superposition theorem. *Neural Networks* 1993; 6:1089–1094.
- [44] Katsuura H, Sprecher DA. Computational aspects of Kolmogorov's superposition theorem. *Neural Networks* 1994;7:455–461.
- [45] Igel'nik Boris. Kolmogorov's spline network. *IEEE Transactions on Information Theory* 2003;14:725–733.
- [46] Hecht-Nielsen R. Theory of the backpropagation neural network. In: Proc. International Joint Conference on Neural Networks. San Diego, CA; 1989. p. 593–605.
- [47] Gallant AR, White H. There exists a neural network that does not make avoidable mistakes. In: Proc. International Joint Conference on Neural Networks. San Diego, CA; 1988. p. 657–664.
- [48] Funahashi K. On the approximate realization of continuous mappings by neural networks. *Neural Networks* 1989;2:183–192.

- [49] Chen T, Chen H, Liu R. A constructive proof of Cybenko's approximation theorem and its extensions. In: Proc. 22nd Symposium on Computing Science and Statistics. East Lansing, MI; 1990. p. 163–168.
- [50] Ito Y. Representation of functions by superpositions of a step or sigmoidal function and their applications to neural network theory. *Neural Networks* 1991;4:385–394.
- [51] Ito Y. Approximation of continuous functions on \mathbb{R}^d by linear combinations of shifted rotations of a sigmoidal function with and without scaling. *Neural Networks* 1992;5:105–115.
- [52] Chen T, Chen H, Liu R. Approximation capability in $C(\bar{\mathbb{R}}^n)$ by multilayer feedforward networks and related problems. *IEEE Transactions on Neural Networks* 1995;6:25–30.
- [53] Lapedes A, Farber R. How neural nets work. In: Anderson DZ, editor. *Neural Information Processing Systems*. American Institute of Physics, New York; 1988. p. 442–456.
- [54] Cybenko G. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals, and Systems* 1989;2:303–314.
- [55] Carroll SM, Dickinson BW. Construction of neural nets using the radon transform. In: Proc. International Joint Conference on Neural Networks. Washington, D.C.; 1989. p. 607–611.
- [56] Castro JL, Mantas CJ, Benítez JM. Neural networks with a continuous squashing function in the output are universal approximators. *Neural Networks* 2000;13:561–563.
- [57] Hornik K. Approximation capabilities of multilayer feedforward networks. *Neural Networks* 1991; 4:251–257.
- [58] Lee S, Kil RM. Multi-layer feedforward potential function network. In: Proc. International Joint Conference on Neural Networks. San Diego, CA; 1988. p. 161–171.
- [59] Cotter NC. The Stone-Weierstrass theorem and its applications to neural networks. *IEEE Transactions on Neural Networks* 1990;1:290–285.
- [60] Mhaskar HN, Micchelli CA. Approximation by superposition of a sigmoidal function and radial basis functions. *Advances in Applied Mathematics* 1992;13:350–373.
- [61] Ito Y. Approximation of functions on a compact set by finite sums of a sigmoid function without scaling. *Neural Networks* 1991;4:817–826.
- [62] Kreinovich VY. Arbitrary nonlinearity is sufficient to represent all functions by neural networks: A theorem. *Neural Networks* 1991;4:381–383.
- [63] Light WA. Ridge functions, sigmoidal functions, and neural networks. In: Cheney EW, Chui CK, Schumaker LL, editors. *Approximation Theory VII*. Academic Press, New York; 1993. p. 163–206.

- [64] Chen T, Chen H. Approximation of continuous functionals by neural networks with application to dynamic systems. *IEEE Transactions on Neural Networks* 1993;4:910–918.
- [65] Chen T, Chen H. Universal approximation to nonlinear operators by neural networks with arbitrary activation functions and application to dynamical systems. *IEEE Transactions on Neural Networks* 1995;6:911–917.
- [66] Attali JG, Pagès G. Approximations of functions by a multilayer perceptron: A new approach. *Neural Networks* 1997;10:1069–1081.
- [67] Burton RM, Dehling HG. Universal approximation in p -mean by neural networks. *Neural Networks* 1998;11:661–667.
- [68] Huang GB, Babri HA. Comments on “Approximation capability in $C(\bar{\mathbb{R}}^n)$ by multilayer feedforward networks and related problems. *IEEE Transactions on Neural Networks* 1998;9:714–715.
- [69] Hornik K. Some new results on neural network approximation. *Neural Networks* 1993;6:1069–1072.
- [70] Leshno M, Lin VY, Pinkus A, Schocken S. Multilayer feedforward networks with a nonpolynomial activation function can approximate any function. *Neural Networks* 1993;6:861–867.
- [71] Kůrková V. Approximation of functions by perceptron networks with bounded number of hidden units. *Neural Networks* 1995;8:745–750.
- [72] Kůrková V. Neural networks as nonlinear approximators. In: Bothe H, Rojas R, editors. *Proc. ICSC Symposia on Neural Computation (NC'2000)*. ICSC; 2000. p. 29–35.
- [73] Chui CK, Li X. Approximation by ridge functions and neural networks with one hidden layer. *Journal of Approximation Theory* 1992;70:131–141.
- [74] Blum EK, Li LK. Approximation theory and feedforward networks. *Neural Networks* 1991;4:511–515.
- [75] Blum EK. *Numerical Analysis and Computation: Theory and Practice*. Addison-Wesley, Reading, MA; 1972.
- [76] Friedman A. *Foundations of Modern Analysis*. Dover, New York; 1982. (Originally published by Holt, Rinehart, and Winston, New York, 1970).
- [77] Helgason S. *The Radon Transform*. Birkhauser; 1980.
- [78] Deans SR. *The Radon Transform and Some of Its Applications*. Wiley; 1983.
- [79] Cheney EW. *Introduction to Approximation Theory*. McGraw-Hill, Inc., USA; 1966.
- [80] Jones LK. Constructive approximation for neural networks by sigmoid functions. *Proceedings of the IEEE* 1990;78:1586–1589.

- [81] Kůrková V. Are sigmoidals the best activation functions in multilayer feedforward networks? *Neural Network World* 1992;1:27–34.
- [82] Stinchcombe M, White H. Universal approximation using feedforward networks with non-sigmoid hidden layer activation functions. In: *Proceedings of the International Joint Conference on Neural Networks*. vol. 1. Washington, D.C.: San Diego: SOS Printing; 1989 (Reprinted in *Artificial Neural Networks: Approximation & Learning Theory*, H. White, Ed., Blackwell, 1992). p. 613–617.
- [83] Brezis H. *Analyse Fonctionnelle - Théorie et Applications*. Masson, Paris; 1983.
- [84] Bellman R. *Dynamic Programming*. Princeton University Press; 1957.
- [85] Barron AR. Universal approximation bounds for superpositions of a sigmoidal function. *IEEE Transactions on Information Theory* 1993;39:930–945.
- [86] Girosi F, Anzellotti G. Rates of convergence for Radial Basis Functions and neural networks. In: Mammone RJ, editor. *Artificial Neural Networks for Speech and Vision*. Chapman & Hall, London; 1993. p. 97–113.
- [87] Makovoz Y. Random approximants and neural networks. *Journal of Approximation Theory* 1996; 85:98–109.
- [88] Burger M, Neubauer A. Error bounds for approximation with neural networks. *Journal of Approximation Theory* 2001;112:235–250.
- [89] Kůrková V, Sanguineti M. Comparison of worst case errors in linear and neural network approximation. *IEEE Transactions on Information Theory* 2002;48:264–275.
- [90] Mhaskar HN. On the tractability of multivariate integration and approximation by neural networks. *Journal of Complexity* 2004;20:561–590.