Neural Networks 91 (2017) 34–41

Contents lists available at ScienceDirect

Neural Networks

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

Probabilistic lower bounds for approximation by shallow perceptron networks

Věra Kůrková [a,*], Marcello Sanguineti [b]

[a] Institute of Computer Science, Czech Academy of Sciences, Pod Vodárenskou věží, 2 - 18207 Prague, Czech Republic

[b] DIBRIS, University of Genova, Via Opera Pia, 13 - 16145 Genova, Italy

HIGHLIGHTS

- Lower bounds on errors in approximation by shallow signum perceptron networks.
- Probabilistic approach to derivation of lower bounds.
- Sets of input–output functions of shallow networks with growing numbers of perceptrons.

ARTICLE INFO

Article history:
Received 28 December 2016
Received in revised form 7 April 2017
Accepted 10 April 2017
Available online 19 April 2017

Keywords:
Shallow networks
Perceptrons
Model complexity
Lower bounds on approximation rates
Chernoff–Hoeffding bounds

ABSTRACT

Limitations of approximation capabilities of shallow perceptron networks are investigated. Lower bounds on approximation errors are derived for binary-valued functions on finite domains. It is proven that unless the number of network units is sufficiently large (larger than any polynomial of the logarithm of the size of the domain) a good approximation cannot be achieved for almost any uniformly randomly chosen function on a given domain. The results are obtained by combining probabilistic Chernoff–Hoeffding bounds with estimates of the sizes of sets of functions exactly computable by shallow networks with increasing numbers of units.

1. Introduction

One-hidden-layer networks have been the standard type of feedforward network architecture until the recent renewal of interest in multilayer networks. Training of networks with more than one hidden layers was inefficient until the advent of graphic processing units allowing considerable acceleration of learning algorithms. In particular, networks with several layers of convolutional and pooling units have become state-of-the-art in pattern recognition (see, e.g., the survey article by LeCunn, Bengio, & Hinton, 2015 and references therein). Networks with several hidden layers are now called *deep* (see, e.g., Bengio, 2009; Chui & Mhaskar, 2016; Hinton, Osindero, & Teh, 2006 and the references therein) to distinguish them from *shallow* nets, which have merely one hidden layer.

It is well-known that networks with one hidden layer of computational units of many common types posses the universal approximation property (see, e.g., Anastassiou, 2011; Costarelli, 2015; Costarelli & Vinti, 2016a, 2016b, 2016c; Gripenberg, 2003; Hahm & Hong, 2016; Pinkus, 1999; Sanguineti, 2008 and references therein). But such topological density results do not provide guidelines for network design as they assume potentially unlimited numbers of network units. For practical applications it is desirable that given tasks can be performed by sufficiently sparse networks. Thus it is important to choose network architectures and types of units which can compute the tasks with numbers of parameters which are not too large.

Bengio and LeCun (2007) conjectured that "most functions that can be represented compactly by deep architectures cannot be represented by a compact shallow architecture". Bianchini and Scarselli (2014) proposed a promising approach to investigation of complexity of shallow and deep networks based on topological characteristics of input–output functions. Mhaskar, Liao, and Poggio (2016a, 2016b) suggested that deep networks are particularly

[*] Corresponding author.
E-mail addresses: vera@cs.cas.cz (V. Kůrková), marcello.sanguineti@unige.it (M. Sanguineti).

http://dx.doi.org/10.1016/j.neunet.2017.04.003
0893-6080/© 2017 Elsevier Ltd. All rights reserved.

suitable for computation of compositional functions and compared VC-dimensions of shallow and deep networks.

However, Ba and Caruana (2014) showed in a recent empirical study that in some cases, shallow networks can learn functions previously learned by deep ones using the same numbers of parameters as the latter. Thus a theoretical understanding complementing empirical results is needed that clarifies which tasks can be computed by deep networks with smaller model complexity than by shallow ones.

An important step towards this goal is the exploration of functions which cannot be computed or approximated by sparse shallow networks. Whereas many upper bounds on approximation errors by shallow networks are known (see, e.g., the survey article by Kainen, Kůrková, and Sanguineti (2012) and references therein), fewer lower bounds are available. Generally, proofs of lower bounds are much more difficult than arguments deriving upper bounds.

Worst-case errors exhibiting the curse of dimensionality in approximation of functions from Sobolev spaces by shallow networks with perceptrons with standard logistic sigmoid and piecewise polynomial activation functions were derived by Maiorov and Meir (2000). A concrete example of functions which cannot be efficiently computed by sparse shallow networks was presented by Bengio, Delalleau, and Le Roux (2006). They proved that for classification of points in the d-dimensional Boolean cube according to their parities by shallow Gaussian networks with fixed centers (a model used in SVM) at least 2^{d-1} units is necessary.

In this paper, we investigate lower bounds on rates of approximation of functions with increasing numbers of perceptrons in shallow networks, focusing on functions on finite domains (such as pixels of photographs, scattered data or discretized high-dimensional cubes). The set of real-valued functions on a finite domain can be identified with a finite-dimensional Euclidean space, which is of high dimension when the domain has large cardinality. Thus, the geometric properties of high-dimensional spaces influence the characterization of functions which cannot be well approximated by “reasonably small” shallow networks.

We exploit the Chernoff–Hoeffding Bound, which is a version of the concentration of measure phenomenon (see, e.g., Ledoux, 2001) to show that when the set of input–output functions of a general computational model is “relatively small” with respect to the size of the domain (depends on the logarithm of its size polynomially), then almost all uniformly randomly-chosen functions have large approximation errors.

To apply this general result to perceptron networks we estimate how quickly numbers of binary-valued functions computable by shallow signum perceptrons networks grow with increasing numbers of units. Combining these estimates with probabilistic methods, we derive lower bounds on errors in approximation of both binary-valued and real-valued functions on finite domains and prove that, unless the number of signum perceptrons units is sufficiently large (larger than any polynomial of the logarithm of the size of the domain), a good approximation cannot be achieved. For large domains, the lower bounds hold for almost any uniformly randomly-chosen function.

The paper is organized as follows. Section 2 contains definitions and notations. In Section 3, some general lower bounds holding for approximation of finite-domain functions are derived. Section 4 covers shallow networks of signum perceptrons, for which the universal representation property is established. In Section 5, there are derived estimates of the growth in the size of the set of binary-valued functions computable by shallow networks of signum perceptrons as the number of units increases. In Section 6, these estimates are combined with probabilistic methods to derive lower bounds on approximation errors of binary and real-valued functions on finite domains by shallow signum perceptron networks. Section 7 contains discussion and Section 8 a summary and some conclusions.

2. Preliminaries

A *one-hidden-layer* (*shallow*) *network with a single linear output* and n hidden units computes input–output functions from the set

$$\operatorname{span}_n G := \left\{ \sum_{i=1}^{n} w_i g_i \,\middle|\, w_i \in \mathbb{R},\ g_i \in G \right\},$$

where G, called *dictionary*, is a set of functions computable by a given type of hidden units. The *linear span* of G is defined as

$$\operatorname{span} G := \left\{ \sum_{i=1}^{n} w_i g_i \,\middle|\, w_i \in \mathbb{R},\ g_i \in G,\ \ n \in \mathbb{N} \right\}.$$

For binary classification tasks, *one-hidden-layer networks with a single threshold output unit* are used. Such networks compute functions from sets

$$\operatorname{sgn} \operatorname{span}_n G := \left\{ \operatorname{sgn} \sum_{i=1}^{n} w_i g_i \,\middle|\, w_i \in \mathbb{R},\ g_i \in G \right\},$$

where sgn denotes the *signum function* defined as

$\operatorname{sgn}(t) := -1$ for $t < 0$ and $\operatorname{sgn}(t) := 1$ for $t \geq 0$.

Dictionaries formed by computational units are parameterized families of functions of the form

$$G_\phi(X, Y) := \{\phi(\cdot, y) : X \to \mathbb{R} \mid y \in Y\},$$

where $\phi : X \times Y \to \mathbb{R}$ is a function of two variables: an input vector $x \in X \subseteq \mathbb{R}^d$ and a parameter vector $y \in Y \subseteq \mathbb{R}^s$. When the set of parameters is the whole $\mathbb{R}^s$, we write shortly $G_\phi(X)$.

For a domain $X \subset \mathbb{R}^d$, we denote by

$$\mathcal{F}(X) := \{f \mid f : X \to \mathbb{R}\}$$

the *set of all real-valued functions on* X and by

$$\mathcal{B}(X) := \{f \mid f : X \to \{-1, 1\}\}$$

the *set of all functions on* X *with values in* $\{-1, 1\}$.

Perceptrons compute functions of the form of plane waves $\sigma(v \cdot . + b) : X \to \mathbb{R}$, where $\sigma : \mathbb{R} \to \mathbb{R}$ is an *activation function*. We denote by ϑ the *Heaviside activation function*, defined as

$$\vartheta(t) := 0 \quad \text{for } t < 0 \quad \text{and} \quad \vartheta(t) := 1 \quad \text{for } t \geq 0$$

and by sgn the *signum activation function* $\operatorname{sgn} : \mathbb{R} \to \{-1, 1\}$, defined as

$$\operatorname{sgn}(t) := -1 \quad \text{for } t < 0 \quad \text{and} \quad \operatorname{sgn}(t) := 1 \quad \text{for } t \geq 0.$$

We denote by $H_d(X)$ the dictionary of functions on $X \subset \mathbb{R}^d$ computable by *Heaviside perceptrons* (which is equal to the set of characteristic functions of half-spaces of $\mathbb{R}^d$), i.e.,

$$H_d(X) := \{\vartheta(v \cdot . + b) : X \to \{0, 1\} \mid v \in \mathbb{R}^d, b \in \mathbb{R}\}$$

and by $P_d(X)$ the dictionary of functions on $X \subset \mathbb{R}^d$ computable by *signum perceptrons*, i.e.,

$$P_d(X) := \{\operatorname{sgn}(v \cdot . + b) : X \to \{-1, 1\} \mid v \in \mathbb{R}^d, b \in \mathbb{R}\}.$$

In theoretical analysis of approximation capabilities of shallow networks, it has to be taken into account that the approximation error $\|f - \operatorname{span}_n G\|$ in any norm $\|\cdot\|$ can be made arbitrarily large by multiplying f by a scalar. Indeed, for every $c > 0$ one has

$$\|cf - \operatorname{span}_n G\| = c\|f - \operatorname{span}_n G\|.$$

Thus approximation errors have to be studied either in sets of normalized functions or in sets of functions of a given fixed norm.

The dictionary $P_d(X)$ is a subset of $\mathcal{B}(X)$. So for X finite, all elements of $P_d(X)$ have the same norm, equal to $\sqrt{\operatorname{card} X}$. Thus in investigation of errors in approximation by shallow networks with threshold units it is more convenient to consider signum perceptrons than Heaviside ones. From the point of view of model complexity, there is only a minor difference between dictionaries of signum and Heaviside perceptrons, as

$$\mathrm{sgn}(t) = 2\vartheta(t) - 1 \quad \text{and} \quad \vartheta(t) = \frac{\mathrm{sgn}(t)+1}{2}.$$

So any network having n signum perceptrons can be replaced with a network having $n+1$ Heaviside perceptrons and vice-versa.

3. Lower bounds for approximation of binary-valued functions

In this section, we derive some general tools to investigate approximation of binary-valued functions on finite domains. The tools depend merely on the sizes of approximating sets and thus they can be applied to both shallow and deep networks for which sizes of sets of input–output functions can be estimated.

Our first proposition shows that when functions to be approximated are binary-valued, then lower bounds by networks with single linear outputs can be obtained from lower bounds holding for networks with single threshold outputs.

Proposition 3.1. *Let $X \subset \mathbb{R}^d$ be finite, $f \in \mathcal{B}(X)$, and $h \in \mathcal{F}(X)$. Then for $p \in \{1, 2\}$*

$$\|f - h\|_p \geq \frac{1}{2}\, \|f - sgn(h)\|_p.$$

Proof. We show that for every $x \in X$ one has $|f(x) - h(x)| \geq (1/2)\, |f(x) - \mathrm{sgn}(h(x))|$. Indeed if $f(x) = \mathrm{sgn}(h(x))$, then the right-hand side equals zero. If $f(x) \neq \mathrm{sgn}(h(x))$, then assuming without loss of generality that $f(x) = 1$, we have $h(x) < 0$ and thus $|f(x) - h(x)| > 1$. As in this case $(1/2)\, |f(x) - \mathrm{sgn}(h(x))| = 1$, the estimate holds. Thus for $X = \{x_1, \ldots, x_m\}$, we have

$$\|f - h\|_1 = \sum_{i=1}^{m} |f(x_i) - h(x_i)| \geq (1/2) \sum_{i=1}^{m} |f(x_i) - \mathrm{sgn}(h)(x_i)| = \frac{1}{2} \|f - \mathrm{sgn}(h)\|_1$$

and

$$\|f - h\|_2^2 = \sum_{i=1}^{m} |f(x_i) - h(x_i)|^2 \geq (1/4) \sum_{i=1}^{m} |f(x_i) - \mathrm{sgn}(h)(x_i)|^2 = \frac{1}{4} \|f - \mathrm{sgn}(h)\|_2^2. \quad \square$$

Sets of input–output functions of networks with a threshold output unit are binary-valued, hence for finite domains they are finite. For finite approximating sets on large domains, probabilistic estimates based on laws of large numbers can be applied to obtain lower bounds on approximation errors that hold for almost all functions. To derive such estimates we use the Chernoff–Hoeffding Bound. It provides an upper bound on the probability that the sum of bounded independent random variables deviates from its expected value (see, e.g., Petrov, 1995, p. 78, 2.6.2 and Hoeffding, 1963, Theorem 2).

Theorem 3.2 (*Chernoff–Hoeffding*). *Let m be a positive integer, $Y_1, \ldots, Y_m$ independent random variables with values in $[0, 1]$, $Y := \sum_{i=1}^{m} Y_i$, and $\lambda > 0$. Then the following hold:*

$$\Pr\left(Y \geq (1+\lambda)E(Y)\right) \leq e^{-\frac{\lambda^2}{2+\lambda}E(Y)};$$

$$\Pr\left(Y \leq (1-\lambda)E(Y)\right) \leq e^{-\frac{\lambda^2}{2}E(Y)}.$$

The following theorem based on the Chernoff–Hoeffding Bound shows that as long as an approximating set is “relatively small” with respect to the size of the domain, almost any randomly chosen binary-valued function cannot be well-approximated.

Theorem 3.3. *Let $X \subset \mathbb{R}^d$ be finite with $\mathrm{card}\, X = m$, $S \subset \mathcal{B}(X)$ such that $\mathrm{card}\, S = k$, and $\lambda > 0$. Then for every randomly uniformly chosen $f \in \mathcal{B}(X)$ and $p \in \{1, 2\}$,*

$$\Pr\Big(\|f - S\|_p \geq (1-\lambda)m\Big) \geq 1 - k\, e^{-\frac{m\lambda^2}{2}}.$$

Proof. By definition, $\|f - S\|_p = \min_{h \in S} \|f - h\|_p$. Fix some $h \in S$. Without loss of generality we can assume that for all $i = 1, \ldots, m$, $h(x_i) := 1$. Otherwise, we apply a sign-flipping operator $F_h : \mathcal{B}(X) \to \mathcal{B}(X)$ defined for every $i = 1, \ldots, m$ as $F_h(f)(x_i) := h(x_i) f(x_i)$. So $F_h(h)(x_i) = 1$ for every $i = 1, \ldots, m$. Moreover, for every $f \in \mathcal{B}(X)$ one has $\|f - h\|_1 = \|F_h(f) - F_h(h)\|_1$ and $\|f - h\|_2 = \|F_h(f) - F_h(h)\|_2$. Since F_h is a one-to-one mapping on $\mathcal{B}(X)$, the uniform distribution is invariant under F_h, and thus $\Pr(\|f - h\|_p \leq c) = \Pr(\|F_h(f) - F_h(h)\|_p \geq c)$ for $p = 1, 2$.

Let $p := 1$. For $i = 1, \ldots, m$, let $Y_i \in \{0, 1\}$ be the independent random variables defined as $Y_i := |f(x_i) - h(x_i)|$ and $Y := \sum_{i=1}^{m} Y_i = \|f - h\|_1$, respectively. Then $E(Y) = m$ and so by Theorem 3.2 we get

$$\Pr\Big(Y \leq (1-\lambda)m\Big) \leq e^{-\frac{m\lambda^2}{2}}.$$

So for all $h \in S$

$$\Pr\Big(\|f - h\|_1 \leq (1-\lambda)m\Big) \leq e^{-\frac{m\lambda^2}{2}}$$

and thus

$$\Pr\Big(\|f - S\|_1 \geq (1-\lambda)m\Big) \geq 1 - k\, e^{-\frac{m\lambda^2}{2}}.$$

Now, let $p := 2$. Similarly, setting $Z_i := (f(x_i) - h(x_i))^2$ for $i = 1, \ldots, m$, and $Z := \sqrt{\sum_{i=1}^{m} Z_i} = \|f - h\|_2$, we get $E(Z) = m$. Thus by Theorem 3.2,

$$\Pr\left(Z \leq (1-\lambda)m\right) \leq e^{-\frac{m\lambda^2}{2}}$$

and so

$$\Pr\Big(\|f - S\|_2 \geq (1-\lambda)m\Big) \geq 1 - k\, e^{-\frac{m\lambda^2}{2}}. \quad \square$$

In Theorem 3.3, the parameter λ determines the trade-off between the size of the lower bound and the probability that it holds. When applied to large domains X, for suitable values of λ the theorem implies large lower bounds. For instance, for $\lambda = 1/2$ a high probability that any uniformly randomly chosen function $f \in \mathcal{B}(X)$ cannot be approximated by functions from sets S within error smaller than $m/2$ holds when $\mathrm{card}\, S$ is not too large to outweigh $e^{-m/16}$ so that $1 - \mathrm{card}\, S\, e^{-m/16}$ remains close to 1. This occurs, for example, when $\mathrm{card}\, S$ depends on m polynomially. If, instead, $\mathrm{card}\, S = k = 2^t$ for some $t > 0$, then the lower bound

$$1 - k\, e^{-\frac{m\lambda^2}{4}} \geq 1 - 2^{t - \frac{m\lambda^2}{4}}.$$

Thus for S such $k = \mathrm{card}\, S \ll 2^t$, with $t \ll \frac{m\lambda^2}{4}$ we have $2^{t-\frac{m\lambda^2}{4}}$ small. For instance, with $\lambda = \frac{1}{2}$, we get

$$\Pr\Big(\|f - S\|_1 \geq \frac{m}{2}\Big) \geq 1 - k\, e^{-\frac{m\lambda^2}{4}}.$$

This bound is close to 1 when $t \ll \frac{m}{16}$, i.e., when $\mathrm{card}\, S \ll 2^{\frac{m}{16}}$.

4. Universal representation by signum perceptron networks

To apply general tools derived in the previous section to signum perceptron networks, we need to estimate the sizes of the sets $\operatorname{sgn}\operatorname{span}_n P_d(X)$. In this section we give a simple geometric argument showing that for the extreme case $n = \operatorname{card} X$, all functions on X can be exactly computed by shallow signum perceptron networks and thus $\operatorname{card}\operatorname{sgn}\operatorname{span}_n P_d(X) = 2^{\operatorname{card} X}$.

We call the capability of a class of networks to exactly compute all functions on any finite domain the *universal representation property*. The following theorem, extending a result by Ito (1992), proves this property for networks with signum perceptrons.

Theorem 4.1. *Let d and m be positive integers and $X \subset \mathbb{R}^d$ such that $\operatorname{card} X = m$. Then every function $f : X \to \mathbb{R}$ belongs to the set $\operatorname{span}_m P_d(X)$.*

Proof. Let $X := \{x_1, \ldots, x_m\}$. Ito (1992) proved that a sufficient condition guaranteeing for a dictionary of the form $G_\varphi(X, Y)$ that every $f : X \to \mathbb{R}$ is in $\operatorname{span}_m G_\varphi(X)$ is the existence of $y_i, \ldots, y_m \in Y$ such that the matrix $M(\varphi)$ defined for every $i, j = 1, \ldots, m$ as $M(\varphi)_{i,j} := \varphi(x_i, y_j)$ is non singular.

We verify this condition for $\varphi : X \times \mathbb{R}^{d+1} \to \mathbb{R}$ defined as $\varphi(x, (v, b)) := \operatorname{sgn}(v \cdot + b)$. Choose $x_0 \notin \operatorname{conv}\{x_1, \ldots, x_m\}$ such that $\|x_i - x_0\| \neq \|x_j - x_0\|$ for $i \neq j$. Without loss of generality we can assume that $\|x_1 - x_0\| \leq \|x_2 - x_0\| \leq \cdots \leq \|x_m - x_0\|$, (otherwise we re-order the set X).

Let v_i and b_i be the normal vector and the bias, resp., of the tangent hyperplane at the point x_i to the ball centered at x_0 with radius $\|x_i - x_0\|$. Denote by A the $m \times m$ matrix defined as

$$A_{ij} := \operatorname{sgn}(v_j \cdot x_i + b_j)$$

(see Fig. 1). It follows from the definition that $A_{ij} = -1$ for $i < j$ and $A_{ij} = 1$ for $i \geq j$. By adding to all columns of A its last column (whose entries are all equal to 1), A can be transformed into a triangular matrix B. So we have

$$A := \begin{pmatrix} 1 & 1 & \cdot & \cdot & 1 & 1 \\ -1 & 1 & \cdot & \cdot & 1 & 1 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ -1 & -1 & \cdot & \cdot & 1 & 1 \\ -1 & -1 & \cdot & \cdot & -1 & 1 \end{pmatrix}$$

$$B := \begin{pmatrix} 2 & 2 & \cdot & \cdot & 2 & 2 \\ 0 & 2 & \cdot & \cdot & 2 & 2 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & \cdot & \cdot & 2 & 2 \\ 0 & 0 & \cdot & \cdot & 0 & 2 \end{pmatrix}.$$

Since the determinant of a matrix is invariant under addition of a column and a triangular matrix is non singular, A is non singular, too. □

Theorem 4.1 guarantees universal representation by assuming model complexity potentially equal to the size m of the domain. Such a universality result does not provide practical guidelines for applications to large domains. Thus sets of functions that can be computed or well approximated by networks with reasonably small numbers of hidden units should be investigated. In the next section, we explore limitations of approximation capabilities of signum perceptron networks with numbers of units considerably smaller than sizes of domains of functions to be approximated.

5. Estimates of sizes of sets of functions computable by signum perceptron networks

The sets of binary-valued functions on finite domains $\mathcal{B}(X)$ are finite, whereas the approximating sets formed by input–output functions of signum perceptrons $\operatorname{span}_n P_d(X)$ are infinite. By Proposition 3.1, lower bounds on the approximation error by $\operatorname{span}_n P_d(X)$ can be obtained from lower bounds on approximation by

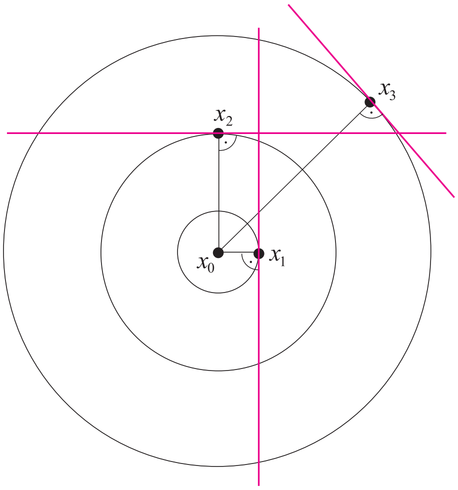

Fig. 1. The construction used in the proof of Theorem 4.1.

$$\operatorname{sgn}\operatorname{span}_n P_d(X) := \operatorname{sgn}\left\{ \sum_{i=1}^{n} w_i \operatorname{sgn}(v_i \cdot . + b_i) \,\middle|\, w_i, b_i \in \mathbb{R},\, v_i \in \mathbb{R}^d \right\},$$

which are finite subsets of $\mathcal{B}(X)$.

To apply Theorem 3.3 to approximation by these sets, we need to estimate their sizes. For positive integers d, m, and n we denote

$$s(m, d, n) := \max \left\{ \operatorname{card}\operatorname{sgn}\operatorname{span}_n P_d(X) \mid X \subset \mathbb{R}^d, \operatorname{card} X = m \right\}.$$

Two extreme cases are $n = 1$ and $n = m$. When $n = m$, the maximal size $s(m, d, m)$ of the set $\operatorname{sgn}\operatorname{span}_m P_d(X)$ is equal to 2^m. Indeed by Theorem 4.1, all real-valued functions on X can be computed by signum perceptron networks with m units. So in particular, all binary-valued functions can be exactly represented by these networks.

In the second extreme case when $n = 1$, the quantity $s(m, d, 1)$ is equal to the maximal size of the dictionary $P_d(X)$ on any set X of m points in $\mathbb{R}^d$. An upper bound on size of the set formed by functions computable by signum perceptrons on m point set in $\mathbb{R}^d$ is well-known since the work of Schläfli (1901), who estimated the number of linearly separated dichotomies of m points in $\mathbb{R}^d$. The upper bound, derived by induction on both d and m, states that for every $X \subset \mathbb{R}^d$ such that $\operatorname{card} X = m$ one has (see, e.g., Cover, 1965)

$$\operatorname{card} P_d(X) \leq 2 \sum_{l=1}^{d} \binom{m-1}{l} \leq 2\frac{m^d}{d!}. \qquad (1)$$

For $n = 1$, the set $\operatorname{sgn}\operatorname{span}_1 P_d(X) = P_d(X)$ and so

$$s(m, d, 1) \leq 2\frac{m^d}{d!}.$$

Thus the set $\operatorname{sgn}\operatorname{span}_1 P_d(X)$ forms only a small fraction of the set of all functions in the set $\mathcal{B}(X)$, whose cardinality is equal to 2^m. With n increasing from 1 to m it eventually reaches the size 2^m. Thus to understand limitations of approximation capabilities of signum perceptron networks, we need to estimate how quickly the numbers $s(m, d, n)$ grow with n increasing. The next theorem provides an upper bound on this grows.

Theorem 5.1. *For all positive integers d, m, n,*

$$s(m, d, n) \leq 2 \sum_{l=0}^{n(d+2)-1} \binom{nm-1}{l} \leq 2\,\frac{(nm)^{n(d+2)-1}}{(n(d+2)-1)!}.$$

Proof. Let $X = \{x_1, \dots, x_m\}$. To estimate $\operatorname{card} \operatorname{sgn} \operatorname{span}_n P_d(X)$ we denote for every $x \in X$ and $u = (u_1, \dots, u_n) = (w_1, v_1, b_1, \dots, w_n, v_n, b_n) \in \mathbb{R}^{n(d+2)}$ by $h(u, x)$ the value at x of the function from $\operatorname{sgn} \operatorname{span}_n P_d(X)$ with parameters $u = (u_1, \dots, u_n)$, i.e.,

$$\begin{aligned} h(u,x) &:= \operatorname{sgn}\left(\sum_{i=1}^{n} g(u_i, x)\right) \\ &= \operatorname{sgn}\left(\sum_{i=1}^{n} w_i \operatorname{sgn}(v_i \cdot x + b_i)\right) : \mathbb{R}^{n(d+2)} \to \{-1, 1\}. \quad (2) \end{aligned}$$

We call two parameter vectors $u := (w_1, v_1, b_1, \dots, w_n, v_n, b_n)$ and $\bar{u} := (\bar{w}_1, \bar{v}_1, \bar{b}_1, \dots, \bar{w}_n, \bar{v}_n, \bar{b}_n)$ from $\mathbb{R}^{n(d+2)}$ *equivalent* if on the finite domain X the functions $h(u, x)$ and $h(\bar{u}, x)$ coincide, i.e., for all $x \in X$, $h(u, x) = h(\bar{u}, x)$. Thus the number $s(m, d, n)$ of elements of the set $\operatorname{sgn} \operatorname{span}_n P_d(X)$ is smaller than or equal to the number of classes of this equivalence.

Sets of equivalent vectors in the parameter space $\mathbb{R}^{n(d+2)}$ are separated by hyperplanes. For each $x \in X$, the function (2) changes its sign in those points $u \in \mathbb{R}^{n(d+2)}$ for which for some $i = 1, \dots, m$ one has $w_i = 0$ or $v_i \cdot x + b_i = 0$. The set of these points has the form of the union of $n(1 + m)$ hyperplanes. To define them, let

$$\alpha_i := (\alpha_{i1}, \dots, \alpha_{in}),$$

where $\alpha_{ik} \in \mathbb{R} \times \mathbb{R}^d \times \mathbb{R}$ and $\alpha_{ik} = (0, 0, 0)$ if $i \neq k$ and $\alpha_{ii} = (1, 0, 0)$ and

$$\beta_{ij} := (\beta_{ij1}, \dots, \beta_{ijkn}),$$

where $\beta_{ijk} \in \mathbb{R} \times \mathbb{R}^d \times \mathbb{R}$ and $\beta_{ijk} = (0, x_j, 1)$. Then, the hyperplanes are defined by the $n(1+m)$ equations $u \cdot \alpha_i = 0$ and $u \cdot \beta_{ij} = 0$, $i = 1, \dots, n$ and $j = 1, \dots, m$.

Thus the number of equivalence classes is bounded from above by the number of regions into which $\mathbb{R}^{n(d+2)}$ can be divided by $n(m+1)$ hyperplanes. However, as $w \operatorname{sgn}(v \cdot x + b) = -w \operatorname{sgn}(-v \cdot x - b)$, the division of $\mathbb{R}^{n(d+2)}$ by hyperplanes described by the equations $u \cdot \alpha_i = 0$ can be excluded. So, the number of equivalence classes is bounded from above by the number of regions into which $\mathbb{R}^{n(d+2)}$ is divided by nm hyperplanes.

It is known (see, e.g., Anthony, 2001, p. 36) that the number of regions $c(D, N)$ into which $\mathbb{R}^D$ can be partitioned by N hyperplanes going through the origin is bounded from above by

$$c(D, N) \leq 2 \sum_{l=0}^{D-1} \binom{N-1}{l}. \tag{3}$$

Thus letting $D = n(d + 2)$ and $N = nm$, from (3) we get

$$s(m, d, n) \leq 2 \sum_{l=0}^{n(d+2)-1} \binom{nm-1}{l}.$$

The partial sum of binomials satisfies (see Rojas, 1996, p. 43 and Winder, 1962)

$$\sum_{l=0}^{M} \binom{N-1}{l} \leq \frac{N^M}{M!}.$$

So, for $M = n(d + 2) - 1$ and $N = nm$ we get the statement. □

Applying Theorem 5.1 to the extreme cases $n = 1$ and $n = m$, we get the following corollary.

Corollary 5.2. *For every $X \subset \mathbb{R}^d$ such that $\operatorname{card} X = m$, the following hold:*

$$\text{(i) } s(m, d, 1) \leq 2\, \frac{m^{d+1}}{(d+1)!}; \tag{4}$$

$$\text{(ii) } s(m, d, m) \leq 2\, \frac{m^{2(m(d+2)-1)}}{(m(d+2)-1)!}. \tag{5}$$

Proof. By Theorem 5.1, we have $s(m, d, 1) \leq 2 \sum_{l=0}^{d+1} \binom{m-1}{l} \leq 2 \frac{m^{d+1}}{(d+1)!}$ and $s(m, d, m) \leq 2 \sum_{l=0}^{m(d+2)-1} \binom{m^2-1}{l} \leq 2 \frac{m^{2(m(d+2)-1)}}{(m(d+2)-1)!}$. □

To compare the bounds from Corollary 5.2(i) with the classical estimate (1) by Schläfli, we note that

$$\frac{m^{d+1}}{(d+1)!} = \frac{m}{d+1} \, \frac{m^d}{d!}.$$

So:

- for $m = d + 1$, the bounds (1) and (4) are equal;
- for $m < d + 1$, the estimate (4) is better than the estimate (1);
- for growing $m > d + 1$, the upper bound (4) becomes increasingly worse than the bound (1).

Simplifying the upper bound from Theorem 5.1 we get the following corollary.

Corollary 5.3. *For all positive integers d, m, n and $\bar{d} = d + 2$,*

$$s(m, d, n) \leq n^{\frac{n\bar{d}}{2}}\, m^{n\bar{d}}.$$

Proof. The estimate from Theorem 5.1 combined with the inequalities $n \leq n\bar{d} - 1$ and $n^{\frac{n}{2}} \leq n!$ (indeed, one has $n^{\frac{n}{2}} \leq n! \leq \left(\frac{n+1}{2}\right)^n$) gives

$$\begin{aligned} s(m, d, n) &\leq 2\, \frac{(nm)^{n\bar{d}-1}}{(n\bar{d}-1)!} \leq 2\, {\frac{(nm)^{n\bar{d}-1}}{(n\bar{d}-1)}}^{\frac{n\bar{d}-1}{2}} \\ &\leq 2\, \frac{(nm)^{n\bar{d}-1}}{n^{\frac{n\bar{d}-1}{2}}} = 2\, n^{\frac{n\bar{d}-1}{2}}\, m^{n\bar{d}-1} \\ &\leq n^{\frac{n\bar{d}}{2}}\, m^{n\bar{d}}. \quad \square \end{aligned}$$

By Theorem 4.1, for all m, d, n one has $s(m, d, n) \leq 2^m$. So the upper bound from Theorem 5.1 is useful only when it is smaller than 2^m. By Corollary 5.3, this holds for n satisfying $2^{\frac{n\bar{d}}{2} \log n}\, 2^{n\bar{d} \log m} \leq 2^m$, i.e., for such n that

$$\frac{n}{2} \log_2 n + n \log_2 m \leq \frac{m}{\bar{d}}. \tag{6}$$

As we assume that $n \leq m$, the condition (6) holds for

$$n \leq \frac{2}{3}\, \frac{m}{\bar{d} \log_2 m}. \tag{7}$$

Hence, our upper estimates of the sizes of sets of functions $\operatorname{sgn} \operatorname{span}_n P_d(X)$ are useful for networks with numbers of units n smaller than $\frac{2}{3} \frac{m}{\bar{d} \log_2 m}$.

Applying these estimates to 2-dimensional domains containing $m = 2^s$ points, we get from (7) the condition $n \leq \frac{2^{s+1}}{12\, s}$. For domains in the form of the d-dimensional Boolean cubes, $m = 2^d$ and (7) implies that the estimates are useful only when $n \leq \frac{2^{d+1}}{3d\,(d+2)}$.

6. Estimates of approximation errors by shallow signum perceptron networks

In this section, we combine general tools for derivation of "most-cases" lower bounds on errors in approximation of binary-valued functions on finite domains with estimates of sizes of sets of functions computable by shallow signum perceptron networks with threshold units which are summarized in Table 1.

By combining Theorem 3.3 with the upper bound on $s(m, d, n)$ from Theorem 5.1, we get the following estimate.

Table 1
Estimates of $s(m, d, n) := \text{card sgn span}_n\, P_d(X)$, where $\bar{d} := d + 2$.

$1 \le n \le m$	$s(m, d, n) \le 2\, \frac{(nm)^{n\bar{d}-1}}{(n\bar{d}-1)!}$	Theorem 5.1
	$s(m, d, n) \le n^{\frac{n\bar{d}}{2}}\, m^{n\bar{d}}$	Corollary 5.3
$n = 1$	$s(m, d, 1) \le \frac{m^d}{d!}$	Eq. (1)
	$s(m, d, 1) \le 2\, \frac{m^{d+1}}{(d+1)!}$	Corollary 5.2(i)
$n = m$	$s(m, d, m) = 2^m$	Theorem 4.1
	$s(m, d, m) \le 2\, \frac{m^{2(m\bar{d}-1)}}{(m\bar{d}-1)!}$	Corollary 5.2 (ii)

Theorem 6.1. *Let $X \subset \mathbb{R}^d$ such that* $\text{card}\, X = m$, $\lambda > 0$, $p \in \{1, 2\}$, *and n be a positive integer, then*

$$\Pr\Big(\|f - S\|_p \ge (1-\lambda)m\Big) \ge 1 - 2\,\frac{(nm)^{n\bar{d}-1}}{(n\bar{d}-1)!}\, e^{-\frac{m\lambda^2}{4}}.$$

Proof. Theorem 3.3 with $S = P_d(X)$ and $k = s(m, d, n)$ provides

$$\Pr\Big(\|f - S\|_p \ge (1-\lambda)m\Big) \ge 1 - s(m, d, n)\, e^{-\frac{m\lambda^2}{4}}.$$

So by Theorem 5.1 we get

$$\Pr\Big(\|f - S\|_p \ge (1-\lambda)m\Big) \ge 1 - 2\,\frac{(nm)^{n\bar{d}-1}}{(n\bar{d}-1)!}\, e^{-\frac{m\lambda^2}{4}}. \qquad \square$$

Similarly, by Theorem 6.1 and the simplified upper bound on $s(m, d, n)$ stated in Corollary 5.3 we get the following probabilistic lower bound.

Corollary 6.2. *For every $X \subset \mathbb{R}^d$ such that* $\text{card}\, X = m$, $p \in \{1, 2\}$ *and every positive integer n,*

$$\Pr\Big(\|f - S\|_p \ge (1-\lambda)m\Big) \ge 1 - e^{-\frac{1}{2}\left(m\,\frac{\lambda^2}{2} - 3\,n\,\bar{d}\,\ln m\right)}.$$

Proof. By Theorem 6.1 and Corollary 5.3, we have

$$\begin{aligned}\Pr\Big(\|f - S\|_p \ge (1-\lambda)m\Big) &\ge 1 - n^{\frac{n\bar{d}}{2}}\, m^{n\bar{d}}\, e^{-\frac{m\lambda^2}{4}} \\ &\ge 1 - m^{\frac{3}{2}n\bar{d}} e^{-\frac{m\lambda^2}{4}} \\ &= 1 - e^{\frac{3}{2}\,n\,\bar{d}\,\ln m - m\,\frac{\lambda^2}{4}} \ge 1 - e^{-\frac{1}{2}\left(m\,\frac{\lambda^2}{2} - 3\,n\,\bar{d}\,\ln m\right)}. \qquad \square\end{aligned}$$

The exponent $-\frac{1}{2}\left(m\,\frac{\lambda^2}{2} - 3\,n\,\bar{d}\,\ln m\right)$ in the lower bound from Corollary 6.2 is positive when $m\,\frac{\lambda^2}{2} \ge 3\,n\,\bar{d}\,\ln m$. Thus we get the following condition:

$$n \le \frac{m}{\ln m}\, \frac{\lambda^2}{6\,\bar{d}}. \qquad (8)$$

For m and n satisfying the condition (8), Corollary 6.2 implies that almost any randomly-chosen function cannot be approximated by signum perceptron network with n units within error smaller than $(1 - \lambda)m$. For instance, when n depends polynomially on $\ln m$, e.g., when $n = (\ln m)^r$ for some positive integer r, then we get from (8) $(\ln\, m)^r \le \frac{m}{\ln\, m}\, \frac{\lambda^2}{6\,\bar{d}}$, i.e.,

$$\frac{m}{(\ln\, m)^{r+1}} \ge \frac{6\,\bar{d}}{\lambda^2}, \qquad (9)$$

which holds for sufficiently large values of m. When number of units is smaller than $\frac{m}{\ln m}\, \frac{\lambda^2}{6\,\bar{d}}$ then for almost any uniformly randomly chosen function on any set X with $\text{card}\, X = m$, a shallow network with signum perceptrons cannot achieve an approximation error smaller than $(1 - \lambda)m$.

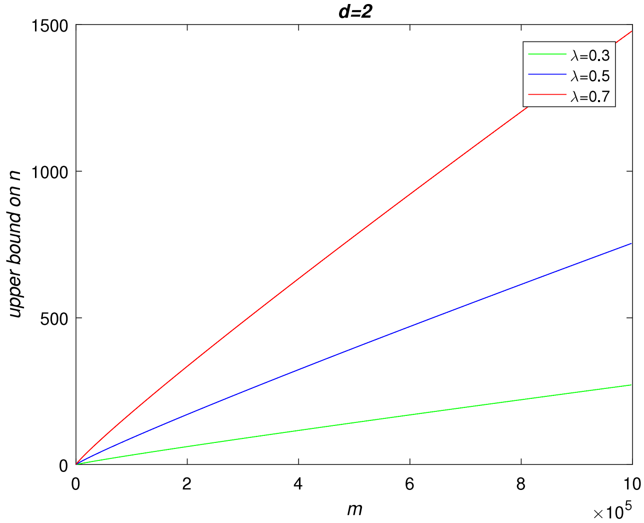

Fig. 2. The upper bound on n expressed by Eq. (8), for $d = 2$ and different values of λ.

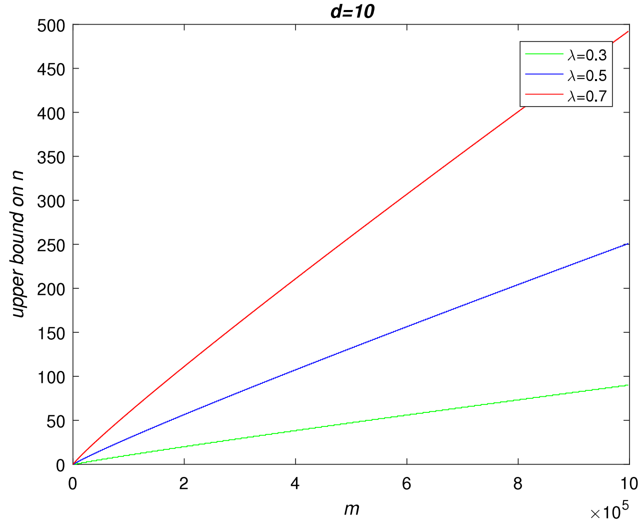

Fig. 3. The upper bound on n expressed by Eq. (8), for $d = 10$ and different values of λ.

Figs. 2–5 illustrate the growth of the quantity $\frac{m}{\ln m}\, \frac{\lambda^2}{6\,\bar{d}}$ in the upper bound (8) as a function of m, for different values of d and λ.

7. Discussion

Recently, various approaches to comparisons of capabilities of shallow and deep networks were proposed. A study based on topological concepts was developed by Bianchini and Scarselli (2014). They proved that topological characteristics (Betti's numbers) of input–output functions of certain deep networks grow exponentially with the number of hidden units, whereas for shallow networks with the same types of units they grow merely polynomially. Mhaskar et al. (2016a, 2016b) compared VC-dimensions of shallow and deep nets. They proved that deep networks can approximate a set of compositional functions with the same accuracy as shallow nets, but with exponentially lower VC-dimensions.

Earlier, a topological approach to derive lower bounds exhibiting the "curse of dimensionality", i.e., an exponential dependence

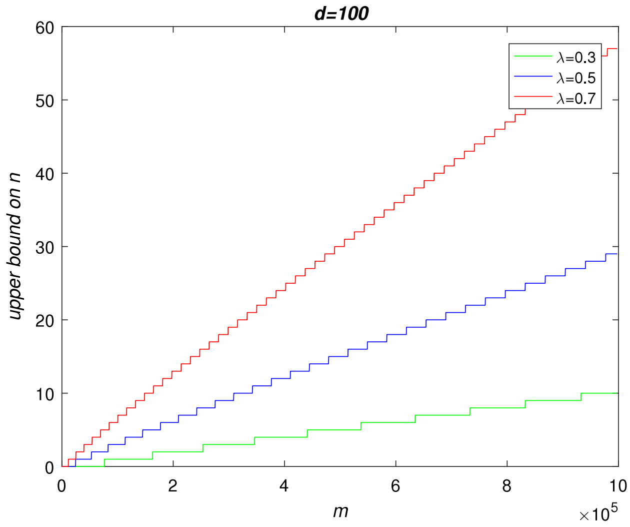

Fig. 4. The upper bound on n expressed by Eq. (8), for $d = 100$ and different values of λ.

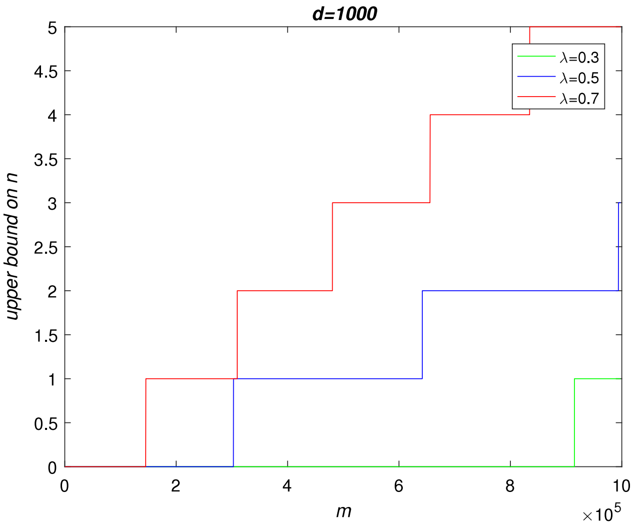

Fig. 5. The upper bound on n expressed by Eq. (8), for $d = 1000$ and different values of λ.

on the number of parameters (Bellman, 1957) was proposed by DeVore, Howard, and Micchelli (1989). They extended a well-known argument from linear approximation theory (Pinkus, 1985) to computational models that have continuous selections of best approximators. This property is typical for linear approximation, but shallow networks of many common types, due to their nonlinear and non-convex natures, do not allow continuous selections of best or even near best approximations, as it was proven by Kainen, Kůrková, and Vogt (1999, 2000, 2001). Actually, the nonlinearity and non-convexity of sets of input–output functions of neural networks are essential properties to make them better tools for high-dimensional tasks than classical linear approximators. Classes of functions for which classical linear approximators suffer from the curse of dimensionality, whereas some shallow networks do not exhibit this drawback, were described by Gnecco (2012, 2016) and Kůrková and Sanguineti (2002) (see also Gnecco, Kůrková, & Sanguineti, 2011a, 2011b; Kainen, Kůrková, & Sanguineti, 2009; Kainen et al., 2012; Klusowski & Barron, 2016; Kůrková & Sanguineti, 2001 and the references therein).

Bengio et al. (2006) suggested that a cause of large model complexities of shallow networks may be the "amount of variations" of functions to be computed. In Kůrková and Sanguineti (2016) we formalized the concept of highly-varying functions in terms of variational norms tailored to computational units and we derived probabilistic lower bounds on these norms. The arguments presented in Kůrková and Sanguineti (2016) were complemented in Kůrková (2017, in press) by concrete constructions of functions with large variations on square domains of sizes $n \times n$. These functions cannot be computed by shallow signum perceptron networks having both the number of units and sizes of all output weights smaller than $\frac{n}{\log_2 n}$, but some of these functions can be computed by two-hidden-layer networks with merely n units. Large variations guarantee large l_1-norms of the vectors of output weights, hence there must be either a large number of units or a large maximum output weight. Both are not desirable. It was remarked by Gorban, Tyukin, Prokhorov, and Sofeikov (2016, p. 144) that "we have to pay for such a significant reduction of the number of elements by ill-conditioning of the approximation problem".

8. Conclusions

Motivated by recent empirical studies comparing the model complexities of shallow and deep networks, we derived lower bounds on rates of approximation of functions by shallow signum perceptron networks, combining the probabilistic Chernoff–Hoeffding Bound with estimates of the sizes of sets of functions exactly computable by shallow networks.

The lower bounds are large for networks with numbers of units "considerably smaller" than the size m of the domain (more precisely, depending polynomially on $\ln m$). Our bounds hold for almost any uniformly randomly chosen function.

Our results are negative in the sense that they prove that one cannot do too much if the number of units does not exceed a polynomial in the logarithm of the size of the domain. However it should be remarked that, while Bengio (2009) and Bengio et al. (2006) proved deterministic statements for a specific function (namely, parity), we derived probabilistic estimates holding for almost any function. In contrast to worst-case results such as those in Maiorov and Meir (2000), ours apply to the average case.

Some of our results are quite general, depending only on the sizes of the approximating sets, so they have nothing to do with deep vs. shallow architecture. Hence, we expect that the bounds are not tight. Consequences of these general estimates hold for shallow signum perceptron networks, because the corresponding families of input–output functions are relatively small. The problem of whether or not it is possible to construct larger sets by arranging the same number of computational units in more layers is still open.

Acknowledgments

V.K. was partially supported by the Czech Grant Agency grant GA15-18108S and institutional support of the Institute of Computer Science RVO 67985807. M.S. is a member of GNAMPA-INdAM (Gruppo Nazionale per l'Analisi Matematica, la Probabilità e le loro Applicazioni—Instituto Nazionale di Alta Matematica).

References

Anastassiou, G. A. (2011). *Intelligent systems reference library: Vol. 19. Intelligent systems: approximation by artifcial neural networks*. Berlin: Springer-Verlag.
Gorban, A. N., Tyukin, I. Y., Prokhorov, D. V., & Sofeikov, K. I. (2016). Approximation with random bases: Pro et contra. *Information Sciences*, 129–145.
Anthony, M. (2001). *Discrete mathematics of neural networks: selected topics*. SIAM.
Ba, L. J., & Caruana, R. (2014). Do deep networks really need to be deep? In Z. Ghahrani, et al. (Eds.), *Advances in neural information processing systems. Vol. 27* (pp. 1–9).
Bellman, R. (1957). *Dynamic programming*. Princeton University Press.

Bengio, Y. (2009). Learning deep architectures for AI. *Foundations and Trends in Machine Learning*, *2*, 1–127.

Bengio, Y., Delalleau, O., & Le Roux, N. (2006). The curse of highly variable functions for local kernel machines. In *Advances in Neural Information Processing Systems. Vol. 18* (pp. 107–114). MIT Press.

Bengio, Y., & LeCun, Y. (2007). Scaling learning algorithms towards AI. In L. Bottou, O. Chapelle, D. DeCoste, & J. Weston (Eds.), *Large-scale kernel machines*. MIT Press.

Bianchini, M., & Scarselli, F. (2014). On the complexity of neural network classifiers: A comparison between shallow and deep architectures. *IEEE Transactions on Neural Networks and Learning Systems*, *25*, 1553–1565.

Chui, C.K., & Mhaskar, H.N. (2016). Deep nets for local manifold learning. arXiv preprint, arXiv:1607.07110.

Costarelli, D. (2015). Neural network operators: Constructive interpolation of multivariate functions. *Neural Networks*, *67*, 28–36.

Costarelli, D., & Vinti, G. (2016a). Approximation by max-product neural network operators of Kantorovich type. *Results in Mathematics*, *69*, 505–519.

Costarelli, D., & Vinti, G. (2016b). Max-product neural network and quasi interpolation operators activated by sigmoidal functions. *Journal of Approximation Theory*, *209*, 1–22.

Costarelli, D., & Vinti, G. (2016c). Pointwise and uniform approximation by multivariate neural network operators of the max-product type. *Neural Networks*, *81*, 81–90.

Cover, T. (1965). Geometrical and statistical properties of systems of linear inequalities with applictions in pattern recognition. *IEEE Transactions on Evolutionary Computation*, *14*, 326–334.

DeVore, R. A., Howard, R., & Micchelli, C. (1989). Optimal nonlinear approximation. *Manuscripta Mathematica*, *63*, 469–478.

Gnecco, G. (2012). A comparison between fixed-basis and variable-basis schemes for function approximation and functional optimization. *Journal of Applied Mathematics*, *2012*, 17. article ID 806945.

Gnecco, G. (2016). On the curse of dimensionality in the Ritz method. *Journal of Optimization Theory and Applications*, *168*, 488–509.

Gnecco, G., Kůrková, V., & Sanguineti, M. (2011a). Can dictionary-based computational models outperform the best linear ones? *Neural Networks*, *24*, 881–887.

Gnecco, G., Kůrková, V., & Sanguineti, M. (2011b). Some comparisons of complexity in dictionary-based and linear computational models. *Neural Networks*, *24*, 171–182.

Gripenberg, G. (2003). Approximation by neural networks with a bounded number of nodes at each level. *Journal of Approximation Theory*, *122*, 260–266.

Hahm, N., & Hong, B. I. (2016). A note on neural network approximation with a sigmoidal function. *Applied Mathematical Sciences*, *10*, 2075–2085.

Hinton, G. E., Osindero, S., & Teh, Y. W. (2006). A fast learning algorithm for deep belief nets. *Neural Computation*, *18*, 1527–1554.

Hoeffding, W. (1963). Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, *58*, 13–30.

Ito, Y. (1992). Finite mapping by neural networks and truth functions. *Mathematical Scientist*, *17*, 69–77.

Kainen, P. C., Kůrková, V., & Sanguineti, M. (2009). Complexity of Gaussian radial-basis networks approximating smooth functions. *Journal of Complexity*, *25*, 63–74.

Kainen, P. C., Kůrková, V., & Sanguineti, M. (2012). Dependence of computational models on input dimension: Tractability of approximation and optimization tasks. *IEEE Transaction on Information Theory*, *58*, 1203–1214.

Kainen, P. C., Kůrková, V., & Vogt, A. (1999). Approximation by neural networks is not continuous. *Neurocomputing*, *29*, 47–56.

Kainen, P. C., Kůrková, V., & Vogt, A. (2000). Geometry and topology of continuous best and near best approximations. *Journal of Approximation Theory*, *105*, 252–262.

Kainen, P. C., Kůrková, V., & Vogt, A. (2001). Continuity of approximation by neural networks in L_p-spaces. *Annals of Operations Research*, *101*, 143–147.

Klusowski, J.M., & Barron, A.R. (2016). Universal approximation by neural networks activated by first and second order ridge splines. arXiv preprint, arXiv:1607.07819v2.

Kůrková, V. (2017). Lower bounds on complexity of shallow perceptron networks. In C. Jayne, & L. Iliadis (Eds.), *Communications in Computer and Information Sciences (Proc. 17th Int. Conf. on Engineering Applications of Neural Networks)* (pp. 283–294). Springer.

Kůrková, V. (2017). Constructive lower bounds on model complexity of shallow perceptron networks. *Neural Computing and Applications*. http://dx.doi.org/10.1007/s00521017-2965-0 (in press).

Kůrková, V., & Sanguineti, M. (2001). Bounds on rates of variable-basis and neural-network approximation. *IEEE Transaction on Information Theory*, *47*, 2659–2665.

Kůrková, V., & Sanguineti, M. (2002). Comparison of worst-case errors in linear and neural network approximation. *IEEE Transaction on Information Theory*, *48*, 264–275.

Kůrková, V., & Sanguineti, M. (2016). Model complexities of shallow networks representing highly varying functions. *Neurocomputing*, *171*, 598–604.

LeCunn, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*, 436–444.

Ledoux, M. (2001). *The concentration of measure phenomenon*. Providence: AMS.

Maiorov, V. E., & Meir, R. (2000). On the near optimality of the stochastic approximation of smooth functions by neural networks. *Advances in Computational Mathematics*, *13*, 79–103.

Mhaskar, H., Liao, Q., & Poggio, T. (2016a). Learning functions: When is deep better than shallow. CBMM Memo No. 045, May 31, 2016. https://arxiv.org/pdf/1603.00988v4.pdf.

Mhaskar, H., Liao, Q., & Poggio, T. (2016b). Learning real and Boolean functions: When is deep better than shallow. CBMM Memo No. 45, March 4, 2016, https://arxiv.org/pdf/1603.00988v1.pdf.

Petrov, V. V. (1995). *Limit theorems of probability theory*. Oxford: Clarendon Press.

Pinkus, A. (1985). *n-widths in approximation theory*. Berlin, Heidelberg: Springer.

Pinkus, A. (1999). Approximation theory of the MLP model in neural networks. *Acta Numerica*, *8*, 143–195.

Rojas, R. (1996). *Neural networks: a systematic introduction*. New York: Springer.

Sanguineti, M. (2008). Universal approximation by ridge computational models and neural networks: A survey. *The Open Applied Mathematics Journal*, *2*, 31–58.

Schläfli, L. (1901). *Theorie der vielfachen kontinuität*. Zürich: Zürcher & Furrer.

Winder, R. O. (1962). *Threshold Logic*. (Doctoral Dissertation), Mathematics Department, Princeton University.