Information Sciences 364–365 (2016) 129–145

Contents lists available at ScienceDirect

Information Sciences

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

Approximation with random bases: Pro et Contra

Alexander N. Gorban[a], Ivan Yu. Tyukin[a,b,1,*], Danil V. Prokhorov[c], Konstantin I. Sofeikov[a]

[a] Department of Mathematics, University of Leicester, University Road, Leicester, LE1 7RH, UK
[b] Department of Automation and Control Processes, St. Petersburg State University of Electrical Engineering, Prof. Popova str. 5, Saint-Petersburg 197376, Russian Federation
[c] Toyota Research Institute NA, Ann Arbor, MI 48105, USA

ARTICLE INFO

Article history:
Received 15 June 2015
Revised 31 August 2015
Accepted 14 September 2015
Available online 25 September 2015

Keywords:
Random bases
Measure concentration
Neural networks
Approximation

ABSTRACT

In this work we discuss the problem of selecting suitable approximators from families of parameterized elementary functions that are known to be dense in a Hilbert space of functions. We consider and analyze published procedures, both randomized and deterministic, for selecting elements from these families that have been shown to ensure the rate of convergence in L_2 norm of order $O(1/N)$, where N is the number of elements. We show that both randomized and deterministic procedures are successful if additional information about the families of functions to be approximated is provided. In the absence of such additional information one may observe exponential growth of the number of terms needed to approximate the function and/or extreme sensitivity of the outcome of the approximation to parameters. Implications of our analysis for applications of neural networks in modeling and control are illustrated with examples.

1. Introduction

The problem of efficient representation and modeling of data is important in many areas of science and engineering. A typical problem in this area involves constructing quantitative models (maps) of the type

$$x_1, x_2, \ldots, x_d \mapsto f(x_1, x_2, \ldots, x_d),$$

where $x_1, x_2, \ldots, x_d$, $x_i \in \mathbb{R}$, $i = 1, \ldots, d$ are variables and $f : \mathbb{R}^d \to \mathbb{R}$ is an unknown functional relation between the variables. The total number, d, of variables, determining input data may be large, and physical models of such relations $f(\cdot)$ are not always available.

In the absence of acceptably detailed prior knowledge of “true” models, $f(\cdot)$, a commonly used alternative is to express the function $f(\cdot)$ as a linear combination of known functions, $\varphi_i(\cdot)$, $\varphi : \mathbb{R}^d \to \mathbb{R}$:

$$f(x) \simeq f_N(x) = \sum_{i=1}^{N} c_i \varphi_i(x), \quad c_i \in \mathbb{R}. \tag{1}$$

[*] Corresponding author at: Department of Mathematics, University of Leicester, University Road, Leicester, LE1 7RH, UK. Tel.: +44 1162525106.
E-mail addresses: ag153@le.ac.uk (A.N. Gorban), I.Tyukin@le.ac.uk (I.Yu. Tyukin), dvprokhorov@gmail.com (D.V. Prokhorov), sofeykov@gmail.com (K.I. Sofeikov).

[1] The work was supported by Innovate UK Technology Strategy Board (Knowledge Transfer Partnership grant KTP009890) and Russian Foundation for Basic Research (research project No. 15-38-20178).

http://dx.doi.org/10.1016/j.ins.2015.09.021
0020-0255/© 2015 Elsevier Inc. All rights reserved.

Numerous classes of functions $\varphi_i(\cdot)$ in (1) have been proposed and analysed to date, starting from $\sin(\cdot)$, $\cos(\cdot)$ and polynomial functions featured in classical Fourier, Fejer, and Weierstrass results, wavelets [32,37], and reaching out to linear combinations of sigmoids [7]

$$f_N(x) = \sum_{i=1}^{N} c_i \frac{1}{1+e^{-(w_i^T x + b_i)}}, \quad w_i \in \mathbb{R}^d, \quad b_i \in \mathbb{R}, \tag{2}$$

radial basis functions [27]

$$f_N(x) = \sum_{i=1}^{N} c_i e^{(-\|w_i^T x + b_i\|^2)} \; w_i \in \mathbb{R}^d, \quad b_i \in \mathbb{R}, \tag{3}$$

and other general functions [10] that are often used in neural network literature [14]. Sometimes the values of parameters w_i, b_i may be pre-selected on the basis of additional prior knowledge, leaving only linear weights c_i for training. If no prior information is available then both nonlinear (w_i and b_i) and linear (c_i) parameters, or weights, are typically subject to training on data specific to the problem at hand (full network training).

One special feature that makes full training of approximators (2), (3) particularly attractive is that in addition to their universal approximation capabilities [7,17,27] and their homogenous structure, they are reported to be efficient when the dimension d of input data is relatively high. In particular, if *all parameters* w_i, c_i, b_i are allowed to vary, the order of convergence rate of the approximation error of a sufficiently smooth function $f(\cdot) \in \mathcal{C}^0[0,1]^d$ as a function of N (the number of elements in the network) is shown to be independent of the input dimension d [4,19]. Furthermore, the achievable rate of convergence of the L_2-norm of $f(\cdot) - f_N(\cdot)$ is shown to be $O(1/N^{1/2})$. This contrasts sharply with the rate $O(d^{-1}/N^{1/d})$ corresponding to the worst-case estimate inherent to linear combinations (1) with $\varphi_i(\cdot)$ given or (2), (3) with w_i, b_i fixed. In particular, it is shown in [4] that if only linear parameters of (2), (3) are adjusted the approximation error *cannot be made* smaller than $Cd^{-1}/N^{1/d}$, where C is independent on N, uniformly for functions satisfying the same smoothness constraints.

Favorable independence of the order of convergence rates on the input dimension of the function to be approximated, however, comes at a price. Construction of such efficient models of data involves a nonlinear optimization routine searching for the best possible values of w_i, b_i. The necessity to adjust parameters entering (2) nonlinearly, (3) restricts practical application of these models in a range of relevant problems (see e.g., [25,29,38,40] for an overview of potential issues).

An alternative to adjustment of nonlinear parameters of (2), (3) has been proposed in [18,26,36] and further developed in [30,31] (see also earlier work by Rosenblatt [33] in which he discussed perceptrons with random weights). In these works the nonlinear parameters w_i and b_i are proposed to be set randomly at the initialization, rather than through training. The only trainable parameters are those which enter the network equation linearly, c_i. Random selection of weights in the hidden layers is supposed to generate a set of (basis) functions that is sufficiently rich to approximate a given function by mere linear combinations from this set. This crucially simplifies training in systems featuring such approximators, and renders an otherwise computationally complex nonlinear optimization problem into a much simpler linear one. Moreover, the rate of error convergence is argued to be $O(1/N^{1/2})$ [18,30,31].

This brings us to a paradoxical contradiction: on the one hand the reported convergence rate $O(1/N^{1/2})$ which a randomized approximator is supposed to achieve contradicts to the earlier worst-case estimate $O(d^{-1}/N^{1/d})$ obtained in [4]. On the other hand numerous studies report successful application of such randomization ideas (see also comments [23]) in a variety of intelligent control applications [15,21,22,24,35] as well as in machine learning (see e.g. [1,41] and references therein). The question, therefore, is if it is possible to resolve such an apparent controversy? Answering to this question is the main purpose of this contribution.

The paper is organized as follows. We begin the analysis with Section 3 in which we review basic reasoning in [18] and compare these results with [4,19]. We show that, although these results may seem inconsistent, they have been derived for different performance criteria. The worst-case estimate in [4] is "crisp", whereas the convergence rate in [18] is probabilistic. That is it involves a measure function with respect to which the rate $O(1/N^{1/2})$ is assured. Introduction of measure into the problem brings out a range of interesting consequences that is discussed and analyzed in Section 4. There we approach data approximation problem as that of representation of a given vector by a linear combination of randomly chosen vectors in high dimensions. A simple logic suggests that if one takes $m \leq n$ random vectors in $\mathbb{R}^n$ then it can be expected that with probability one these vectors will be linearly independent because the set of linearly dependent corteges $\{x_1, \dots, x_m\}$ $(m \leq n)$ is a proper algebraic subset in the space of corteges $(\mathbb{R}^n)^m$. We can select n random vectors, and with probability one it will be a basis. Every given data vector y can be represented by coordinates in this basis. If these n vectors are (accidentally) too close to dependence we can generate few more vectors that will enable us to represent the data vector y. We will show, however, that this simple and correct reasoning loses its credibility in high dimensions. We show that in an n-dimensional unit cube $[-1,1]^n$ a randomly generated vector x will be almost orthogonal to a given data vector y (the angle between x and y will be close to $\pi/2$ with probability close to one). To compensate for this *waist concentration* effect [11–13] one needs to generate exponentially many random vectors. Typicality of such exponential growth is an inherent feature of high-dimensional data representation, including problems of data approximation and modeling. Moreover, for high-dimensional data representation the following two seemingly contradictory situations are typical in some sense:

- *with probability close to one linear combinations of $n-k$ random vectors approximate any normalized vector with accuracy ε if $k \ll n$ and no constraints on the values of coefficients in linear combinations are imposed (and this probability is one, if $k = 0$);*