Neural Networks 21 (2008) 1311–1317

Contents lists available at ScienceDirect

Neural Networks

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

A systematic investigation of a neural network for function approximation

Leila Ait Gougam, Mouloud Tribeche*, Fawzia Mekideche-Chafa

Theoretical Physics Laboratory, Faculty of Sciences-Physics, University of Bab-Ezzouar, USTHB, B.P. 32, El Alia, 16111, Algiers, Algeria

ARTICLE INFO

Article history:
Received 5 July 2007
Revised and accepted 19 June 2008

Keywords:
Function approximation
Neural network
Wavelet
Gradient decent
Activation function
Response function
Formal neuron
Scale parameters merging
Kolmogorov theorem

ABSTRACT

A model which takes advantage of wavelet-like functions in the functional form of a neural network is used for function approximation. The scale parameters are mainly used, neglecting the usual translation parameters in the function expansion. Two training operations are then investigated. The first one consists of optimizing the output synaptic weights and the second one on optimizing the scale parameters hidden inside the elementary tasks. Building upon previously published results, it is found that if $(p+1)$ scale parameters merge during the learning process, derivatives of order p will emerge spontaneously in the functional basis. It is also found that for those tasks which induce such mergings, the function approximation can be improved and the training time reduced by directly implementing the elementary tasks and their derivatives in the functional basis. Attention has been also devoted to the role transfer functions, number of iterations, and formal neurons number may play during and after the learning process. The results complement previously published results on this problem.

1. Introduction

For several years now, neural network models have enjoyed wide popularity, being applied to problems of regression, classification, computational science, computer vision, data processing and time series analysis (Haykin, 1994; Hopfield & Tank, 1986; Lippmann, 1987). They have been also successfully used for the identification and the control of dynamical systems, mapping the input–output representation of an unknown system and, possibly, its control law (Narendra & Parthasaranthy, 1990). The problem of determining the analytical description for a set of data arises in numerous sciences and applications and can be referred to as data modeling or system identification. Neural networks are a convenient means of representation because they are known to be universal approximators (Hornik, Stinchcombe, & White, 1989) that can learn data. The desired task is usually obtained by a learning procedure which consists in adjusting the "synaptic weights". For this purpose, many learning algorithms have been proposed to update these weights. The convergence for these learning algorithms is a crucial criterion for neural networks to be useful in different applications (Arik, 2005; Liang & Cao, 2004). In fact, considerable effort has gone into developing techniques for accelerating the convergence of the training algorithms. In addition, there is a growing understanding that the choice of transfer functions is at least as important as the network architecture and learning algorithm. Sigmoids (Cybenko, 1988; Hornik et al., 1989) have been the first proposed transfer functions. It has been proved by many authors that they can approximate an arbitrary continuous function on a compact domain with arbitrary precision given sufficient number of neurons. Despite the fact that sigmoids are the most commonly used functions, there is no *a priori* reason why they should be optimal in all cases. For this purpose, several transfer functions have been proposed and we just mention Gaussian (Hartman, Keeler, & Kowalski, 1990), Lorentzian (Giraud, Lapedes, Liu, & Lemm, 1995), plane wave (Giraud, Liu, Bernard, & Axelrad, 1991), and rational fraction ones (Leung & Haykin, 1993).

During the last two decades, function approximation based on wavelets (Benveniste & Zhang, 1992; Oussar & Dreyfus, 2000; Zhang, Walter, Miao, & Lee, 1995) has attracted a great deal of interest as a very good alternative to more classical techniques. In fact, wavelets became a necessary mathematical tool in many investigations. Analysis and processing of different classes of nonstationary or inhomogeneous signals is the main field of applications of wavelet analysis. Because of their unique properties, wavelets were used in functional analysis in mathematics, in studies of multi-fractal properties, singularities and local oscillations of functions, for solving some differential equations, for investigation of inhomogeneous processes involving widely different scales of interacting perturbations, for pattern recognition, for image and sound compression (Erlebacher, Hussaini, & Jameson, 1996; Mallat, 1998). The most general principle of the wavelet construction is to use dilations and translations. Commonly used wavelets form a complete orthonormal system of functions with

* Corresponding author.
E-mail addresses: ag_leila@yahoo.fr (L. Ait Gougam), mouloud-tribeche@lycos.com (M. Tribeche), fazia_mekideche@yahoo.fr (F. Mekideche-Chafa).

0893-6080/$ – see front matter © 2008 Elsevier Ltd. All rights reserved.
doi:10.1016/j.neunet.2008.06.015

a finite support constructed in such a way. That is why by changing a scale (dilations) they can distinct the local characteristics of a signal at various scales, and by translations they cover the whole region in which it is studied. It has been proven that any function can be written as a superposition of admissible wavelets, and there exists a numerically stable algorithm to compute the coefficients for such an expansion. Moreover, these coefficients completely characterize the function, and it is possible to reconstruct it in a numerically stable way by knowing these coefficients. Wavelets constitute a class of functions that satisfy a set of important mathematical properties. Consider, without loss of generality, that these functions are limited, have a compact support and that their translations and dilations generate a basis of functions for the representation of any element within a family of square integrable functions. It was proved that these families of functions are universal approximators (Daubechies, 1992). Therefore, wavelet networks can be considered as an alternative to neural and radial basis function networks.

Recently, Giraud and Touzeau (2001), used a well-known simplified architecture to show that the latter provides a reasonably efficient, practical and robust, multifrequency analysis. The training algorithm, optimizing the task with respect to the widths of the responses, revealed two distinct training modes. The first one induced some of the formal neurons to become identical, and the other one kept them distinct. In Giraud and Touzeau (2001), the suggestion has been made that the first training mode may represent new neural units, which are the derivatives of the elementary task with respect to their scale parameters. The aim of this paper is therefore to explore this suggestion, within the theoretical framework of Giraud and Touzeau (2001) to which the reader is referred. It is also legitimate to wonder about the role transfer function and the dimension of the elementary task basis may play on the approximation of a given target task.

In the present paper, we use wavelets stemming from the continuous wavelet transform. This approach uses a set of formal neurons (wavelets) with various translation parameters b and scale ones λ, hence a space of elementary responses translated by b and scaled with λ. The function to be approximated is then expanded in this set. The integral form of the expansion is then reduced to a discrete sum where the development coefficients are the output synaptic weights and are the unknowns of the problem.

The manuscript is organized as follows. In the next section, the model and the architecture of the neural network are succinctly outlined. Two training operations are then presented, optimizing both the output synaptic weights and the scale parameters. A numerical implementation is carried out and discussed in Section 3. A summary of our results and findings is given in Section 4.

2. Network architecture and training

Before proceeding further, it is well to give an outline of the model and the architecture of the neural network that emanates directly from the Kolmogorov's seminal results (Kolmogorov, 1956, 1957). The emphasis is on Kolmogorov's second theorem, which subsequently has been reformulated by Kurkova (1992) in more practical ways for artificial neural networks applications with somewhat relaxed requirements on approximating functions employed. The Kolmogorov's theorem states that *any function continuous on the n-dimensional unit cube $E^n, E = [0, 1]$ can be represented in the form*

$$f(x_1, x_2, \ldots, x_n) = \sum_{j=1}^{2n+1} \Psi_j \left(\sum_{i=1}^{n} \phi_{ij}(x_i) \right) \tag{1}$$

where Ψ_j and ϕ_{ij} are real continuous functions of one variable, and the functions ϕ_{ij} are independent of the given function f while only the functions Ψ_j are specific for the given function. Later, Hecht-Nielsen (1987) pointed out a resemblance between the formal structure of Kolmogorov's expansion of continuous functions through other auxiliary functions with three layer feed-forward neural networks and reformulated the Kolmogorov's theorem as *any continuous function defined on the n-dimensional cube $E^n, E = [0, 1]$ can be implemented exactly by a three-layered network having $2n + 1$ units in the hidden layer with transfer functions of a sigmoidal type from the input to the hidden layer.*

In other words, a network with one hidden layer should be able to describe any continuous function using $(2n + 1)$ nodes with n being the number of inputs. Of course, these theoretical minima may in practice be achieved only at the expense of a large number of hidden units.

Going parallel to what has been stated above, let us consider a three-layered network consisting of an input X, a hidden layer of elementary responses and a linear output neuron. The neural elementary units receive the same input X. Each unit returns an output f which depends on two parameters: the translation parameter b and the scale one λ. Output synaptic weights $\omega(b, \lambda)$ linearly regroup these elementary outputs into a global output $F(X)$. We have then the expansion (Mallat, 1989)

$$F(X) = \int \omega(b, \lambda) f\left(\frac{X - b}{\lambda}\right) \mathrm{d}b\mathrm{d}\lambda. \tag{2}$$

We first consider the case when the translation parameter is neglected and discuss later the case when it is included. Hence, discretizing Eq. (2) with N units and neglecting the translation parameter, we obtain the following approximation function

$$F_{\text{app}}(X) = \sum_{i=1}^{N} \omega(\lambda_i) f\left(\frac{X}{\lambda_i}\right). \tag{3}$$

To define the "best" F_{app}, one has to minimize the square norm of the error $F - F_{\text{app}}$, first in terms of the output weights ω_i and second of the scale parameters λ_i. In terms of the ω_i's, this consists in solving the equation

$$\frac{\partial}{\partial \omega_i} \left(\left\langle F - F_{\text{app}} \mid F - F_{\text{app}} \right\rangle \right) = 0. \tag{4}$$

Using Eq. (3) and adopting short notation, we obtain

$$\frac{\partial}{\partial \omega_i} \left(\langle F \mid F \rangle - 2\sum_{j=1}^{N} \omega_j \left\langle f_j \mid F \right\rangle + \sum_{j,k=1}^{N} \omega_j \left\langle f_j \mid f_k \right\rangle \omega_k \right) = 0$$
$$i = 1, \ldots, N. \tag{5}$$

the solution of which is given by

$$\omega_i = \sum_{j=1}^{N} \left(g^{-1}\right)_{ij} \left\langle f_j \mid F \right\rangle, \quad i = 1, \ldots, N \tag{6}$$

where g is the matrix with elements

$$g_{jk} = \left\langle f_j \mid f_k \right\rangle. \tag{7}$$

Let now minimize the mean square norm of the error (MSE) $\varepsilon = \left\langle F - F_{\text{app}} \mid F - F_{\text{app}} \right\rangle$ in terms of scale parameters λ_i. Note that we need only the derivatives of f_i with respect to their scales λ_i. We have thus

$$\frac{\partial \varepsilon}{\partial \lambda_j} = \frac{2\omega_j}{\lambda_j^2} \left\langle X f'\left(\frac{X}{\lambda_j}\right) \mid F - F_{\text{app}} \right\rangle = 0, \quad j = 1, \ldots, N \tag{8}$$

where f' represents the derivative of the elementary task, before any scaling. For the latter (also refereed to as the transfer function) several choices are possible. In the following numerical analysis, three transfer functions will be tested, viz., a sigmoidal function, a Gaussian one, and two wavelets.