Neurocomputing 166 (2015) 96–108

Contents lists available at ScienceDirect

Neurocomputing

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

Computational cost improvement of neural network models in black box nonlinear system identification

Hector M. Romero Ugalde [a,*], Jean-Claude Carmona [b], Juan Reyes-Reyes [c], Victor M. Alvarado [c], Juan Mantilla [a]

[a] Laboratoire Traitement du Signal et de l'Image, Université de Rennes 1, LTSI, and INSERM, U1099, 35042 Rennes, France
[b] Laboratoire des Sciences de l'Information et des Systemes, UMR CNRS 7296, ENSAM, 13100 Aix en Provence, France
[c] Centro Nacional de Investigación y Desarrollo Tecnológico, CENIDET, 62490 Cuernavaca, Mor., Mexico

ARTICLE INFO

Article history:
Received 5 May 2014
Received in revised form
23 March 2015
Accepted 10 April 2015
Communicated by W. Yu
Available online 20 April 2015

Keywords:
Non-linear system identification
Black box
Neural networks
Computational cost reduction
Estimation quality

ABSTRACT

Models play an important role in many engineering fields. Therefore, the goal in system identification is to find the good balance between the accuracy, complexity and computational cost of such identification models. In a previous work (Romero-Ugalde et al., 2013 [1]), we focused on the topic of providing balanced accuracy/complexity models by proposing a dedicated neural network design and a model complexity reduction approach. In this paper, we focus on the reduction of the computational cost required to achieve these balanced models. More precisely, the improvement of the preceding method presented here leads to a significantly computational cost reduction of the neural network training phase. Even if this reduction is achieved by a convenient choice of the activation functions and the initial conditions of the synaptic weights, the proposed architecture leads to a wide range of models among the most encountered in the literature assuring the interest of such a method. To validate the proposed approach, two different systems are identified. The first one corresponds to the unavoidable Wiener–Hammerstein system proposed in SYSID2009 as a benchmark. The second system is a flexible robot arm. Results show the interest of the proposed reduction methods.

1. Introduction

A model is a mathematic representation of a real system which can be constructed according to two ways or a combination of them [2]. An approach is based on the physical mechanisms that govern the system's behavior. The models thus achieved are adequate approximations of the real process [3]. But, in many cases, involving complex nonlinear systems, it is very difficult or impossible to derive dynamic models based on all the physical processes involved [4–6]. On the contrary, black box system identification techniques use general mathematical approximation functions to describe the systems input/output relations. One of the most important advantages of these approaches is the limited physical insights required to develop the model [7], but as a trade-off, these techniques imply the use of model structures that are as flexible as possible. Often, this flexibility leads to a high number of parameters [8].

Neural networks are suitable for modeling complex nonlinear systems when we consider the plant as a black-box [9–13]. However, it is well known that they require a large number of neurons to deal with complex systems [14]. Numerous neurons favor a better approximation but lead to a more complex model [15,16], and higher computational cost. In engineering applications, such as inverse control, the adaptive controllers have the same complexity as the reference models [17–20]. If low order reference models are used, the number of parameters to be computed will be rather small [21]. To address this problem many works tried to derive balanced accuracy, complexity and computational cost models.

Let us present some works which tend to find the best trade-off between the model complexity and approximation accuracy by finding the "optimal" number of neurons. Trial and error is one of these techniques. Based on this approach many authors [7,22,23] proposed models with a good "quality level". In the sequel, we shall denote for greatest convenience, "quality" as the balance between accuracy and complexity of the model. Although this procedure is laborious and it may not lead to the "best compromise" between the model complexity and the approximation accuracy [7,24]. Pruning based techniques have been successfully used for structural optimization [25–27]. In this approach [28,29],

[*] Corresponding author.
E-mail address: hector@cenidet.edu.mx (H.M. Romero Ugalde).

http://dx.doi.org/10.1016/j.neucom.2015.04.022
0925-2312/© 2015 Elsevier B.V. All rights reserved.

besides optimizing the number of neurons, the connections between the neurons are also optimized. More recently, other evolutionary techniques have been employed in order to derive “optimal” structures, for example, genetic algorithms (GAs) in [14,30,24], dissimilation particle swarm optimization (PSO) in [31], genetic programming (GP) in [4], a combination of GA and singular value architectural recombination (SVAR) in [32]. As the pruning approach, the previously outlined techniques, based on the evolution of the neural network, have been successfully applied for structural optimization. They share, however, the disadvantage of excessive time consumption needed to find the most convenient number of neurons, since training is required each time the neural network is restructured [11]. Moreover, to solve the problem of finding the best trade-off between model complexity and model accuracy, a rather subjective criterion is always used to decide whether the evolution of the neural network is appropriate and sufficient. Other techniques trying to solve the same problem are based on the design of the neural network. In [33] a novel time-delay recurrent neural network (TDRNN) is proposed to generate a simple structure. In [1] a neural network design and a model reduction approach are proposed in order to generate balanced accuracy/complexity models [13]. In [34] a neural network using a competitive scheme is proposed in order to provide an effective method with less network complexity. In [35] the selection of an appropriate FLANN (Functional Link Artificial Neural Network) structure as the backbone of the model offers low complexity by means of a single layer ANN structure. In [36] a pipeline bilinear recurrent neural network (PBLRNN) is proposed in order to reduce both the model and computational complexities of a bilinear recurrent neural network (BLRNN).

Now, let us present some works devoted to improve the balance between the accuracy and the computational cost of the model. In the sequel, we shall denote “price” as the computational cost to generate it [3], for convenience. This can be achieved by the implementation of an efficient learning algorithm or by a convenient choice of the model structure. Different learning algorithms can be applied in order to improve the accuracy and reduce the computational burden. In [37] a Kalman filter-based algorithm with faster convergence is proposed, although this algorithm is more complex than the gradient based algorithms, a decoupling technique is used to decrease the computational burden. Rubio and Pacheco [38] propose a combination of clustering, gradient and Kalman filter algorithms leading to a quick and efficient approach for modeling. A Bounding Ellipsoid algorithm is proposed in [39] for high computational efficiency and fast convergence. Extreme learning machine (ELM), which consists in randomly choosing the hidden weights and analytically determining the output weights, is proposed in [40] to reduce the computational cost required to train single layer neural networks (SLFN). An improved simultaneous perturbation stochastic approximation (SPSA) algorithm [41] yields an improved model in terms of time of convergence and a smaller identification error. In [42] a combination of GAs and Levenberg-Marquardt (LM) algorithms takes advantage of the global search of GA and the estimation ability of LM to improve the accuracy and reduce the computation time. Subudhi et al. [43] propose two Memetic Algorithms (MA), combining evolutionary algorithms (i.e. GA and differential evolution (DE), which are global search methods) with a backpropagation (BP) learning algorithm. These algorithms have faster convergence in comparison with only evolutionary computation and avoid the possibility of local minima normally existing in the gradient algorithms. This result is extended in [44] where the PSO is combined with BP too. Following the same idea, an opposition based differential evolution (ODE) algorithm combined with LM is used by Subudhi et al. [11] for training the feedforward neural network. The results obtained in the previous works are interesting owing to its convergence properties. However, since the improvements are achieved by the manipulation of the learning algorithms, the accuracy, the complexity and the price of the model are still affected by the complexity of the neural network structure. In [45,36] the improvements of the computational cost and/or the approximation accuracy are based on the choice of a convenient model structure, that is, by the design of the neural network.

With the conviction that the improvement of the quality and price of a model is linked to a suitable neural network design, we decided to tackle the problem as follows: firstly, by using the neural network design and the model reduction approach proposed in [1], and secondly by proposing a computational cost reduction approach. Combining these two approaches, we propose a black box system identification method which yields to accurate models with a small number of parameters at a low price. As presented in [1], the model complexity reduction approach is developed in two steps: the first step consists in training a three-layers neural network chosen to tolerate an initial large number of neurons. In a second step, the three-layers architecture is transformed into a two-layer representation with a significant reduced number of neurons retaining the approximation accuracy of the previous three-layers model. The main contribution of this paper consists in the improvement of the first step of the model reduction approach, that is, the computational cost required to train the complex three layers neural network in order to significantly reduce it. Even if it depends on the particular selection of two factors: (a) the activation functions in each layer and (b) the initial conditions of the synaptic weights, the reader shall notice that the proposed architecture nevertheless remains sufficiently general to provide a wide range of useful model types [13] with a good quality at a low price. These model types are currently used for model-based control techniques [10,46–49,21]. The learning algorithm used to optimize the synaptic weights is a classical steepest descent algorithm with a back propagation configuration.

The paper is organized as follows: Section 2 is devoted to describe the new neural network structure which allows us to investigate the balance among complexity, Section 3 the theorem and corollary on which our model complexity reduction approach and our computational complexity reduction approach are based are given. In Section 4, the paper discusses the results of the identification of a benchmark system and a robot arm. Subsequently conclusions and perspectives are given in Section 5.

2. Neural network design

As mentioned above the neural network design proposed in [1] is presented in this section. Fig. 1 shows a three layers neural network with $2 \times nn$ neurons in the input layer, 2 neurons in the hidden layer and 1 neuron in the output layer ($2nn$–2–1 neural network). Notice that the number of neurons in the hidden layer is fixed and the number of neurons in the first layer (nn neurons used to process the regressors input vector and nn neurons used to process the regressors output vector) is chosen by the user. After the training phase, this particular configuration allows us to reduce the $2nn$–2–1 neural network into the 2–1 neural network shown in Fig. 2, and moreover this architecture allows us to reduce the computational cost required during the training phase into the one required to train the 2–2–1 architecture shown in Fig. 3. As we shall see, even if this structure is somehow particular, by different combinations of activation functions, the proposed architecture allows us to generate easily the classical models presented in [47]. The reader shall notice that this architecture remains sufficiently general, from a user point of view. The mathematical representation of the proposed neural network architecture is given by the