Neurocomputing 129 (2014) 428–436

Contents lists available at ScienceDirect

Neurocomputing

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

Learning of a single-hidden layer feedforward neural network using an optimized extreme learning machine

Tiago Matias [a,b,*], Francisco Souza [a,b], Rui Araújo [a,b], Carlos Henggeler Antunes [b,c]

[a] Institute of Systems and Robotics (ISR-UC), University of Coimbra, Pólo II, PT-3030-290 Coimbra, Portugal
[b] Department of Electrical and Computer Engineering (DEEC-UC), University of Coimbra, Pólo II, PT-3030-290 Coimbra, Portugal
[c] Institute for Systems Engineering and Computers (INESC), R. Antero de Quental, 199, PT-3000-033 Coimbra, Portugal

ARTICLE INFO

Article history:
Received 18 January 2013
Received in revised form
1 August 2013
Accepted 2 September 2013
Communicated by A. Abraham
Available online 23 October 2013

Keywords:
Optimized extreme learning machine
Single-hidden layer feedforward neural networks
Genetic algorithms
Simulated annealing
Differential evolution

ABSTRACT

This paper proposes a learning framework for single-hidden layer feedforward neural networks (SLFN) called optimized extreme learning machine (O-ELM). In O-ELM, the structure and the parameters of the SLFN are determined using an optimization method. The output weights, like in the batch ELM, are obtained by a least squares algorithm, but using Tikhonov's regularization in order to improve the SLFN performance in the presence of noisy data. The optimization method is used to the set of input variables, the hidden-layer configuration and bias, the input weights and Tikhonov's regularization factor. The proposed framework has been tested with three optimization methods (genetic algorithms, simulated annealing, and differential evolution) over 16 benchmark problems available in public repositories.

1. Introduction

Multilayer feedforward neural networks (FFNN) have been used in the identification of unknown linear or non-linear systems (see, e.g. [1,2]). Their appeal is based on their universal approximation properties [3,4]. However, in industrial applications, linear models are often preferred due to faster training in comparison with multilayer FFNN trained with gradient-descent algorithms [5]. In order to overcome the slow construction of FFNN models, a new method called extreme learning machine (ELM) is proposed in [6]. This method is a new batch learning algorithm for single-hidden layer FFNN (SLFN) where the input weights (weights of connections between the input variables and neurons in the hidden-layer) and the bias of neurons in the hidden-layer are randomly assigned. The output weights (weights of connections between the neurons in the hidden-layer and the output neuron) are obtained using the Moore–Penrose (MP) generalized inverse, considering that the activation function of the output neuron is linear.

Since in ELM the output weights are computed based on the random input weights and bias of the hidden nodes, there may exist a set of non-optimal or unnecessary input weights and bias of the hidden nodes. Furthermore, the ELM tends to require more hidden neurons than conventional tuning-based learning algorithms (based on error backpropagation or other learning methods where the output weights are not obtained by the least squares method) in some applications, which can negatively affect SLFN performance in unknown testing data [6]. The use of the least squares method without regularization in noisy data also makes the model displaying a poor generalization capability [7]. Fitting problems may also be encountered in the presence of irrelevant or correlated input variables [5].

Optimization methods have been used jointly with analytical methods for network training. In [8] a new method to choose the most appropriate FFNN topology, type of activation functions and parameters of the training algorithm using a genetic algorithm (GA) is proposed. Each chromosome is composed of the specification of the minimization algorithm used in the back-propagation (BP) method, the network architecture, the activation function of the neurons of the hidden layer, and the activation function of the neurons of the output layer using binary encoding. In [9] a new nonlinear system identification scheme is proposed, where differential evolution (DE) is used to optimize the initial weights used by a Levenberg–Marquardt (LM) algorithm in the learning of a FFNN. In [10] a similar method is proposed using a simulated annealing (SA)

* Corresponding author at: Department of Electrical and Computer Engineering (DEEC-UC), University of Coimbra, Pólo II, PT-3030-290 Coimbra, Portugal. Tel.: +351 96 1569024.
E-mail addresses: tmatias@isr.uc.pt (T. Matias), fasouza@isr.uc.pt (F. Souza), rui@isr.uc.pt (R. Araújo), ch@deec.uc.pt (C.H. Antunes).

0925-2312/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.neucom.2013.09.016

approach. In these algorithms, the evaluation of each individual or state is made by training the FFNN with a BP method, which is computationally expensive.

In [11] an improved GA is used to optimize the structure (connections layout) and the parameters (connection weights and biases) of a SLFN with switches. The switches are unit step functions that make possible the removal of each connection. Using a real encoding scheme, and new crossover and mutation techniques, this improved GA obtains better results in comparison with traditional GAs. The structure and the parameters of the same kind of SLFN with switches are also tuned in [12], in this case using a hybrid Taguchi GA. This approach is similar to a traditional GA but a Taguchi method [13] is used for the crossover process. The use of this method implies the construction of a $(n+1)\times n$ two-level orthogonal matrix, where n is the number of variables for the optimization process. However, the construction of this orthogonal matrix is not simple. There are some standard orthogonal matrices but they can be only used when n is small. In large networks, n is large and therefore this method is not a good practical approach. In these methodologies, the weights between the hidden-layer and the output layer are optimized by the GA. Using the ELM approach, the output weights could be calculated using the Moore–Penrose generalized inverse (considering an output neuron with linear activation function) and a good solution could be quickly obtained, reducing the convergence time of the GA. Furthermore, as the number of variables of the optimization process is lower, the search space to be explored by the GA narrows. This approach was used in [14] where a GA is used to tune the (selective existence of) connections and parameters between the input layer and the hidden layer, and a least squares algorithm is applied to tune the parameters between the hidden layer and the output layer. However, in this type of approach it is difficult to deal with the tendency to require more hidden nodes than conventional tuning-based algorithms, as well as the problem caused by the presence of irrelevant variables is difficult to solve. These problems occur also in the methods proposed in [15] and [16]. In [15] a learning method called evolutionary ELM (E-ELM) is proposed, where the weights between the input layer and the hidden layer and the bias of the hidden layer neurons are optimized by a DE method and the output weights are calculated using the Moore–Penrose generalized inverse like in ELM. In [16] a similar method called self-adaptive E-ELM (SaE-ELM) is proposed; however, in this methodology the generation strategies and control parameters of the DE method are self-adapted by the optimization method.

In this paper, a novel learning framework for SLFNs called optimized extreme learning machine (O-ELM) is proposed. This framework uses the same concept of the ELM where the output weights are obtained using least squares, however, with the difference that Tikhonov's regularization [17] is used in order to obtain a robust least squares solution. The problem of reduction of the ELM performance in the presence of irrelevant variables is well known, as well as its propensity for requiring more hidden nodes than conventional tuning-based learning algorithms. To solve these problems, the proposed framework uses an optimization method to select the set of input variables and the configuration of the hidden-layer. Furthermore, in order to optimize the fitting performance, the optimization method also selects the weights of connections between the input layer and the hidden-layer, the bias of neurons of the hidden-layer, and the regularization factor. Using this framework, no trial-and-error experiments are needed to search for the best SLFN structure. In this paper, three optimization methods (GA, SA, and DE) are tested in the proposed framework.

selection of the optimal number of neurons in this layer and the activation function of each neuron, trying to overcome the propensity of ELM for requiring more hidden nodes than conventional tuning-based learning algorithms. In this paper, three optimization methods (GA, SA, and DE) are tested in the proposed framework.

The paper is organized as follows. The SLFN architecture is overviewed in Section 2. Section 3 gives a brief review of the batch ELM. The proposed learning framework is presented in Section 4. Section 5 gives a brief review of the optimization methods tested in the O-ELM. Section 6 presents experimental results. Finally, concluding remarks are drawn in Section 7.

2. Adjustable single hidden-layer feedforward network architecture

The neural network considered in this paper is a SLFN with adjustable architecture as shown in Fig. 1, which can be mathematically represented by

$$y = g\left(b_O + \sum_{j=1}^{h} w_{jO} v_j\right), \tag{1}$$

$$v_j = f_j\left(b_j + \sum_{i=1}^{n} w_{ij} s_i x_i\right). \tag{2}$$

n and h are the number of input variables and the number of the hidden layer neurons, respectively; v_j is the output of the hidden-layer neuron j; x_i, $i=1,\ldots,n$, are the input variables; w_{ij} is the weight of the connection between the input variable i and the neuron j of the hidden layer; w_{jO} is the weight of the connection between neuron j of the hidden layer and the output neuron; b_j is the bias of the hidden layer neuron j, $j=1,\ldots,h$, and b_O is the bias of the output neuron; $f_j(\cdot)$ and $g(\cdot)$ represent the activation function of the neuron j of the hidden layer and the activation function of the output neuron, respectively. s_i is a binary variable used in the selection of the input variables during the design of the SLFN.

Using the binary variable s_i, $i=1,\ldots,n$, each input variable can be considered or not. However, the use of variables s_i is not the single tool to optimize the structure of the SLFN. The configuration of the hidden layer can be adjusted in order to minimize the output error of the model. The activation function $f_j(\cdot)$, $j=1,\ldots,h$, of each hidden node can be either zero, if this neuron is unnecessary, or any (predefined) activation function.

3. Extreme learning machine

The batch ELM was proposed in [6]. In [18] it is proved that a SLFN with randomly chosen weights between the input layer and the hidden layer and adequately chosen output weights are

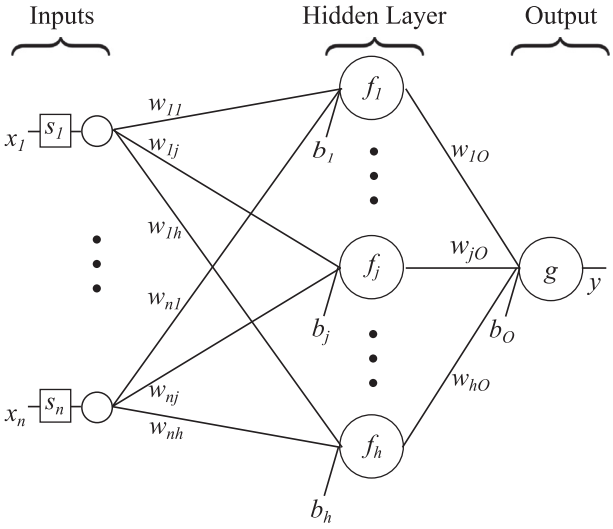

Fig. 1. Single hidden-layer feedforward network with adjustable architecture.