Computers and Mathematics with Applications 64 (2012) 128–136

Contents lists available at SciVerse ScienceDirect

Computers and Mathematics with Applications

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

Novel weighting in single hidden layer feedforward neural networks for data classification

Sattar Seifollahi [a,*], John Yearwood [a], Bahadorreza Ofoghi [b]

[a] Graduate School of Information Technology and Mathematical Sciences, University of Ballarat, PO Box 663, 3356, Australia

[b] Institute of Sport, Exercise, and Active Living, Victoria University, Melbourne, Victoria 3000, Australia

ARTICLE INFO

Article history:
Received 9 June 2011
Received in revised form 10 January 2012
Accepted 10 January 2012

Keywords:
Binary classification
Radial basis function network
Extreme learning machines
Influence weights
Attribute weighting

ABSTRACT

We propose a binary classifier based on the single hidden layer feedforward neural network (SLFN) using radial basis functions (RBFs) and sigmoid functions in the hidden layer. We use a modified attribute-class correlation measure to determine the weights of attributes in the networks. Moreover, we propose new weights called as *influence weights* to utilize in the weights connecting the input layer and the hidden layer nodes (hidden weights) of the network with sigmoid hidden nodes. These weights are calculated as the sum of conditional probabilities of attribute values given class labels. Our learning procedure of the networks is based on the extreme learning machines; in which the parameters of the hidden nodes are first calculated and then the weights connecting the hidden nodes and output nodes (output weights) are found. The results of the networks with the proposed weights on some benchmark data sets show improvements over those of the conventional networks.

1. Introduction

Over the past two decades, single hidden layer feedforward neural networks (SLFNs) have become an interesting topic of research. There are two main variations for SLFNs, those with additive hidden nodes and those with radial basis function (RBF) hidden nodes [1]. Using both nonlinear transfer functions provides the power of nonlinearity for the networks.

Radial basis function networks, first introduced in the neural network domain by Broomhead and Lowe [2], represent a specific class of SLFNs in which the linearly weighted structure of the networks allows for easy and fast training using linear optimization techniques. In such networks, the parameters in the hidden layer (known as hidden parameters) can often be pre-fixed. RBF networks have been proved capable of universal approximation [3,4]. Moreover, the total number of candidate basis functions involved in an RBF network model is not very large and does not increase when the number of input variables increases [5]. Due to these attractive properties, RBF networks have become a widely used network model in many areas such as function approximation [2,4] and classification and pattern recognition [6–9].

After selection of the number of the hidden nodes and basis functions in the RBF network, there are generally three types of parameters that need to be determined: (1) the position of the RBF centers, (2) the values of RBF widths, and (3) the connecting weights between the hidden layer and the output layer neurons (output weights). These parameters can be determined by performing either a combined procedure or separate procedures [5].

In combined approaches [5], all the three types of parameters are simultaneously determined by performing appropriate nonlinear optimization methods. Unlike combined learning approaches, the separate procedures include two phases of

* Corresponding author. Tel.: +61 421768635.
E-mail address: sattarseif@gmail.com (S. Seifollahi).

0898-1221/$ – see front matter © 2012 Elsevier Ltd. All rights reserved.
doi:10.1016/j.camwa.2012.01.042

learning [9–12]. In the first phase, the hidden parameters, including the centers and the widths, are determined, and in the next phase, the output weights are calculated.

For the learning of the output weights in an RBF network, as well as for the learning of an SLFN with sigmoid hidden nodes, error back propagation (EBP) is the most cited algorithm [11]. The main drawbacks of the EBP algorithm are its slow training and unreliability in its convergence due to weight initialization [13]. Although, the training time for an RBF network with the EBP learning algorithm is shorter than that for a multi-layer perceptron network, this time is still rather long and the efficiency of the algorithm depends on the choice of initial values [12].

Recently, Huang et al., [14], proposed a new fast algorithm for training SLFNs, known as the extreme learning machine (ELM), that can tackle some of the shortcomings of the EBP algorithm.

In the ELM algorithm, the weights and biases between the input and the hidden layer nodes are randomly assigned. The only unknown parameters that need to be determined are the output linear weights which are assigned through the least-square method [15,16].

The ELM is fast in classification tasks and also generates a high performance generalization compared with most of the existing methods such as backpropagation (BP) networks and support vector machine (SVM), reported in [14]. Moreover, the experimental results in [14] show that the standard deviations of the results obtained by the ELM algorithm are less than those of other methods. Here, our procedure for training the networks is the same as the theory of the ELM algorithm with a strong emphasis on attribute weighting and the hidden weights of the SLFNs.

The novelty of this work is two-fold. First, we use attribute weighting in the networks which uses an attribute-class correlation measure. The attribute-class correlation measure used in this work is a modification of the theory described in [17] which is originally used for attribute ranking.

Second, we introduce influence weights to utilize in the hidden weights of the SLFNs with sigmoid kernels in the hidden layer. The influence weights are calculated from the conditional probabilities of the attributes given class labels. These weights have a similar form to the influence weights proposed by Quinn et al. [18] which are directly used for data classification.

The paper is structured as follows. Sections 2 and 3 give a brief review on extreme learning machines and an RBF network, respectively. In Section 4, we propose influence weights and attribute weighting to improve the network performance which follows by the network learning procedure in Section 5. The experimental results and comparison of the proposed approaches with the existing methods, including the RBF network and the SLFN with sigmoid hidden nodes, are reported in Section 6. We conclude the paper in Section 7 followed by a few directions for future work.

2. Extreme learning machines

This brief outline of the ELM algorithm is based on that in [14]. In the ELM, the parameters of hidden nodes, such as weights and biases between the input and the hidden layer nodes, are randomly assigned. Therefore, the only parameters which need to be determined are the output weights.

Let us assume a data set $S = \{(\mathbf{x}_s, \mathbf{y}_s)|s = 1, \ldots, N\}$ of N arbitrary distinct samples, where $\mathbf{x}_s$ is an n-dimensional vector of decision-making attributes, $\mathbf{x}_s = [x_{s1}, x_{s2}, \ldots, x_{sn}]^T$, and $\mathbf{y}_s$ is the desired output corresponding to the input $\mathbf{x}_s$, $\mathbf{y}_s = [y_{s1}, y_{s2}, \ldots, y_{sm}]^T$, then the output of an SLFN, with $\widetilde{N}$ additive nodes, can be mathematically modeled as:

$$\sum_{j=1}^{\widetilde{N}} \alpha_j h_j(\mathbf{x}_s) + \alpha_0 = \sum_{j=1}^{\widetilde{N}} \alpha_j h(\mathbf{w}_j.\mathbf{x}_s + b_j) + \alpha_0 = \widehat{\mathbf{y}}_s \quad s = 1, \ldots, N \tag{1}$$

where $\mathbf{w}_j = [w_{1j}, w_{2j}, \ldots, w_{nj}]^T$ and b_j are the learning parameters of the j-th hidden node, $\alpha_j, j = 1, \ldots, \widetilde{N}$ is the weight vector connecting the j-th hidden node to the output nodes, α_0 is the bias vector to the output layer, h is the hidden node activation function and $\mathbf{w}_j.\mathbf{x}_s$ denotes the inner product of vectors $\mathbf{w}_j$ and $\mathbf{x}_s$ in $\mathbf{R}^n$.

If an SLFN with $\widetilde{N}$ hidden nodes can approximate these N samples with zero error, meaning that $\sum_{s=1}^{N} \|\mathbf{y}_s - \widehat{\mathbf{y}}_s\| = 0$, it then implies that there exist α_j, $\mathbf{w}_j$ and b_j such that:

$$\mathbf{H}\alpha = \mathbf{Y} \tag{2}$$

where

$$\mathbf{H} = \begin{bmatrix} 1 & h_1(\mathbf{x}_1) & \cdots & h_{\widetilde{N}}(\mathbf{x}_1) \\ \vdots & \vdots & \ldots & \vdots \\ 1 & h_1(\mathbf{x}_N) & \cdots & h_{\widetilde{N}}(\mathbf{x}_N) \end{bmatrix},$$

$$\alpha = [\alpha_0, \alpha_1, \ldots, \alpha_{\widetilde{N}}]^T$$

and

$$\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_N]^T.$$