Applied Soft Computing 13 (2013) 3831–3838

Contents lists available at SciVerse ScienceDirect

Applied Soft Computing

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

Improvement of RBF neural networks using Fuzzy-OSD algorithm in an online radar pulse classification system

Gholam Ali Montazer*, Hessam Khoshniat, Vahid Fathi

Information Technology Engineering Department, School of Engineering, Tarbiat Modares University, P.O. Box 14115-179, Tehran, Iran

ARTICLE INFO

Article history:
Received 27 February 2012
Received in revised form 16 October 2012
Accepted 29 April 2013
Available online 30 May 2013

Keywords:
Fuzzy C-Means
Optimum Steepest Descent (OSD)
Radial basis function (RBF) neural network
Three-Phase OSD
Fuzzy-OSD

ABSTRACT

In this paper a new methodology for training radial basis function (RBF) neural networks is introduced and examined. This novel approach, called Fuzzy-OSD, could be used in applications, which need real-time capabilities for retraining neural networks. The proposed method uses fuzzy clustering in order to improve the functionality of the Optimum Steepest Descent (OSD) learning algorithm. This improvement is due to initialization of RBF units more precisely using fuzzy C-Means clustering algorithm that results in producing better and the same network response in different retraining attempts. In addition, adjusting RBF units in the network with great accuracy will result in better performance in fewer train iterations, which is essential when fast retraining of the network is needed, especially in the real-time systems. We employed this new method in an online radar pulse classification system, which needs quick retraining of the network once new unseen emitters detected. Having compared result of applying the new algorithm and Three-Phase OSD method to benchmark problems from Proben1 database and also using them in our system, we achieved improvement in the results as presented in this paper.

1. Introduction

Radial basis function (RBF) neural networks introduced into the literature by Broomhead and Lowe in 1988 [1]. These feed-forward networks, which are trained using a supervised algorithm, have been extensively used for interpolation regression and classification [2,3]. This popularity is due to some advantages comparing to other neural network, such as better approximation characteristics, simpler network architecture and faster training procedures. For this reason, they have been widely used in interpolation regression and pattern classification problems and researchers have kept on working on improving performance of learning algorithms [4].

These networks follow a different approach in designing a supervised neural network from other neural networks. In spite of the back propagation (BP) algorithm or the design of a multi-layer perceptron (MLP) which may be similar to a stochastic approximation method [5], design of the network in RBF neural networks could be referred as a curve-fitting (approximation) problem in a space with higher dimensionality than the input space [6].

RBF neural networks have a three-layer feed forward architecture with a single hidden layer of units. The first layer, consisting of n input units, connects the input space to the environments. A hidden layer, which consists of N_h RBF units with a basis function as the activation function, transforms the input space to the hidden space which is of higher dimensionality than the input layer, and the output layer of m linear units produces the final classification or output to the input pattern. Each hidden unit estimates similarity between the input pattern and its connection weights or centers, locally. These networks implement the mapping $f{:}R^n \to R^m$ that:

$$Y=(y_1,\ \dots,y_s,\ \dots,y_m) : R^n \to R^m : y_s(X)=\sum_{j=1}^{N_h} w_{js}\varphi\left(\frac{\left\|X-C_j\right\|}{\sigma_j}\right) \tag{1}$$

where $X \in R^n$ is an input pattern, y_s is sth network output, $w_{js} \in R$ refers to the weight of the link between jth hidden neuron and sth neuron in the output layer, and finally C_j and σ_j are the center and width of the jth RBF unit in the hidden layer, respectively. Also the term φ denotes an activation function such as Gaussian function, defined by the equation [7]:

$$\varphi(r) = e^{-r^2} \tag{2}$$

Besides being efficient function approximator, RBF networks are also capable to solve pattern classification task [8,9]. In such applications, RBF neural network should map continuous input space into a set of classes by assigning the input pattern to the class of the output unit with the maximum value. Throughout this mapping process, the hidden layer performs a nonlinear transformation on the input patterns into a set of corresponding patterns in

* Corresponding author. Tel.: +98 21 82883990; fax: +98 21 82883990.
E-mail addresses: montazer@modares.ac.ir (G.A. Montazer), h.khoshniat@modares.ac.ir (H. Khoshniat), v.fathi@modares.ac.ir (V. Fathi).

1568-4946/$ – see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.asoc.2013.04.021

high-dimensional space whereas the output layer does a linear one. The underlying principle of these transformations is Cover's theorem on separability of patterns. According to this principle, patterns mapped into a space with higher dimensionality than the input space are more likely to be classified linearly [9]. As a result, patterns which are the result of the hidden layer are more probable to be linearly separable.

The learning process of RBF neural network, in the most applications, consists of two steps [10] which are an unsupervised learning stage for adjusting the parameters of the hidden layer including RBF centers $C_j \in R^n$, $j = 1, \ldots, K$, and widths of the units $\sigma_j \in R$ and a supervised one for estimating weights for connectors of these units $w_{js} \in R$. Due to different applications of these networks, a large variety of learning algorithms, has been employed for training RBF networks, have been proposed by several researchers including randomly selection of fixed centers and supervised selection of centers [11], using a regularization approach to estimate [8], orthogonal least squares algorithm [12], employing Expectation–Maximization (EM) algorithm for optimizing the cluster center after an initial clustering phase [13], regression tree to find the centers and width of the RBF units [14], self-organized selection of centers containing *K*-mean clustering procedure and the self-organizing feature map clustering procedure [15], individual training of each hidden unit based on functional analysis [16] or initial selection of a large number of hidden units which is reduced as the algorithm proceeds [17]. Genetic algorithms are employed in [18] to select the structure of the network and the parameters simultaneously. Also the pseudo-inverse (minimum-norm) method [19], the Least-Mean-Square (LMS) method [20], the Steepest Decent (SD) method [21], and the Quick Propagation (QP) method [22] are used for calculating weights of the network.

In our previous work [22], we proposed an optimized version of Steepest Descent method, called Optimum Steepest Decent (OSD), used an optimum learning rate in each epoch of the training process. In addition to the higher speed of learning process, this method also attains an absolute stability in network response. Although this improvement produced a better final result, randomly selection of the centers and widths of the RBF units was deficient. In order to improve the performance of the OSD, we introduced Three-Phase learning method which optimized the functionality of OSD learning method by attaining greater precision in initializing center and width of RBF units [23]. This method used K-Means clustering algorithm to calculate center of RBF units in the first phase and then their widths are estimated using K-Nearest Neighbor (KNN) algorithm. Adjusting the weights between hidden and output layer is performed in the third phase via OSD method. Despite the fact that these improvements guarantee that the global minimum in the weight space will be attained, there is a shortcoming that Three-Phase learning method may result in sub-optimal solutions. Because of being sensitive to the initial choice of cluster centers and widths, the K-Means algorithm and K-Nearest Neighbor algorithm may get stuck in a local optimum solution. This may cause learning process take more epochs to achieve a desired response.

Some critical applications, like helicopter sound identification system [23] or our online radar pulse classifier, need to be assured that the same response would be produced in a reasonable time, if the training process is done on the same training set. In this paper we improve the functionality of the RBF neural network by a new learning method called Fuzzy-OSD. In this approach we have replaced the K-Means clustering algorithm with a fuzzy clustering algorithm which is not sensitive to the initial values of the cluster centers. In other words, whenever we retrain the network with parameters defined by the fuzzy clustering algorithm, we get the same result. This is highly desirable in applications where producing the same response on the same training set through the retraining process is essential, in addition to keeping performance and speed of learning at the same level.

The organization of the paper is as follows. Section 2 describes in depth how fuzzy clustering is going to be used in the unsupervised learning stage. The Fuzzy-OSD method is presented in Section 3. Section 4 presents the implementation of the new method on several benchmark problems. Also the performance of this method in comparison with previous ones is discussed in this section. Experimental results of employing the proposed method in our online radar pulse classification system and a comparison of the mentioned methods is presented in Section 5. The last section, Section 6, concludes the previously presented sections.

2. Fuzzy clustering

Cluster analysis is a technique for grouping a set of unlabeled patterns or objects into a set of clusters based on similarity or dissimilarity among them so that the similarity between patterns assigned to the same cluster are as much as possible while dissimilarity in objects from different clusters should as less. A clustering task aims to find previously hidden structure in the objects, assuming that a natural grouping exists in the data [24]. According to the way of assigning object to the clusters, partial assignment or full assignment, there are two main approaches in clustering techniques, hard (crisp) clustering and fuzzy clustering, which are discussed in the following sections.

2.1. Hard and fuzzy clustering

Some clustering techniques force the full assignment of the objects even if they are equally similar to two or more clusters. In other words, although an object may be in the same distance of two clusters, it should be assigned to one of them and this hard assignment does not reflect the uncertainty of membership of the objects in the clusters [25]. These methods are called crisp or hard clustering. In the classical clustering algorithm such as K-Means objects are divided into some partition so that every object should be assigned only to one of those pairwise disjoint partitions. A common objective function, which is going to be minimized in a crisp clustering procedure, defines as the sum of distances between objects and the cluster centers that is [26]:

$$J_H = \sum_{i=1}^{K} J_i = \sum_{i=1}^{K} \left(\sum_{\substack{k \\ u_k \in C_i}} \left\| u_k - c_i \right\|^2 \right) \tag{3}$$

where K is number of clusters, u_k refers to kth object of the ith cluster, C_i denotes the ith cluster and c_i refers to the centroid of it.

On the other hand, fuzzy cluster analysis provides uncertainty and ambiguity in assignment of members to the clusters which is gradual membership of objects to different clusters in [0, 1]. These membership degrees express how ambiguously or definitely an object should belong to a cluster. The first use of fuzzy set concept in clustering was proposed by Ruspini in 1969 [20]. He mentioned that points in the center of a cluster could have a degree equal to 1, while membership degree of the boundary points depends on their distance to the cluster centers. The much closer an object to a cluster center, the closer degree of its membership to that cluster will be to 1 [24].

In the field of fuzzy clustering two types of fuzzy cluster partitions, which are much richer means for representing cluster structure, have evolved: probabilistic and possibilitic [24]. The most widely used approach, the probabilistic fuzzy clustering, was firstly