Neurocomputing 73 (2010) 3012–3019

Contents lists available at ScienceDirect

Neurocomputing

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

A sequential learning algorithm for self-adaptive resource allocation network classifier

S. Suresh [a,*], Keming Dong [b], H.J. Kim [b]

[a] School of Computer Engineering, Nanyang Technological University, Singapore
[b] CIST, Korea University, Seoul, Republic of Korea

ARTICLE INFO

Article history:
Received 12 August 2009
Received in revised form
14 July 2010
Accepted 14 July 2010
Communicated by F. Rossi
Available online 22 August 2010

Keywords:
Resource allocation network
Self-adaptive control parameters
Sequential learning
Multi-category classification
Extended Kalman filter

ABSTRACT

This paper addresses sequential learning algorithm for self-adaptive resource allocation network classifier. Our approach makes use of self-adaptive error based control parameters to alter the training data sequence, evolve the network architecture, and learn the network parameters. In addition, the algorithm removes the training samples which are similar to the stored knowledge in the network. Thereby, it avoids the over-training problem and reduces the training time significantly. Use of misclassification information and hinge loss error in growing/learning criterion helps in approximating the decision function accurately. The performance evaluation using balanced and imbalanced data sets shows that the proposed algorithm generates minimal network with lesser computation time to achieve higher classification performance.

1. Introduction

Neural networks are powerful tools that can capture the underlying relationship between the input and output data by learning. In the classical batch learning algorithm, the objective is to construct a network that can predict the output of new data given complete training data set. Here, the samples are presented simultaneously and presented as often as desired. In most practical applications complete set of training samples may not be available a priori or contain large training set. Some of the practical problems like cancer classification, human behavior prediction and scene understanding in video surveillance [22] and some industrial problems [1] allow temporal changes in the task being learnt. Hence, classical batch learning is experienced to be rather infeasible and online/sequential learning is employed instead.

In neural network model, selection of network architecture is a critical issue. One has to find a minimal architecture that accurately fits the true function described by the training data. A large network may accurately fit the training data, but may have poor generalization performance due to over-fitting. On the other hand, small network requires lesser computational effort, but may not be able to approximate the given function. Research in architecture selection has resulted in many algorithms to solve this problem. A complete literature on architecture selection is given in [16]. Finding appropriate architecture for a given training data set itself is a challenging problem. When the training data set is not defined prior to the learning process, the complexity of finding the minimal architecture increases further. In this paper, we address learning algorithms which evolve the architecture by itself in online/sequential learning framework.

Online/sequential learning is performed in sequence of trails. Here, the training samples arrive one-by-one and the samples are discarded after learning. So, it can accommodate the temporal changes in the task and require less memory and computational time for learning process. A sequential learning has been analyzed extensively in the framework of radial basis function network (RBFN) [5–15]. One of the first sequential learning algorithm was the resource allocation network (RAN) [11]. The RAN starts with zero hidden neuron and add neurons based on the novelty of the incoming data [7,11]. It uses the least mean square algorithm for network parameter update. Similar approach is used in the minimal resource allocation network (MRAN) [13] and the extended minimal resource allocation network (EMRAN) [12]. Besides adding neuron, the MRAN and EMRAN prune insignificant neurons from the network based on the contribution of error over a window of samples. In the MRAN, extended Kalman filter (EKF) was used to update the network parameters. Since the EKF is computationally intensive, the EMRAN updates the parameter of nearest neuron for the current sample. In [14,15], a single stage

* Corresponding author.
E-mail address: ssundaram@ntu.edu.sg (S. Suresh).

0925-2312/$ - see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.neucom.2010.07.003

incremental learning approach is presented. Here, the neurons are added to the network based on the local error history in a fixed interval which may leads to unnecessary or untimely insertions. In the RAN, MRAN and EMRAN algorithms add neuron continuously based on the error history and hence, the network generated by these algorithms are better than the incremental learning approach. In [6], neuron significance with respect to input distribution is used as a criterion for growing and pruning the network architecture, and is called growing and pruning RBFN (GAP-RBFN). Aforementioned algorithms learn samples one-by-one and only once and also evolve the network architecture automatically. It has been shown in literature that the aforementioned algorithms provide better generalization performance for function approximation problems than the classification problems [5].

For classification problems, the generalization performance depends heavily on the optimal selection of control parameters of the learning algorithms and the input data distribution as they use the basic function approximation approach for classification. Recently in [5], a sequential learning multi-category classifier is proposed. Here, the neurons are added based on misclassification information and distance between the current sample and nearest neuron in the same class. It is also shown that the use of hinge loss function improves the performance significantly [5].

For a fixed single-layer neural network architecture, online/sequential learning algorithm using recursive least square is presented in [8]. It is referred as an online sequential extreme learning machine (OS-ELM). In the OS-ELM algorithm, the input weights are selected randomly and output weights are calculated analytically using the least square error. For sequential learning, the output weights are updated using recursive least. Here, the training data was presented one-by-one or chunk-by-chunk. Recently in [10], incremental convex extreme learning machine was proposed. Both the OS-ELM and the incremental version produces better generalization performance with smaller computational time. In case of sparse and imbalance data sets, the random selection in the OS-ELM and incremental versions affects the performance significantly [19,23]. Similar to that of sequential learning neural network, in support vector machine framework, an incremental and decremental learning algorithm handles the training samples one-by-one or chunk-by-chunk [20]. Here, the solution is constructed recursively by retaining the Kucnh–Tucker (KT) conditions on all previously seen data, while adding a new data to the solution.

Aforementioned sequential learning algorithms (both constructive and fixed architecture networks) uses all training samples one-by-one and only once. If the training data set contains more similar data, then the resultant classifier has poor generalization due to over-fitting. In addition, the sequence in which the training samples presented to the sequential algorithms affects the performance significantly. Also, the problem dependent algorithm control parameters influence the approximation ability of the network. As the existing sequential algorithms suffer from these drawbacks, we need an algorithm which alters the sequence of training samples based on the information content to achieve a good generalization performance. We propose one such sequential learning algorithm, named as, self-adaptive resource allocation network (SRAN). The SRAN classifier uses radial function network as a basic building block. The control parameters in the proposed sequential algorithm are self-regulated, so, they are fixed, and are mostly independent of the problem considered. The control parameters alter the sequence in which the SRAN classifier approximates the decision function, based on the difference between the information contained in each sample and the knowledge acquired by the network. The higher the difference, the earlier a sample participates in learning. A few samples with lesser differences are pushed to the rear end of the sample data stack. These samples are later used to fine-tune the network parameters. Also, a few samples with redundant information are discarded from the training data set, thus avoiding over-training. Thus, the finally realized network is compact and provides better generalization performance.

The performance of the proposed SRAN classifier is evaluated by comparing it with other sequential learning neural algorithms like MRAN [13], EMRAN [12], GAP-RBF [6], SMC-RBF [5], fixed network online learning OS-ELM [8], and incremental and decremental SVM [20]. We also compare the results with batch learning SVM [4] classifier. For experimental evaluation, we consider: (i) image segmentation (IS), (ii) vehicle classification (VC), and (iii) glass identification (GI) problems from UCI machine learning repository [2]. Among the three real-world examples, GI and VC problems are sparse in nature with high sample imbalance. First, we use balanced IS data set to highlight the advantages of the proposed algorithm. Next, we use high sample imbalance VC and GI data sets to show the effectiveness of the proposed algorithm. The results clearly highlight that the proposed SRAN classifier is compact and provides better generalization performance. Finally, we highlight some issues related to SRAN algorithm.

The paper is organized as follows: Section 2 describes the proposed SRAN classifier. Section 3 presents experimental results and performance comparison with other existing sequential learning algorithms. Section 4 summarize the main conclusions from this study.

2. A sequential learning algorithm for self-adaptive resource allocation network (SRAN) classifier

In this section, we describe the principles behind self-adaptive resource allocation network (SRAN) first and then provide the various steps involved in the algorithm and finally summaries the algorithm in a pseudo code form.

2.1. Problem definition

Online/sequential learning for a multi-category classification problem can be stated in the following manner. The observation data arrives one-by-one and one at a time. After learning, the sample is discarded from the sequence. Suppose we have the observation data $\{(\mathbf{x}_1,\mathbf{y}_1),(\mathbf{x}_2,\mathbf{y}_2),\ldots,(\mathbf{x}_t,\mathbf{y}_t),\ldots\}$, where $\mathbf{x}_t \in \Re^m$ is an m-dimensional features of observation t and $\mathbf{y}_t \in \Re^n$ is its coded class label. Here, n represents the total number of classes. For notational convenience, the subscript t is left out in all further discussion. If the feature observation $\mathbf{x}$ is assigned to the class label c, then cth element of $\mathbf{y}=[y_1,\ldots,y_c,\ldots,y_n]^T$ is 1 and other elements are -1.

$$y_j = \begin{cases} 1 & \text{if } j==c \\ -1 & \text{otherwise,} \end{cases} \quad j=1,2,\ldots,n \tag{1}$$

The observation data are random variables and the observation $\mathbf{x}$ provides some useful information on probability distribution over the observation data to predict the corresponding class label with certain accuracy. Hence, the classification problem is to predict the coded class label $\mathbf{y}$ of a new observation $\mathbf{x}$. This requires us to estimate a functional relationship between the coded class label and feature space from sequential training data. In the SRAN classifier, a radial basis function network is used as a building block. The SRAN network approximates the functional relationship between the feature space and the coded class label.