Expert Systems with Applications 42 (2015) 8064–8077

Contents lists available at ScienceDirect

Expert Systems with Applications

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

Optimum steepest descent higher level learning radial basis function network

Kirupa Ganapathy [a,*], V. Vaidehi [a], Jesintha B. Chandrasekar [b]

[a] Department of Electronics Engineering, Anna University, Chennai, Tamilnadu, India
[b] Department of Information Technology, Anna University, Chennai, Tamilnadu, India

ARTICLE INFO

Article history:
Available online 6 July 2015

Keywords:
Neural network
Radial basis function
Dynamic learning
Optimum steepest descent
Higher level components
Healthcare

ABSTRACT

Dynamically changing real world applications, demands for rapid and accurate machine learning algorithm. In neural network based machine learning algorithms, radial basis function (RBF) network is a simple supervised learning feed forward network. With its simplicity, this network is highly suitable to model and control the nonlinear systems. Existing RBF networks in literature are applied to static applications and also faces challenges such as increased model size, neuron removal, improper center selection etc leading to erroneous output. To overcome the challenges and handle complex real world problems, this paper proposes a new optimum steepest descent based higher level learning radial basis function network (OSDHL-RBFN). The proposed OSDHL-RBFN implements major components inspired from the human brain for efficient learning, adaptive structure and accurate classification. Higher level learning and thinking components of the proposed network are sample deletion, neuron addition, neuron migration, sample navigation and neuroplasticity. These components helps the classifier to think before learning the samples and regulates the learning strategy. The knowledge gained from the trained samples are used by the network to identify the incomplete sample, optimal center and bond strength of hidden & output neurons. Adaptive network structure is employed to minimize classification error. The proposed work also uses optimum steepest descent method for weight parameter update to minimize the sum square error. OSDHL-RBFN is tested and evaluated in both static and dynamic environments on nine benchmark classification (binary and multiclass) problems for balanced, unbalanced, small, large, low dimensional and high dimensional datasets. The overall and class wise efficiency of OSDHL-RBFN is improved when compared to other RBFN's in the literature. The performance results clearly show that the proposed OSDHL-RBFN reduces the architecture complexity and computation time compared to other RBFN's. Overall, the proposed OSDHL-RBFN is efficient and suitable for dynamic real world applications in terms of detection time and accuracy. As a case study, OSDHL-RBFN is implemented in real time remote health monitoring application for classifying the various abnormality levels in vital parameters.

1. Introduction

Artificial neural network (ANN) is a parallel computing system that consists of huge number of simple nodes (processors) interconnected by weighted links. Neural network is one of the best classifier that does efficient classification. Several researchers have used neural network for multiclass and binary classification (Buchtala, Klimek, & Sick, 2005; Yang, Zeng, Zhong, & Wu, 2013). Traditional neural networks are best suitable for linear and static applications (Huang, Saratchandran, & Sundararajan, 2005; Zhang, 2006). However, real world applications are dynamic in nature and the input samples of such systems are highly nonlinear. For example, applications such as healthcare (Kourek & Dogan, 2010) speech recognition and robotic control (Lian, 2014), it is hard to describe the input and output relationship of the training data a prior. Static applications perform batch processing to model a network. Standard modeling techniques work eventually more severe in dynamic problems. Therefore, traditional neural network are to be modified to develop a stable, efficient adaptive network for solving dynamic applications. Neural network modeling for dynamic problem is a difficult task as it is necessary to adjust the model behavior with the online data stream. Recently, there have been enormous works carried out to solve static and dynamic classification problems. Thirst still exists in search for a neural network with simple architecture, less memory usage, less number of

* Corresponding author.
E-mail addresses: kirupa1980@gmail.com (K. Ganapathy), vaidehi@annauniv.edu (V. Vaidehi), jesinthabala@gmail.com (J.B. Chandrasekar).

http://dx.doi.org/10.1016/j.eswa.2015.06.036
0957-4174/

neurons, accuracy and speed for handling dynamic problems. Radial basis function neural network (RBFN) is the popular choice to model and control the nonlinear systems. RBF (Yang, 2006) is a simple supervised learning feed forward network that avoids iterative training process and trains the data at one stage. Advantage of iterative less learning ability and simple design strongly recommends the use of RBF model in dynamic environments. This is a special kind of artificial neural network that uses radial basis function as the activation function. This network can be applied to various applications to solve control, curve fitting and classification problems (Oyang, Hwang, Ou, Chen, & Chen, 2005). Due to the RBFN's universal approximation, compact topology and fast learning this network is suitable for binary and multiclass classification. Though, RBFN has more advantages, it works more severe in non stationary system.

There are various parameters in RBFN that contributes to accuracy and speed (Han & Qiao, 2012) of the classifier. The parameters or factors that determine the speed and accuracy of the RBFN are number of hidden neurons, center value, width and the weight between the hidden and the output layer. Several techniques are applied to improve traditional RBFN learning mechanism in static and dynamic environments. To solve the issues, various static and dynamic RBFN methods have been experimented and tested (Karayiannis & Randolph-Gips, 2003). In recent years, investigation of RBFN in changing environment problems has become one of the most important issue for real time applications. Few techniques recently developed tries to improve the performance of RBF neural network and their issues are briefly discussed below.

Rubio-Solis and Panoutsos (2015) proposed a hybrid method which implements Interval Type-2 fuzzy logic, RBF neural network and back propagation algorithm principle. Initially, from the membership functions the fuzzy rules are generated and establishes rule base of the RBF neural network. Then the RBF parameters such as center and width are optimized using back propagation algorithm. The work concludes to be computationally efficient. However, integration of three algorithms is a tedious process and the weight update process in back propagation algorithm is computationally expensive. Lian, Hu, and Zak (submitted for publication) developed an adaptive structure RBF network with feedback controller for dynamic applications. This model implements two learning components such as neuron addition and neuron deletion. Due to removal or addition of neurons, the structure of the network is more prone to changes. Therefore, Lyapunov function is used for stability of the adaptive structure. But, in most of the real time applications, the removal of neuron is not advisable. Infrequent samples are considered to be highly important for detecting the abnormal events. So, the deletion of RBF unit is not advisable unless the pattern is redundant.

Lian (2014) proposed a radial basis-function neural network to regulate the parameters such as membership function and fuzzy rules of a self-organizing fuzzy controller. The Lyapunov stability theorem is used to prove the stability of the fuzzy controller. However, the work uses traditional RBF and concentrates more on adaptive fuzzy controller for robotic motion control. Also, this work is more application specific and involves complex mathematical operations. Regis (2014) developed a optimization technique in which a RBF network acts as a surrogate to assist evolutionary programming algorithms. In this work, for the given objective function and constraints, the RBF neural network selects the highly suitable off spring for the next generation. The selection of offspring using RBF is accurate. But, this algorithm is computationally expensive for higher number of simulations. Also, entire off springs are learnt by the neurons in RBFN which increases the network structure.

Yu, Reiner, Xie, Bartczak, and Wilamowski (2014) proposed a RBFN to reduce or correct the error for accurate classification. This algorithm adds one RBF unit on every iteration to adjust the weight for correcting the error. The method proves to be computationally effective and accurate with less number of neurons compared to the other similar algorithms in the literature. However, the work is an offline algorithm which works well for static applications without learning components as well as the center & width of the neurons are fixed. Since this algorithm is for offline applications, it compromises the training time. Also, the architecture becomes complex for real time applications as it involves RBF unit addition for every iteration. Alexandridis et al. (2014) proposed a RBF neural network for estimation of large earthquake event occurrence. Fuzzy technique is used to train the neural network. Traditional RBF employed in this work predicts the possible earthquake occurrence. This algorithm is proved to be efficient by employing fuzzy sets to remove the redundant or unwanted training data. But, the works well for less training data and more application specific. Also, it lacks global usage of this algorithm over various applications. The work will fail to work for dynamic environment.

Chen, Gong, and Hong (2013) proposed a online RBF modeling for dynamic systems. Fixed number of neurons are used in the hidden layer. For weight update, multi-innovation recursive least square algorithm is used. In this work, the worst node which do not contribute much for learning are replaced with the new one by using quantum particle swarm optimization technique. The performance of the model is validated with different error estimation technique and the number of nodes. But, the fixed number of nodes will cluster huge samples which may lead to misclassification when learning large dataset. Also, this work includes randomized parameter initialization in particle swarm optimization approach which will affect the performance of the system in dynamic situations. This paper also fails to give information on computational time though it employs two different algorithms. Babu and Suresh (2013) proposed a sequential learning RBF network for classification problems. This network employs think and learn concept using cognitive and meta cognitive components. Sample overlapping conditions and knowledge of the past samples in the form of pseudo samples are used for proper initialization of new hidden neurons. The optimal weight is identified initially for training the samples using projection based weight update technique. But, this network is suitable for static applications as most of the real time applications do not know the sequence of the input earlier. Also, the optimal weight is identified before training which will reduce the learning efficiency of the classifier when RBF units are added.

Alexandridis (2013) proposed a technique to determine the number of centers and locations using fuzzy means algorithm. Optimal fuzzy partition of the linguistic levels are selected using particle swarm optimization approach. The synaptic weights are calculated using linear regression method. This technique achieves higher prediction accuracy and reduced network size. But, implements various algorithms with more number of random parameter initialization. Therefore, this approach will fail to be accurate and faster for dynamic applications. Assaf, El Assad, Harkouss, and Zoaeter (2012) proposed a classification algorithm using K-means and rival penalized competitive algorithms. It implements online training mode for learning the parameters of a Bayesian RBF neural network. The centers of the hidden neurons are equal to the number of channel states. Weight and the spread of the hidden neurons are learnt using gradient descent algorithm. This method proves to be advantageous for online & offline training, stability and high speed of convergence. But, this method is computationally expensive as it incorporates various techniques to learn the samples.

Ko (2012) proposed a time-varying learning algorithm using particle swarm optimization approach to optimize the learning rates of the RBF network. Support vector regression method is adopted to determine the nodes, kernel parameters and weights