Dynamic Threshold Selection for Sequential Learning in Radial Basis Function Networks

W. S. Lim*, W. L. Yeoh
Multimedia University, Faculty of Engineering & Technology, 75450 Melaka, Malaysia.

* Corresponding author. Email: wslim@mmu.edu.my
Manuscript submitted July 29, 2015; accepted December 28, 2015.
doi: 10.17706/ijcce.2016.5.5.311-320

Abstract: For sequential learning in Radial Basis Function (RBF) Networks, there is a requirement for dynamic selection of threshold because a constant threshold is inadequate to accommodate functions of varying amplitudes. In this paper, a new criterion is defined for the dynamic selection of the Euclidean output deviation threshold. Its effect on the learning process experienced by RBF networks with regard to functions of variable amplitude is shown. This improved network can automatically select a suitable threshold for its own supervised learning depending on the objective parameters set to achieve certain accuracy level of the desired output. This paper also proposes further automation to neuron growing and pruning within Radial Basis Function (RBF) neural networks. The proposed dynamic threshold selection technique has shown significant improvement in achieving stable neuron growth rate in dealing with signal amplitude variation.

Key words: Dynamic threshold selection, radial basis function, neural network, sequential learning, nonlinear function approximation, minimal resource allocation network.

1. Introduction

Many techniques have been proposed to solve the threshold selection problem. One of the recent works presented in [1], a dynamic threshold selection method has been developed for multi-label newspaper topic identification. A technique based on general topic model normalization was introduced for determining a threshold defining the boundary between the “correct” and the “incorrect” topics of a newspaper article. The topic identification module uses the Naive Bayes classifier for the multiclass and multilabel classification problem and improved results were presented.

In [2], a dynamic threshold-based edge detection method was presented. In most object detection schemes, edges are the regions of interest and the ability to identify it allows the determination of the boundaries of objects which fall within an image. It is an important concept, both in the area of object recognition and motion tracking. The authors have presented an adaptive thresholding based edge-detection method using morphological operators. The results obtained using the dynamic multi-level thresholding technique show remarkable improvement in contrast to the bi-level thresholding used in most of the common edge extractors.

In the filming industry, many types of scene change detection schemes have been proposed in literature. Computational schemes define a similarity measure between two consecutive frames. When this measure reveals a big enough change, a scene change is declared. These schemes define a threshold, typically a fixed one. If the value of the measure exceeds the threshold, a scene change is detected. However, a fixed
threshold value cannot perform well for all videos mainly due to the diversity of their characteristics. The key problem is to obtain an optimal value for such fixed threshold. If it is set too high, there is high probability that some cuts remain undetected. If it is too low, the detection scheme produces false detections. In real world videos, both cases can occur simultaneously. A dynamic threshold selection technique introduced by Dimou et al. [3] has efficiently trace scene changes in video frames. The experimental results indicate very good performance with significant improvement in accuracy with minimum complexity.

In 2010, Suresh et al. has developed a sequential learning algorithm for self-adaptive resource allocation network classifier [4]. The algorithm utilizes self-adaptive error based control parameters to alter the training data sequence, evolve the network architecture, and learn the network parameters. Besides, it also removes the training samples which are similar to the stored knowledge in the network. With that, the over-training problem can be avoided and hence reduces the training time significantly. The researchers have used the misclassification information and hinge loss error in growing/learning criterion that helps to approximate the decision function accurately, and thereby, generating minimal network with lesser computation time.

In another work [5], an algorithm is designed to construct and train wavelet neural network based on the variation of the number of hidden neurons dynamically during the training process. The proposed method determines the optimal number of the hidden neurons and solves the optimization problem of wavelet neural network structure. The problem of finding a good neural model is then discussed through solutions offered by wavelet neural networks trained by conjugate gradient algorithm.

In [6], the processing of sonar signals has been carried out using a Minimal Resource Allocation Network (MRAN) in identification of commonly encountered features in indoor environments. The stability-plasticity behaviors of the network have been investigated. From previous observations, the experimental results show that MRAN possesses lower network complexity but experiences higher plasticity and, is unstable. A novel approach is proposed to solve these problems in MRAN and has also been experimentally proven that the network generalizes faster at lower number of neurons compared to the original MRAN.

In our paper, we have proposed a Radial Basis Function (RBF) network that is able to dynamically select suitable threshold values regardless of the specific features present in the function to be detected. The RBF network will select its thresholds based on statistical information collected from the input and targeted output vectors provided during supervised training. As the RBF network undergoes increasing amount of training patterns, statistical information based on input and targeted output vectors will converge towards what can be considered an equilibrium value. Thus, the thresholds which are being continuously derived from these statistical values can then be regarded as optimally selected thresholds.

2. Minimal Resource Allocation Network

To address the need for a simple and practical implementation of neural networks with optimized computation overhead, a branch of Radial Basis Function (RBF) based neural network utilizing sequential learning called the Minimal Resource Allocation Network (MRAN) has been developed by Lu et al. [7], [8]. The MRAN algorithm manages the size of the neural network by checking criterions for adding and pruning RBF hidden neurons as necessary, so that an optimized implementation of the RBF network can be attained as a result of supervised training.

Ever since the pruning algorithm in MRAN was introduced, the MRAN algorithm has been successfully applied and furthermore, improvised to attain optimal neuron configurations in the hidden layer as in EMRAN and HMRAN developed by Li et al. and Nishida et al. in [9], [10] respectively. By removing unnecessary neurons in the hidden layer of the network, minimal computation power is required in the
actual implementation of the system.

Improvements upon the original algorithm have been researched in the field of sequential learning RBF networks over the years. In 2004, Huang et al. derived an equation for the calculation of the “significance” of a hidden RBF neuron [11]. This derivation resulted in the introduction of a new algorithm for neuron pruning known as growing and pruning (GAP)-RBF.

Further success has been achieved in implementations of MRAN on communication channel equalization by Deng et al. [12] employing an extended learning algorithm known as Complex MRAN (CMRAN). MRAN has also been successfully applied to solve problems in areas of function approximation and time series prediction in [7], [8].

3. Problem Statement

Applications of growing and pruning RBF networks for nonlinear system identification have evolved over the years to provide better and faster learning capabilities with increasing automation of training. As reported in previous literatures, existing automation in RBF networks includes automatic growing and pruning of the number of artificial neurons within the hidden layer of the network. Automation as such allows neural networks of this type to adapt to the complexity of the nonlinear system to be identified. However, the training of such neural networks still require a finite amount of human guidance when it comes to the predetermination of thresholds as well as other learning parameters to facilitate optimal hidden neuron growth. Because the growing and pruning criterions ultimately governs the number of neurons in the hidden layer, it is apparent that the selection of thresholds related to these growing and pruning criterions constitutes a critical factor in computational complexity. In this paper, optimal learning parameters differ on a case to case basis will be proven, depending heavily on several of the system’s key characteristics.

The first modification to be presented is a modification to the growth criterion as presented by MRAN. This modification proposes a combination of the two criterion parameters, namely the error function and the distance from nearest neuron. In order to understand the reasoning behind the integration of these values, it is essential that the limitations of the existing criteria are brought to light.

The two important MRAN growth criteria described in [7], [8] are the error function criterion, and the distance from nearest neuron criterion, as presented in (1) and (2) respectively. They are of interest in this paper.

\[ \|e_i\| > \epsilon_{\text{min}} \quad (1) \]

\[ \|x_i - \mu_{nr}\| > \epsilon_i \quad (2) \]

The Euclidean output error threshold which is commonly denoted by \( \epsilon_{\text{min}} \), and the Euclidean distance threshold, denoted by \( \epsilon_i \) are of remarkable importance. The distance from nearest neuron criterion has been designed with the assumption that the problems posed to RBF networks seldom have outputs that show drastic change with respect to the input vector. Since this criterion has been developed based on such an assumption, it poses a limitation to the differentiability of two almost similar input vectors to the neural network.

It is clear that in such a circumstance where there are two almost similar input vectors that require significantly different output vectors, it is impossible for an RBF network with the existing growth criteria to correctly approximate the solution. This is because a new neuron is not allowed to be grown right beside an existing one. With such neuron configuration, calculation for a solution in that proximity can only satisfy
either one of the problems correctly.

To successfully differentiate between two almost similar input vectors without sacrificing the growth criterion’s capabilities to prevent unnecessary neuron growth, a new combined criterion has been developed. This new criterion combines the error function criterion together with the distance from the nearest neuron so that even if the distance from the nearest neuron is relatively low, a neuron can still be grown if a large enough error function is obtained.

The combined criterion is shown as below.

\[
\| e \times (x_i - \mu_i) \| > e_{\text{min}}
\]  

(3)

Since the intention of implementing such a combined criterion is to achieve mutual scaling of the error function and the distance of the nearest neuron, the choice of operation to combine has been narrowed down to multiplication. From the given combined criterion, it is seen that the values of the error function and the distance of the nearest neuron has been multiplied together and compared with the similar error threshold value. The threshold has been maintained with the assumption that the distance from nearest neuron in the calculation ranges from 0 to 1 only. If that is not the case, the threshold should be scaled up according to the increase in range of possible distances from the nearest neuron.

In response to this problem, an extension to the automation of RBF network implementations is required. This extension will need to bring about the capabilities of an artificial neural network that is able to observe, on its own, the characteristics of the sequence of input vectors, and how the target output vector of the system relates to the input vector. To achieve that, statistics of the training samples will be stored in order to help the RBF network select reasonable thresholds for the growing and pruning criterions. In the results section, the problem is illustrated by first simulating the training process of an RBF networks with a constant value of error threshold in the growth and pruning criterion onto functions of different amplitude. From there, it is shown that a good threshold is highly dependent on the function of a specific problem. Finally, an implementation of dynamic selection of threshold is simulated on a generated data set and compared with a RBF network using a constant value of threshold.

4. Dynamic Selection of Threshold

The implementation of a neural network is an extremely time consuming process. In specific, information on the observations need to be collected beforehand and analyzed to decide on functionally useful values of thresholds. Apart from threshold setting, the learning process also requires substantial amount of computational strength and therefore inevitably time consuming to calculate and adjust connection weights. Since the Extended Kalman Filter (EKF) is used to tweak neuron connection weights [7], computation time in these sequential learning RBF networks further increases.

However, it is important to note that only the selection of threshold values is not an automated process, but rather one that requires human intervention. In environments where the range of input variables is unknown, selection of threshold values is difficult because a mistake in selecting these threshold values will consequently affect the neuron growth and pruning criterions of the RBF network, as will be proven later in the results section.

In every implementation of a sequential learning RBF network, there is a need for the selection of multiple threshold values. These threshold values will consequently affect the neuron growing and pruning criterions of the RBF network. It is true that in 2004 Huang et al. has proposed the GAP-RBF in [11] which minimizes the amount of threshold selections required. However, this is clearly not adequate because even among the most reduced systems, selection of the Euclidean output error threshold, \( e_{\text{min}} \), and the Euclidean distance threshold, \( e_i \), is extremely crucial.
In this paper, the focus of discussion is on the Euclidean output deviation threshold, \( e_{\text{min}} \). This does not mean in any way that the concepts presented here will not be applicable to the Euclidean distance threshold as well as any other thresholds that might be required for neuron growing and pruning. However, further research and analysis will need to be done to justify the dynamic selection of those thresholds. Here, an equation to calculate the value of \( e_{\text{min}} \) is defined, which can effectively and efficiently grow and prune hidden layer neurons in an RBF network.

\[
e_{\text{min}}(n) = e_{\text{min}}(n-1) - \left\{\left[e_{\text{min}}(n-1) - \epsilon_{\text{rms}}\right] \times r\right\}
\]

(4)

From the observation of (4), the value of \( e_{\text{min}} \) is dependent on multiple factors. In this equation, the value of \( e_{\text{min}} \) from the previous learning iteration is used as a reference, and adjustments to this value is done by comparing itself with the value of the RMS error, \( \epsilon_{\text{rms}} \). The difference in these values is multiplied by the learning rate, \( r \), to obtain the change in threshold value from the previous iteration to the current one. A high value of the learning rate will result in faster learning convergence while a low learning rate will provide a more consistent output. Clearly, the evaluation of such an equation is not possible for the first iteration of the sequential learning algorithm. Thus, care needs to be taken during implementation to modify the equation according to specific requirements. The equation shown here has been simplified for the sake of clarity.

5. Simulation Results and Discussion

In this section, it is first shown that the case of a single threshold value for an RBF network cannot be applied to all functions with equal effectiveness. In order to prove this, a MATLAB simulation was carried out to show how an RBF network with a constant threshold fails to adapt to similar functions of logarithmically varied amplitudes. After such a requirement for threshold selection is established, the proving of how a statistically driven system is able to provide automation for the dynamic selection of a threshold is shown, particularly the Euclidean error threshold. As an example, an arbitrary input function which is sinusoidal in nature was considered, as expressed in (5).

\[
Y = 0.5 A \sin (12 X_1) + 0.5 A \cos (13 X_2)
\]

(5)

A random set of data has been generated for the function described above, and provided to the constant threshold RBF networks sequentially. For the sake of clarity, Fig. 1 below shows the function to be approximated in a three dimensional graph, with amplitude, \( A = 1 \). Fig. 2 shows the response of the constant threshold RBF network with respect to the corresponding graph in Fig. 1. Then, Fig. 3 and Fig. 4 show response of the constant threshold RBF network with similar functions of amplitude \( A = 10 \), and \( A = 100 \).

![Fig. 1. Sinusoidal function of amplitude 1 to be approximated by the RBF network.](image-url)
From the simulations done on the sinusoidal functions of different amplitude, the resulting hidden neuron count after 1000 iterations of the constant threshold RBF network is shown in Table 1.

<table>
<thead>
<tr>
<th>Sinusoidal Function Amplitude, $A$</th>
<th>1</th>
<th>10</th>
<th>100</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Hidden Neurons Developed</td>
<td>61</td>
<td>330</td>
<td>711</td>
</tr>
</tbody>
</table>

From Table 1, it is obviously observed that a significant increase in the resulting number of hidden neurons with respect to deviation in amplitude. Undoubtedly, it is proven that a characteristic of the function to be approximated, specifically the amplitude, is an important factor to threshold selection.
Since the need for dynamic selection of thresholds has been justified, next is to test the potential of such automation to the sequential learning of such RBF networks. To do that, the new algorithm using the same randomly selected data set with three different amplitude levels as described above is tested. Since the data used in this test is completely identical to the one used before, all irregularities within the data set will be reproduced, therefore eliminating the possibility of data inconsistency.

The figures appended below shows the outcome of the growing and pruning RBF networks after a sequential learning of 1000 iterations. Fig. 5 shows the resulting approximation of the sinusoidal function with amplitude $A = 1$, while Fig. 6 shows the approximation for a similar function with amplitude $A = 10$. Finally, Fig. 7 shows the approximation for $A = 100$.

![Fig. 5. Mapped resulting approximation of the sinusoidal function with amplitude 1 using dynamically selected threshold levels.](image1)

![Fig. 6. Mapped resulting approximation of the sinusoidal function with amplitude 10 using dynamically selected threshold levels.](image2)

![Fig. 7. Mapped resulting approximation of the sinusoidal function with amplitude 100 using dynamically selected threshold levels.](image3)
It can be observed from Table 2, for all of these successful function approximations, the number of hidden neurons by the end of the simulation does not deviate far and stays in the range of 30 to 40 neurons. In other words, for a system with dynamically selected thresholds, the final complexity of the RBF network as well as its ability to converge is independent of the function amplitude. Thus, it is proven that a dynamic selection of threshold values is fundamentally sound solution to fast selection of these threshold values, as well as a potential reduction in implementation cost of an RBF network.

<table>
<thead>
<tr>
<th>Sinusoidal Function Amplitude, A</th>
<th>1</th>
<th>10</th>
<th>100</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Hidden Neurons Developed</td>
<td>30</td>
<td>31</td>
<td>36</td>
</tr>
</tbody>
</table>

To provide a clearer picture on how the dynamic selection of thresholds operates to provide optimal learning, observation is carried out on the error function and thresholds selected throughout the entire RBF network learning iterations.

As shown in the Fig. 8-Fig. 10, the minimum error threshold (indicated by solid line), $e_{\text{min}}$, changes over the span of 1000 learning iterations, effectively choosing only the critical error points to grow additional neurons. By employing such neuron growing procedures, overpopulation of neurons is minimized.
6. Conclusion

Motivated by the possibility of developing sequential RBF neural networks into completely automated systems capable of system identification, a technique for dynamic selection of thresholds has been proposed where an equation is defined for the dynamic selection of the Euclidean output deviation threshold. Its effect on the learning process experienced by RBF networks with regard to functions of variable amplitude is shown and subsequently proven to be effective in determining neuron growth.

References


using minimal hyper basis function networks. *Proceedings of SICE Conference* (pp. 2610-2615).


**Way-Soong Lim** received his Ph.D. degree in electronic engineering from Multimedia University of Malaysia, in 2007. He joined the university in 1999 as a lecturer in the Faculty of Engineering & Technology. Since 2008, he is serving the faculty as a deputy dean and later promoted to associate professor in 2010. His research interest covers areas in artificial intelligence, neural network, pattern recognition, conditions monitoring and robotics navigation. Along his career path, he has published numerous number of international journals and conference papers, book chapters and engineering text book.