View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by Publications at Bielefeld University

ESANN'2007 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 25-27 April 2007, d-side publi., ISBN 2-930307-07-2.

# Controlling complexity of RBF networks by similarity

Ralf Eickhoff[1] and Ulrich Rückert[2] *

1- Circuit Design and Network Theory
Technical University Dresden - Germany

2- Heinz Nixdorf Institute - System and Circuit Technology
University of Paderborn - Germany

**Abstract**. Using radial basis function networks for function approximation tasks suffers from unavailable knowledge about an adequate network size. In this work, a measuring technique is proposed which can control the model complexity and is based on the correlation coefficient between two basis functions. Simulation results show good performance and, therefore, this technique can be integrated in the RBF training procedure.

## 1 Introduction

Radial basis functions (RBF) are used for different purposes in science, e.g. classification or function approximation [1]. However, a difficult choice is to determine the number of functions used for superposition, and, therefore, several methods try to improve the model complexity considering the number of basis functions [2] or the input dimension [3]. The network size has to compromise the approximation quality, which usually improves as the network grows, and the training effort, which increases with the network size. Moreover, too complex models can show insufficient generalization properties requiring small networks [1]. Furthermore, in terms of hardware or software realization smaller networks occupy less area due to reduced memory needs. Hence, controlling the network size is one major task during training.

In this work, a method is presented which can control the model complexity of RBF networks. RBFs originate from a surface reconstruction where regularization theory is applied to solve this ill-posed problem. Thus, the network output can be described by a transfer function [4]

$$f_m(\vec{x}) = \sum_{i=1}^{m} \alpha_i \exp\left(\frac{-\|\vec{x} - \vec{c}_i\|^2}{2\sigma_i^2}\right) \tag{1}$$

where each individual Gaussian function obtains an output weight $\alpha_i$, a variance $\sigma_i^2$ and a center $\vec{c}_i$, and at all $m$ Gaussian functions are used for superposition. Besides the Gaussian function, other types of functions can be used with respect to the regularization parameter[4].

The remainder is organized as follows. In Section 2 the similarity measure is introduced, which is able to quantify the model complexity. Section 3 shows simulation results of the proposed method followed by conclusions in Section 4.

*This work was supported by the Graduate College 776 - Automatic Configuration in Open Systems - University of Paderborn, and funded by the German Research Foundation.

## 2 Equivalence measurement

Determining the equivalence between two different functions can help to improve the model complexity if functions obtaining nearly identical outputs are replaced by a new transfer function. Therefore, the amount of neurons in the network is decreased which also reduces the amount of free parameters offering a positive impact on the training algorithm. A reduced optimization space speeds up the convergence and lowers the computational costs of the optimization technique, e.g. gradient descent [1].

Moreover, the approximation quality can benefit from controlling the model complexity. If too many basis functions are utilized in the network non-available information can be extracted of the training data, which leads to an overfitted approximation scheme. This results into low generalization qualities of the neural network [1]. Consequently, the trade-off between approximation quality and network size have to be optimized.

Determining the similarity between two functions can help to decrease the model complexity. In communication theory, similarity is specified by the cross-correlation between two functions, where the cross-correlation is defined as [5]

$$\rho_{ij} = \frac{E\left\{g_i(\vec{x}) \cdot g_j(\vec{x})\right\}}{\sqrt{E\left\{g_i^2(\vec{x})\right\} \cdot E\left\{g_j^2(\vec{x})\right\}}} \tag{2}$$

where $g_i(\vec{x})$ denotes the corresponding function and $E$ the expected value.

Therefore, the cross-correlation of (2) between two Gaussian functions can be evaluated as

$$\rho_{ij} = \frac{\operatorname{sign}(\alpha_i)\operatorname{sign}(\alpha_j)\sqrt{2\sigma_i\sigma_j}\,\mathrm{e}^{-\frac{\|\vec{c}_i-\vec{c}_j\|^2}{2(\sigma_i^2+\sigma_j^2)}}}{\sqrt{\sigma_i^2+\sigma_j^2}} \tag{3}$$

where sign$(\cdot)$ denotes the sign function. Hence, (3) can be used to determine the similarity between two Gaussian functions and in the case of high correlation, both functions can be replaced by a new basis function. The parameters of the inserted function have to be determined which can be accomplished by an additional training step or by geometrical representations. The additional training will decrease the efficiency because this step has to be performed after a new function is inserted. Skipping this retraining will exclude too many functions from the optimization process because they have no initialized parameters and will lead to unnecessary large networks.

Determining the parameters of the inserted Gaussian function from geometrical representations does not require an additional training step, and this initial configuration can be used in the next optimization step. The new parameters of

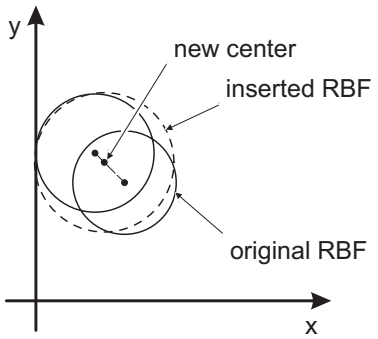


Fig. 1: Determination of the center and the variance of the new Gaussian function in the case of a two-dimensional input vector

the inserted Gaussian function can be determined as

$$\hat{\alpha} = \frac{\alpha_i + \alpha_j}{2} \tag{4}$$

$$\hat{\vec{c}} = \frac{1}{\alpha_i \sigma_i^n + \alpha_j \sigma_j^n} \left( \alpha_i \sigma_i^n \vec{c}_i + \alpha_j \sigma_j^n \vec{c}_j \right) \tag{5}$$

$$\hat{\sigma}^2 = \left( \frac{\sigma_i + \sigma_j}{2} + \frac{\min\left( \|\vec{m} - \vec{c}_i\|, \|\vec{m} - \vec{c}_j\| \right)}{2} \right)^2 \tag{6}$$

where $n$ denotes the input dimension of the network ($\dim(\vec{x}) = n$).

The new output weight is the mean of the original weights, whereas the new center of the inserted Gaussian function can be extracted from the geometric centroid of the volume occupied by the original basis functions. Further, the width of the inserted function in (6) is determined by geometrical representations shown in Figure 1.

The variance of the Gaussian function can be expressed as a circle[1] with radius $\sigma$ and center $c$, where the Gaussian function has the same output value because of its radial symmetry. If two basis functions obtain a high correlation coefficient, both circles overlap by a significant amount as in Figure 1. The new center can be found on the connection of the old centers whereas the new radius is a mixture of the mean of the old radii and the half of the minimum distance from one old center to its new one. As can be concluded from Fig. 1, for a high correlation the inserted Gaussian function covers nearly an identical area as the original *two* basis functions.

Therefore, the model simplification can be performed as follows. First, the correlation coefficient of all basis functions is determined and two Gaussian functions, which have the highest correlation, are replaced by a new function. The new function is initialized by the parameters of (4)–(6). These steps are repeated until the maximal correlation of (3) falls below a defined threshold.

Moreover, besides controlling the model complexity the correlation coefficient can further be used to identify overfitted models. In the extreme case,

[1] In in a higher dimensional case the variance can be referred to a hypersphere.

each Gaussian function represents only one test point and, therefore, the correlation between all functions tends to zero. Hence, the median of all correlation coefficients tends also to zero and can be further used as an indicator for overfitted models.

## 3 Simulation results

To show the practicability of the proposed method simulations are performed using popular test functions. For training the networks, a learning set is generated based on 1000 equally distributed samples drawn from the domain of each function. The network size[2] is set to 50 Gaussian functions and the initial configuration of the networks is determined by NETLAB. Then, the network size is minimized by applying the method of Section 2. As stopping criteria the threshold of correlation is set to the median of the distances between all original Gaussian functions. To evaluate the results an equally sized RBF network as reference is trained by NETLAB again.

Table 1 shows the results of the proposed method where the mean of 50 runs is presented. As approximation quality, the normalized mean squared error is determined by a test set consisting of 5000 data points and the resulting network sizes[3] are denoted by $m$. Due to the used NETLAB algorithm large output weights occur in the RBF network. In this case, the calculation of the new center according to (5) is not sufficient because large output weights can force the new center to lie outside the domain. This effect only occurs if the output weights have different signs and, therefore, two slightly different methods are performed. As can be seen from Fig. 1 and Eq. (3), the output weights are not necessary to determine the similarity since the center and the variance are already sufficient. Thus, for the first method only the activations of the neurons (setting $\alpha = 1$ for all neurons) are considered whereas for a second variant only positive correlation coefficients and, therefore, Gaussian functions with equally signed output weights are examined. To determine the output weights of the first method after minimizing the network size, an additional training is used. Here, all output weights are determined by solving the linear equality system, which arises from the activation and the target network output. To solve this system the least squares method is used utilizing the pseudo inverse.

Because only Gaussian functions obtaining positive correlation coefficients are considered in this variant II, at all less basis functions can be removed which results into larger networks sizes. Approximately, the original method produces half-sized networks. Both methods show similar approximation properties such as the reference network and compromising the accuracy and network size tradeoff. Although slight increases in the nmse can be observed, still good approximation properties can be guaranteed with a significantly reduced network size.

[2]The complexity of the networks is intentionally chosen as too high in order to apply the proposed method. Smaller network sizes are sufficient to achieve similar approximation properties.

[3]$m$ is the mean of 50 runs.

Table 1: Approximation properties of the original trained network and of a reduced network size determined by the similarity between Gaussian functions

| | **nmse RBF network** | | | |
|---|---|---|---|---|
| **function** | **NETLAB** | **reference** | **similarity** | **m** |
| Rechenberg | $1,06 \cdot 10^{-9}$ | $1,17 \cdot 10^{-8}$ | $4,33 \cdot 10^{-9}$ | 18,68 |
| (variant II) | $8,65 \cdot 10^{-10}$ | $3,00 \cdot 10^{-5}$ | $1,48 \cdot 10^{-5}$ | 9,12 |
| Rosenbrock | $5,60 \cdot 10^{-10}$ | $8,87 \cdot 10^{-6}$ | $3,00 \cdot 10^{-7}$ | 18,00 |
| (variant II) | $1,27 \cdot 10^{-10}$ | 0,0017 | 0,0007 | 8,88 |
| $(x_1 + x_2)^2$ | $8,45 \cdot 10^{-10}$ | $1,90 \cdot 10^{-7}$ | $1,17 \cdot 10^{-7}$ | 18,98 |
| (variant II) | $1,87 \cdot 10^{-9}$ | 0,0009 | 0,0003 | 9,56 |
| Schwefel | 0,0007 | 0,0015 | 0,0013 | 18,48 |
| (variant II) | 0,0008 | 0,0024 | 0,0026 | 8,82 |
| Griewank | 0,0069 | 0,0561 | 0,0535 | 18,70 |
| (variant II) | 0,0075 | 0,2551 | 0,2881 | 8,88 |
| Schaffer | 0,2285 | 0,2567 | 0,2565 | 17,96 |
| (variant II) | 0,2450 | 0,3052 | 0,3074 | 8,80 |

The results of variant II can be further improved if the same training as for the first technique is used after minimizing the network size.

Fig. 2 shows the overfitting properties with respect to the similarity of Gaussian functions. Here, the RBF network is forced to overfit the training data by choosing small variances approximating the test functions with the same network size ($m = 50$). After training, the median of the similarity between all Gaussian functions is determined whereas the overfitting property is evaluated by the quotient of the mse of the test data set to the mse of the learning data set. Therefore, large quotients represent an overfitted model. As can be concluded from Fig. 2, the similarity of Gaussian functions can be used to determine the model complexity with respect to overfitting properties. Here, the overfitting decreases as the median of the similarity increases for all test cases.

## 4 Conclusion

In this work, a method to control the model complexity of RBF networks is presented which is based on a similarity measure determined by the correlation of two Gaussian functions. This technique allows reducing the network size significantly with only slight degradation in the approximation qualities. Furthermore, the similarity measurement is able to qualify the model complexity with respect to overfitting capabilities and, therefore, this technique can be further considered as an additional objective in a multiobjective optimization process to balance network size, approximation quality and robustness of RBF networks [6].

To apply this technique knowledge about the internal parameters of each Gaussian function has to be provided in order to determine the correlation between two functions. Albeit, it is favorable to have no knowledge about the

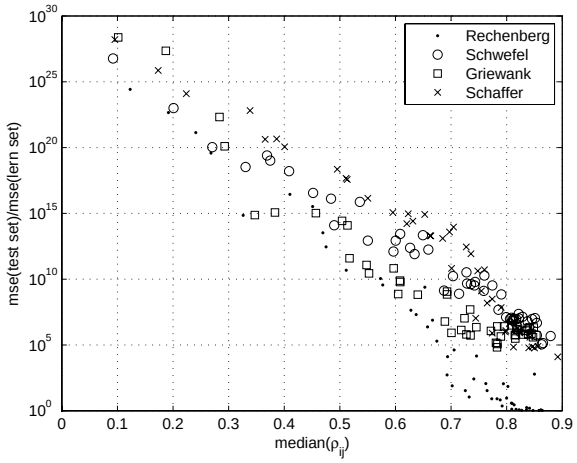


Fig. 2: Dependency between overfitting properties and the similarity of Gaussian functions

internal structure, the activation of each neuron is not sufficient to determine the similarity between two functions. Due to the radial symmetry of a Gaussian function, two activation responses are identical if two inputs have the same impact, which means this method is unable to identify differences between basis functions, which are reflected by the origin.

Nonetheless, the effectiveness of the proposed technique primarily depends on the defined threshold of the correlation, which is desired to be chosen automatically. As used in the simulations, the median of the distance between all original centers provides a good choice for this threshold. However, further investigations on the relationship between this threshold, the network size and the approximation quality respectively have to be performed.

## References

[1] Simon Haykin. *Neural Networks. A Comprehensive Foundation*. Prentice Hall, New Jersey, USA, second edition, 1999.

[2] J. Gonzalez, I. Rojas, J. Ortega, H. Pomares, F.J. Fernandez, and A.F. Diaz. Multiobjective evolutionary optimization of the size, shape, and position parameters of radial basis function networks for function approximation. *IEEE Trans. on Neural Netw.*, 14(6):1478–1495, 2003.

[3] D. François, V. Wertz, and M. Verleysen. The permutation test for feature selection by mutual information. In Michel Verleysen, editor, *Proc. of the 14th European Symposium on Artificial Neural Networks*, pages 239–244, Evere, Belgium, 26-28 April 2006.

[4] T. Poggio and F. Girosi. Networks for approximation and learning. *Proceedings of the IEEE*, 78(9):1481–1497, 1990.

[5] A. Papoulis. *The Fourier Integral and its Applications*. McGraw-Hill, 1962.

[6] Ralf Eickhoff and Ulrich Rückert. Pareto-optimal Noise and Approximation Properties of RBF Networks. In *Artificial Neural Networks – ICANN 2006*, volume 4131 of *LNCS*, pages 993–1002. Springer, 2006.