ESANN'1994 proceedings - European Symposium on Artificial Neural Networks
Brussels (Belgium), 20-21-22 April 1994, D-Facto public., ISBN 2-9600049-1-4, pp. 175-180

An optimized RBF network for approximation of functions

Michel Verleysen[1]* and Kateřina Hlaváčková[2]

[1] Université Catholique de Louvain, Laboratoire de Microélectronique - DICE,
3 Place du Levant, B-1348 Louvain-La-Neuve, Belgium
[2] Academy of Sciences of the Czech Republic, Institute of Computer Science,
Pod vodárenskou věží 2, 182 07 Prague 8, Czech Republic

Abstract. RBF networks are widely used for the non-parametric estimation of real-valued multi-dimentional functions through a finite set of samples. This paper describes a method to compute the parameters of a RBF network with Gaussian kernels, namely their locations, widths and weights; the parameters are first estimated through uncorrelated procedures to fit the inherent characteristics of the learning data set distribution, then are refined to minimize locally the global mean-square error of the network. As the simulation shows, this method can give better approximation results than conventional RBF procedures with no feedback between the computation of the three sets of parameters.

1 Introduction

Complex data processing is a challenge encountered in various domains of the engineering sciences. Classification, identification and approximation are closely related fields. In each a set of data is used to build non-parametric or parametric functions, which can then be used as a gerenalization on other data.

Radial basis functions have been widely used for such function interpolation [1]. The principle of radial basis functions networks is to fit a weighted sum of radial functions ϕ to the function f to approximate. Such functions ϕ depend only on the norm of the difference between their argument and a center, called *centroid*; they generally can be tuned by a *width factor*, or some kind of *variance*. After having determined the number of radial basis functions to use in a particular problem, the purpose of a RBF algorithm is to fix the locations of the kernels, their widths and the weighting factors.

Several algorithms and heuristics have been proposed to evaluate these parameters. In the algorithm by Moody and Darken [4], locations of centroids, widths and weights are sequentially determined, without feedback between these three computations. Locations of centroids are first fixed by an estimation of the probability density of the function, then widths are computed to ensure a defined "overlap" between radial basis functions centered on these centroids, and

Part of the work realized at UCL has been funded by the ESPRIT project 6891, ELENA-Nerves II, supported by the Commision of the European Communities.
* Senior Research Assistant of Belgian National Fund for Scientific Research (FNRS).

ESANN'1994 proceedings - European Symposium on Artificial Neural Networks
Brussels (Belgium), 20-21-22 April 1994, D-Facto public., ISBN 2-9600049-1-4, pp. 175-180

finally weights are computed to minimize a quadratic error function. Unfortunately, once the locations of the centroids are fixed , they are not allowed to be adjusted in order to globally minimize the quadratic error function. The same remark applies for widths.

We propose in this paper an algorithm to compute efficiently parameters in a RBF network similar to Moody and Darken's one. In our algorithm, however, width factors are determined with respect to the standard deviation inside the "region of influence" of a centroid, rather than by fixing an a priori overlapping factor between functions. Furthermore, values found for locations of centroids, width factor and weights are not definitive, but are optimized by a gradient descent algorithm to finally minimize the global mean-square error on the learning data set between the desired output values and the approximated ones.

2 RBF networks

The problem of interpolation of real multivariable functions can be expressed as follows. Let us consider a set of N data points in the input space $\mathcal{R}^d$, together with their associatied ***desired output*** values in $\mathcal{R}$:

$$\mathcal{D} = \left\{(\mathbf{x}_i, y_i) \in \Re^d \times \Re, 1 \leq i \leq N \mid f(\mathbf{x}_i) = y_i\right\}. \tag{1}$$

This data set can be used to characterize a function with one-dimensional output values; multi-dimensional interpolation can be done by generalizing the following equations and algorithms, while considering separately each component of the output vectors. We consider only one dimensional output in the following.

The RBF approach to approximate function f uses P functions $\phi_j(\mathbf{u}) = \phi_j(\|\mathbf{u} - \mathbf{c}_j)\|)$, where ϕ_j are radial functions $\phi_j : \Re^+ \rightarrow \Re, 1 \leq j \leq P, \mathbf{u} \in \Re^d, \mathbf{c}_j \in \Re^d$. The $\mathbf{c}_j$ are the locations of the centroids (the centers of the radial basis functions), while $\|\cdot\|$ denotes a norm on $\Re^n$ (usually Euclidean).

The approximation of function f may be expressed as a linear combination of the radial basis functions

$$\hat{f}(\mathbf{u}) = \sum_{j=1}^{P} \lambda_j \phi\left(\|\mathbf{u} - \mathbf{c}_j\|\right). \tag{2}$$

The most common radial function in practice is a Gaussian kernel given by

$$\phi\left(\|\mathbf{u} - \mathbf{c}_j\|\right) = e^{-\left(\frac{\|\mathbf{u}-\mathbf{c}_j\|}{\sigma_j}\right)^2}, \tag{3}$$

where σ_j is the width factor of kernel j.

Once the general shape of the ϕ_j functions is chosen, the purpose of a RBF algorithm is to find parameters $\mathbf{c}_j$, σ_j and λ_j to best fit function f. Fitting means here that the global mean-square error between the desired outputs y_i for all data point $\mathbf{x}_i, 1 \leq i \leq N$ and the estimated outputs $\hat{f}(\mathbf{x}_i)$ is minimized. This error is given by

$$E_{ms} = \frac{1}{2}\sum_{i=1}^{N}\left(y_i - \hat{f}(\mathbf{x}_i)\right)^2 = \frac{1}{2}\sum_{i=1}^{N}\left(f(\mathbf{x}_i) - \hat{f}(\mathbf{x}_i)\right)^2. \tag{4}$$

ESANN'1994 proceedings - European Symposium on Artificial Neural Networks
Brussels (Belgium), 20-21-22 April 1994, D-Facto public., ISBN 2-9600049-1-4, pp. 175-180

3 Training of RBF networks

As pointed out in the previous section, training a RBF networks means to find parameters $\mathbf{c}_j$, σ_j and λ_j used in equations 2 and 3 in order to get the best approximation $\hat{f}$ of the function f. In a conventional RBF training (see [4, 2]), these three sets of parameters are succesively and independently computed.

3.1 Locations of centroids

Locations of centroids are chosen according to the probability density of the input set $\mathbf{x}_i$. The reason for this is that one tries to best estimate function f in the regions where there are many data available, and to give less importance to the estimate in regions where there are only a few input points. From the point of view that the error which minimizes equation 4 is in fact a sum on all individual errors for all points in the data set, placing centroids according to the probability density of the input set seems reasonable. Moody and Darken [4] proposed to use a k-means clustering algorithm to find the locations of the centroids $\mathbf{c}_j$ minimizing the vector quantization error

$$E_{vq}(\mathbf{c}_1 \dots \mathbf{c}_P) = \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{P} \delta_{ij} \left\| \mathbf{x}_i - \mathbf{c}_j \right\|^2, \tag{5}$$

where δ_{ij} is the cluster membership function whose value is 1 if vector $\mathbf{x}_i$ falls inside the region of influence of $\mathbf{c}_j$, and 0 otherwise; a *region of influence* is defined as the set of points in the input space closer to one centroid than to another.

According to [2], we suggest using a LVQ1 iterative algorithm introduced in [3] to find the centroids $\mathbf{c}_j$ minimizing equation 5. This algorithm, used here for a one-class problem, has two parts. First, centroids $\mathbf{c}_j$ are randomly distributed over the input space, if possible in regions where one can find input patterns; one way to do this is to initialize the P centroids $\mathbf{c}_j$ to the first P input patterns $\mathbf{x}_i$, $1 \leq i \leq P$. Next, input vectors $\mathbf{x}_i$, $1 \leq i \leq N$ are sequentially or randomly presented to all centroids $\mathbf{c}_j$, and the centroid $\mathbf{c}$ closest to $\mathbf{x}_i$ according to

$$\|c_a - x_i\| \leq \|c_j - x_i\|, \forall j \in \{1 \dots P\}, 1 \leq a \leq P \tag{6}$$

is selected. Then, the selected centroid $\mathbf{c}_a$ is moved in the direction of input $\mathbf{x}_i$ according to the learning rule

$$\mathbf{c}_a(t+1) = \mathbf{c}_a(t) + \alpha(t) \left\| \mathbf{x}_j - \mathbf{c}_a(t) \right\|, \tag{7}$$

where $\alpha(t)$ is a time-decreasing adaptation factor, $0 < \alpha(t) < 1$. After convergence of this LVQ1 procedure, the probability density of the centroids will approximate the probability density of the input data, making the *regions of influence* of all centroids equiprobable. The reason why using a LVQ1 procedure will become more clear in the next section.

ESANN'1994 proceedings - European Symposium on Artificial Neural Networks
Brussels (Belgium), 20-21-22 April 1994, D-Facto public., ISBN 2-9600049-1-4, pp. 175-180

3.2 Width factors

A method to estimate the width factors σ_j of the Gaussian RBF units has been proposed by Moody and Darken [4]. They suggest to minimize an error function of the form

$$E_{ov}(\sigma_1 \ldots \sigma_P) = \frac{1}{2}\sum_{r=1}^{P}\left[\sum_{s=1}^{P} e^{-\left(\frac{\|\mathbf{c}_s-\mathbf{c}_r\|}{\sigma_r}\right)^2}\left(\frac{\|\mathbf{c}_s-\mathbf{c}_r\|}{\sigma_r}\right)^2 - Q\right]^2, \quad (8)$$

where Q is an overlapping factor a priori fixed to ensure overlap between kernels and thus a smooth estimator $\hat{f}$. Note that this only takes into account the locations of the centroids $\mathbf{c}_j$.

Setting an appropriate overlap parameter Q is however quite difficult. We suggest that the width of a kernel j be estimated by the mean value of its argument, i.e. of $\|\mathbf{x}_i - \mathbf{c}_j\|$. In fact, the width is taken to be twice this mean value to ensure a good overlap between kernels. Further, we propose to estimate it in parallel with the iteration of the LVQ1 algorithm used in the previous section to find the locations of the centroids. We thus have first to initialize all σ_j to 0, $1 \leq j \leq P$, and then to apply equation

$$\sigma_a(t+1) = (1-\beta(t))\sigma_a(t) + \beta(t)2\,\|\mathbf{x}_i - \mathbf{c}_a(t)\| \quad (9)$$

each time a vector $\mathbf{x}_i$ is presented and a centroid $\mathbf{c}_a$ is selected; $\beta(t)$ is a time-decreasing adaptation factor, $0 < \beta(t) < 1$, which can be chosen equal to $\alpha(t)$.

3.3 Weights

The last parameters to estimate in this RBF algorithm are the weights λ_j. As the final goal of the RBF network is to minimize the global mean-square error E_{ms} (equation 4) between the function f and the estimate $\hat{f}$ over the whole data set $\mathcal{D}$, the first methods chose the weights λ_j in order to minimize this error, all other parameters being fixed. We suggest to use the direct method presented in [2] to compute these weights. By setting the partial derivatives of the global error E_{ms} to zero, we obtain the weights $(1 \leq j \leq P)$

$$\lambda_j = \sum_{k=1}^{P}(\Phi^{-1})_{kj}\left[\sum_{i=1}^{N}\phi_k(\mathbf{x}_i)y_i\right], \; j = 1 \leq j \leq P, \quad (10)$$

where

$$(\Phi)_{kj} = \sum_{l=1}^{N}\phi_k(\mathbf{x}_l)\phi_j(\mathbf{x}_l), \quad (11)$$

and

$$\phi_k(\mathbf{x}_l) = \phi(\|\mathbf{x}_l - \mathbf{c}_k\|),\; 1 \leq k \leq P. \quad (12)$$

ESANN'1994 proceedings - European Symposium on Artificial Neural Networks
Brussels (Belgium), 20-21-22 April 1994, D-Facto public., ISBN 2-9600049-1-4, pp. 175-180

4 Optimization of the network parameters

We pointed out in a previous section that the locations of the centroids were chosen to approximate the probability density of the inputs, but that this choice need not lead to a minimization of the global error E_{ms} given by equation 4. The same comment applies to the proposed heuristic to set width factors σ_j. However, the selected locations $\mathbf{c}_j$ and widths σ_j reflect the inherent characteristics of the function f to approximate, and are thus good preliminary estimates of the optimal factors.

We propose here to use the values found with the above algorithms for parameters $\mathbf{c}_j$, σ_j and λ_j as a starting point for a minimization of the error E_{ms} given by equation 4, but now with respect to the three types of parameters together. Of course, it is no more possible to find an analytical solution, as in the estimation of the weights described in the previous paragraph. A gradient descent must be used, by iteratively adapting parameters $\mathbf{c}_j$, σ_j and λ_j in the opposite direction of the respective partial derivatives of the error:

$$\frac{\delta E_{ms}}{\delta c_{mn}} = -2\sum_{i=1}^{N}\left[\frac{\lambda_m}{\sigma_m^2}(x_{in}-c_{mn})\,e^{-\left(\frac{\|\mathbf{x}_i-\mathbf{c}_m\|}{\sigma_m}\right)^2}\left(y_i-\sum_{j=1}^{P}\lambda_j e^{-\left(\frac{\|\mathbf{x}_i-\mathbf{c}_j\|}{\sigma_j}\right)^2}\right)\right],$$

$$\frac{\delta E_{ms}}{\delta \sigma_l} = -2\sum_{i=1}^{N}\left[\lambda_l\frac{\|\mathbf{x}_i-\mathbf{c}_l\|^2}{\sigma_l^3}e^{-\left(\frac{\|\mathbf{x}_i-\mathbf{c}_l\|}{\sigma_l}\right)^2}\left(y_i-\sum_{j=1}^{P}\lambda_j e^{-\left(\frac{\|\mathbf{x}_i-\mathbf{c}_j\|}{\sigma_j}\right)^2}\right)\right],$$

$$\frac{\delta E_{ms}}{\delta \lambda_k} = -\sum_{i=1}^{N}\left[e^{-\left(\frac{\|\mathbf{x}_i-\mathbf{c}_k\|}{\sigma_k}\right)^2}\left(y_i-\sum_{j=1}^{P}\lambda_j e^{-\left(\frac{\|\mathbf{x}_i-\mathbf{c}_j\|}{\sigma_j}\right)^2}\right)\right], \qquad (13)$$

where c_{mn} is the n^{th} coordinate of centroid $\mathbf{c}_m$. By using a gradient descent on error E_{ms} with respect to the set of parameters $\{c_{mn}, \sigma_l, \lambda_k\}$, kernel locations, widths and weights are adapted to minimize error E_{ms} of the approximation $\hat{f}$, and thus to best fit the function f. Numerical experiments have shown that the initial parameters found via the procedures of this paper are usually in a basin of attraction associated with a local minimum that is reasonably close to the global minimum of the error function (4).

5 Simulations

Figure 5 shows an example of the simulation of the proposed algorithm, and comparison to the standard Moody and Darken's one. For the clarity of the illustration, the approximation of a 1-D function on the interval $[0, 1]$ is presented. Stars on the figure represent the data points. The probability density of the input data points (x-axis) is constant. Moody and Darken's method thus converged to centroids approximately located at $c_1 = 0.1, c_2 = 0.3, c_3 = 0.5, c_4 = 0.7, c_5 = 0.9$, which is the result of the convergence of a LVQ1 algorithm on a uniform distribution in range $[0, 1]$. These locations are efficiently adapted by our optimization

ESANN'1994 proceedings - European Symposium on Artificial Neural Networks
Brussels (Belgium), 20-21-22 April 1994, D-Facto public., ISBN 2-9600049-1-4, pp. 175-180

procedure to best fit function f. Figure 5 respectively shows interpolator $\hat{f}$ for the Moody and Darken's method, after initialization of our procedure, and after its optimization.

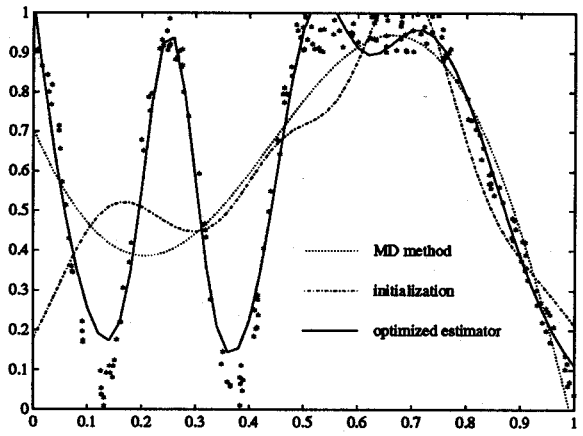

Fig. 1. Approximation of a 1-D function with noise

6 Conclusion

A method to approximate real-valued multivariable functions through radial-basis Gaussian kernels has been presented. It first estimates values for the locations, widths and weights of the kernels, and then optimizes these parameters to reach a minimum in the mean-square error. By comparison with standard methods where the three sets of parameters are independently computed, in simulations this method has shown to give a better estimate of the function to approximate.

References

1. D.S. Broomhead and D. Lowe. Multivariable functional interpolation and adaptive networks. *Complex Systems*, 2:321–355, 1988.
2. K. Hlaváčková and R. Neruda. Radial basis functions networks. *Neural Network World*, 1:93–101, 1993.
3. T. Kohonen. *Self-Organization and Associative Memory.* Springer-Verlag, 1989. 3rd Edition.
4. J. Moody and C. Darken. Learning with localized receptive fields. In D. Touretzky, G. Hinton, and T. Sejnowski, editors, *Proceedings of the 1988 Connectionist Models Summer School*, San Mateo, CA, 1989. Morgan Kaufmann.