PERGAMON

AVAILABLE AT
www.ComputerScienceWeb.com
POWERED BY SCIENCE DIRECT®

Neural Networks 16 (2003) 69–77

Neural Networks

www.elsevier.com/locate/neunet

A new EM-based training algorithm for RBF networks

Marcelino Lázaro, Ignacio Santamaría*, Carlos Pantaleón

Departamento Ingeniería de Comunicaciones, ETSIIT, Universidad de Cantabria, Av. Los Castros s/n, 39005 Santander, Spain

Received 27 July 2001; accepted 3 August 2002

Abstract

In this paper, we propose a new Expectation–Maximization (EM) algorithm which speeds up the training of feedforward networks with local activation functions such as the Radial Basis Function (RBF) network. In previously proposed approaches, at each E-step the residual is decomposed equally among the units or proportionally to the weights of the output layer. However, these approaches tend to slow down the training of networks with local activation units. To overcome this drawback in this paper we use a new E-step which applies a soft decomposition of the residual among the units. In particular, the decoupling variables are estimated as the posterior probability of a component given an input–output pattern. This adaptive decomposition takes into account the local nature of the activation function and, by allowing the RBF units to focus on different subregions of the input space, the convergence is improved. The proposed EM training algorithm has been applied to the nonlinear modeling of a MESFET transistor.

Keywords: Radial basis functions; Generalized radial basis functions; Expectation–maximization; Training; MESFET; Intermodulation

1. Introduction

Radial Basis Function (RBF) networks have become one of the most popular feedforward neural networks with applications in regression, classification and function approximation problems (Bishop, 1997; Haykin, 1994). The RBF network approximates nonlinear mappings by weighted sums of Gaussian kernels. Therefore, an RBF learning algorithm must estimate the centers of the units, their variances and the weights of the output layer. Typically, the learning process is separated into two steps: first, a nonlinear optimization procedure to select the centers and the variances and, second, a linear optimization step to fix the output weights. To simplify the nonlinear optimization step, the variances are usually fixed in advance and the centers are selected at random (Broomhead & Lowe, 1988) or applying a clustering algorithm (Moody & Darken, 1989).

Other approaches try to solve the global nonlinear optimization problem using supervised (gradient-based) procedures to estimate the network parameters, which minimize the mean square error (MSE) between the desired output and the output of the network (Karayanis, 1997; Lowe, 1989; Santamaría et al., 1999). However, gradient descent techniques tend to be computationally complex and suffer from local minima.

As an alternative to global optimization procedures, a general and powerful method such as the Expectation–Maximization (EM) algorithm (Dempster, Laird, &, Rubin, 1977) can be applied to obtain maximum likelihood (ML) estimates of the network parameters. In the neural networks literature, the EM algorithm has been applied in a number of problems: supervised/nonsupervised learning, classification/function approximation, etc. Here we concentrate on its application to supervised learning in function approximation problems. In this context, Jordan and Jacobs (1994) proposed to use the EM algorithm to train the mixture of experts architecture for regression problems. The EM algorithm has been also applied to estimate the input/output joint pdf, modeled through a Gaussian mixture model, and then estimating the regressor as the conditional pdf (Ghahramani & Jordan, 1994). In both cases the missing data select the most likely member of the mixture given the observations, and then each member is trained independently.

More recently, the EM algorithm has been applied for efficient training of feedforward and recurrent networks (Ma & Ji, 1998; Ma et al., 1997). The work in Ma et al. (1997) connects to the previous work of Feder and Weinstein (1988) for estimating superimposed signals in noise. In both

* Corresponding author. Tel.: +34-42-201552; fax: +34-42-201488.
E-mail address: nacho@gtas.dicom.unican.es (I. Santamaría).

0893-6080/03/$ - see front matter © 2002 Elsevier Science Ltd. All rights reserved.
PII: S0893-6080(02)00215-0

Nomenclature

N	total number of training samples
G	number of components (neurons) of the RBF
$\mathbf{y}$	vector of observations (incomplete data)
x_k	kth input
y_k	kth observation
e_k	error in the approximation of the kth observation
$\mathbf{z}$	vector of complete data
$z_{k,i}$	complete data for kth observation and ith component
$v_{k,i}$	missing data for kth observation and ith component
t_i	decoupling variable for ith component (conventional EM approaches)
$t_{k,i}$	decoupling variable for kth observation and ith component
$o_i(\cdot)$	activation function of the ith RBF unit
λ_i	weight of the ith RBF unit
$g_i(\cdot)$	output of the ith RBF unit ($g_i(\cdot) = \lambda_i o_i(\cdot)$)
$P(\cdot)$	a probabilitiy density function
$\mathbf{I}$	identity matrix
$E\{\cdot\}$	the expectation operation

methods, the E-step reduces to decompose at each iteration the total residual into G components (G being the number of neurons). In Feder and Weinstein (1988), the variables used to decompose the residual can be arbitrary values, as long as they sum one, but constant over the function domain: for instance, they propose to decompose the residual into G equal components. In Ma et al. (1997), the residual is decomposed proportionally to the weights of the output layer. Both approaches work well for feedforward networks with global activation functions such as the MLP, but tend to be rather slow for networks with local activation functions since each individual unit is forced to approximate regions far away from the domain of the activation unit.

To overcome this drawback we propose in this paper a new EM algorithm, specific for RBF networks, which aims to accelerate its convergence. We perform a soft decomposition of the residual, taking into account the locality of the basis functions. Different examples show that this modification speeds up the convergence in comparison with previous EM approaches.

The paper is organized as follows. In Section 2, the main features of the EM algorithm are presented. In Section 3, we present some EM-based approaches for the training of feedforward neural networks. In Section 4, the EM algorithm is applied to train an RBF network taking advantage of the local nature of its activation function. Simulation results are provided in Section 5 to validate the proposed algorithm. In Section 6, we apply this algorithm to the small-signal modeling of a MESFET transistor to reproduce the intermodulation behavior. To conclude the paper, in Section 7, the main conclusions are exposed.

2. The EM algorithm

The EM algorithm (Dempster et al., 1977) is a general method for ML estimation of parameters given incomplete data. The word incomplete indicates that, using this formulation, it is convenient to associate two sets of random variables with the problem, Y and V, only one of which, Y, is directly observable. However, the underlying model is expressed in terms of both Y and $Z = \{Y, V\}$. In the original formulation of the EM algorithm, Y was called the incomplete data, V the missing data, and the combined set Z was called the complete data. The goal of the algorithm is to find the set of parameters, say θ, which will maximize the likelihood of the observed values of Y. The maximization is carried out in terms of the joint distribution of Y and Z.

Let us consider the probability density of the incomplete data Y to be $f_Y(y;\theta)$; then, the probability density associated with the complete data Z can be expressed as follows

$$f_Z(z;\theta) = f_{Z|Y=y}(z;\theta) f_Y(y;\theta), \tag{1}$$

where $f_{Z|Y=y}(z;\theta)$ is the conditional probability density of Z given $Y = y$. Taking the logarithm on both sides of Eq. (1),

$$\log f_Y(y;\theta) = \log f_Z(z;\theta) - \log f_{Z|Y=y}(z;\theta), \tag{2}$$

and taking the conditional expectation given $Y = y$ with a parameter value $\theta = \theta'$, the following relationship is obtained

$$\log f_Y(y;\theta) = E\{\log f_Z(z;\theta)|Y = y;\theta'\} - E\{\log f_{Z|Y=y}(z;\theta)|Y = y;\theta'\}. \tag{3}$$

The main theoretical result associated with the EM algorithm is that to increase $\log f_Y(y;\theta)$ can be reduced to increase the first term of the right side in Eq. (3). This allows to increase the likelihood of the observed data by means of the following iterative algorithm

E-Step: To estimate

$$E\{\log f_Z(z;\theta)|Y = y;\theta^{(n)}\}. \tag{4}$$

M-Step: To evaluate

$$\theta^{(n+1)} = \arg\max_{\theta} E\{\log f_Z(z;\theta)|Y = y;\theta^{(n)}\}, \tag{5}$$

where $\theta^{(n)}$ is the set of parameters in the nth iteration. If Eq. (4) is a continuous function in both θ and $\theta^{(n)}$, the algorithm converges to a stationary point in the log-likelihood function (Wu, 1983), and the maximization in

Eq. (5) ensures that each iteration increases the likelihood. Of course, the convergence point may not be the global maximum of the likelihood function and, therefore, the experiment must be repeated from different initial conditions.

3. EM-based training of feedforward networks

In this section we introduce the notation and describe previous work on training two-layer feedforward networks using EM-based approaches. Without loss of generality, let us consider an RBF network with G Gaussian units, which approximates an one-dimensional mapping, $g(x): R \rightarrow R$, as follows

$$\tilde{g}(x) = \sum_{i=1}^{G} \lambda_i o_i(x), \tag{6}$$

where i indexes the RBF units, λ_i the amplitude, and $o_i(x)$ is the activation function of each unit, which is given by

$$o_i(x) = \exp - \left(\frac{(x - \mu_i)^2}{2\sigma_i^2} \right). \tag{7}$$

Our training problem consists in estimating the amplitudes, λ_i, centers, μ_i, and variances, σ_i^2, of an RBF model given a set of inputs and the corresponding noisy outputs, $\{x_k, y_k\}$. The noisy observations may be characterized using the following model

$$y_k = \sum_i g_i(x_k; \theta_i) + e_k, \tag{8}$$

where $g_i(x_k; \theta_i) = \lambda_i o_i(x_k)$ and, as usually, we assume that e_k is a zero-mean white Gaussian noise of variance σ^2. Then, the log-likelihood of the parameters is given by

$$L(\theta; \mathbf{x}, \mathbf{y}) = K - \frac{1}{2\sigma^2} \sum_k \left(y_k - \sum_i g_i(x_k; \theta_i) \right)^2, \tag{9}$$

where K is a constant which can be neglected in the optimization process, $\theta = \{\theta_1, \ldots, \theta_G\}$, and $\theta_i = \{\lambda_i, \mu_i, \sigma_i\}$.

From Eq. (9) we see that, under the Gaussian noise assumption, to obtain ML estimates reduces to minimize the conventional MSE. This multiparameter nonlinear optimization process can be accomplished through a global gradient descent algorithm (Karayanis, 1997; Lowe, 1989; Santamaría et al., 1999): its shortcomings have been already mentioned.

A computationally more efficient procedure to obtain ML estimates is based on the EM algorithm. A good choice for the missing or the complete data of this algorithm is necessary to simplify the maximization of the likelihood. A particularly useful selection for this problem was proposed in Feder and Weinstein (1988): the complete data is obtained by decomposing each observation into G signal components, according to

$$z_{k,i} = g_i(x_k; \theta_i) + e_{k,i} \qquad i = 1, \ldots, G, \tag{10}$$

where the residuals $e_{k,i}$ are also obtained by decomposing the total residual $e_k = y_k - \sum_i g_i(x_k; \theta_i)$ into G components, i.e.

$$e_{k,i} = t_i e_k \qquad i = 1, \ldots, G, \ \ \forall k. \tag{11}$$

In Feder and Weinstein (1988), it was shown that the decoupling variables t_i can be arbitrary constants, constrained to sum 1. A decomposition of the residual equally among all the neurons is then proposed

$$t_i = \frac{1}{G}, \qquad i = 1, \ldots, G. \tag{12}$$

Finally, using Eq. (12) to decompose the residual this EM algorithm, for training two-layer feedforward networks, can be summarized as follows

E-step: for $i = 1, \ldots, G$ compute

$$\hat{z}_{k,i} = g_i(x_k; \theta_i) + t_i \left(y_k - \sum_{j=1}^{G} g_j(x_k; \theta_j) \right). \tag{13}$$

M-Step: for $i = 1, \ldots, G$ evaluate

$$\{\lambda_i, \mu_i, \sigma_i^2\} = \arg \min_{\theta_i} \sum_k (\hat{z}_{k,i} - g_i(x_k; \theta_i))^2, \tag{14}$$

where the index denoting iteration has been omitted. Note that the problem of globally training an RBF network with G neurons has been decoupled into G simpler problems of training a single neuron.

Another EM approach to train feedforward neural networks has been proposed in Ma et al. (1997). In this case, the missing data, $v_{k,i}$, are the desired outputs of the neurons of the hidden layer. With this choice, it is necessary to select some probabilistic models to obtain the probability of the complete data given the observations and the parameters. Gaussian models have been selected for the conditional distribution of the missing data given the input, and for the conditional distribution of the output given the missing data. With these probabilistic models, it is shown that the E-step is reduced to obtain the expected value of the missing variables as

$$\hat{v}_{k,i} = o_i(x_k) + e_{k,i} \qquad i = 1, \ldots, G. \tag{15}$$

It is interesting to remark the difference with respect to Eq. (10). Now we use the outputs of the hidden layer, $o_i(x_k)$, instead of the outputs of each RBF unit, $\lambda_i o_i(x_k)$. The residual $e_{k,i}$ is obtained again by decomposing the total residual into G components according to Eq. (11),

but now the decomposition variables are

$$t_i = \frac{\alpha_2 \lambda_i}{\alpha_1 + \alpha_2 \sum_j \lambda_j^2}, \qquad i = 1, \ldots, G. \tag{16}$$

Therefore, the error is decomposed proportionally to the weights of the output layer. In Eq. (16), $1/(2\alpha_1)$ is the variance of the missing data given the input, and $1/(2\alpha_2)$ is the variance of the observations given the missing data. These parameters correspond to the proposed Gaussian probabilistic models.

The M-step consists of two consecutive steps: the centers and variances are adapted individually for each neuron as

$$\{\mu_i, \sigma_i\} = \arg \min_{\mu_i, \sigma_i} \sum_k (\hat{v}_{k,i} - o_i(x_k; \mu_i, \sigma_i))^2, \qquad i = 1, \ldots, G, \tag{17}$$

and, later, the weights λ_i are globally adapted according to

$$\{\lambda_i\} = \arg \min_{\{\lambda_i\}} \sum_k \left(y_k - \sum_i \lambda_i o_i(x_k; \mu_i, \sigma_i) \right)^2, \qquad i = 1, \ldots, G, \tag{18}$$

using the parameters μ_i and σ_i obtained in the previous partial step.

It is necessary to point out that by assuming the missing data to be stochastically independent (Feder & Weinstein, 1988) or to have a multivariate Gaussian distribution with diagonal covariance matrix (Ma et al., 1997), any cooperation between the components is destroyed, and this leads to the decoupled optimization scheme. While this decoupling accelerates the convergence, it can also lead to a poor local maximum of the likelihood surface. In despite of this shortcoming, the algorithm has been successfully applied in a number of applications.

4. Fast EM training of RBF networks

The decoupling variables (12) or (16), which are constant over the whole input space, have provided good results in feedforward neural networks with nonlocal activation functions, such as the Multilayer Perceptron (MLP). However, they are not well suited for networks with local activation functions, such as the RBF. For this type of networks its convergence is slow due to the fact that, using the previous decoupling variables, at each M-step we are trying to fit a Gaussian to a very large region of the input space. Intuitively, we could make a better job if we localize somehow the error associated to each RBF unit. This is the idea that we exploit in this paper to obtain a faster convergence.

4.1. Algorithm development

In this section we develop the EM-based learning algorithm, showing that the modifications introduced to take advantage of the locality of the RBF units still can be cast within the general framework established in Feder and Weinstein (1988). In this way, the convergence of our algorithm is guaranteed.

Following the idea shown in Feder and Weinstein (1988), from a set of N observations $\mathbf{y} = [y_1, \ldots, y_N]^T$ (incomplete data), the complete data is obtained by decomposing the observations into its signal components

$$z = \begin{bmatrix} \mathbf{z_1} \\ \mathbf{z_2} \\ \vdots \\ \mathbf{z_G} \end{bmatrix}, \qquad \text{with } \mathbf{z_i} = \begin{bmatrix} z_{1,i} \\ z_{2,i} \\ \vdots \\ z_{N,i} \end{bmatrix}, \tag{19}$$

where

$$z_{k,i} = g_i(x_k; \theta_i) + e_{k,i}. \tag{20}$$

The $e_{k,i}$ are obtained decomposing the total residual e_k into G components, so that the relation between the complete and the incomplete data can be written as follows

$$\mathbf{y} = \sum_{i=1}^{G} \mathbf{z_i} = \mathbf{H}\mathbf{z}, \tag{21}$$

where

$$\mathbf{H} = \underbrace{[\mathbf{I}\, \mathbf{I} \cdots \mathbf{I}]}_{G \text{ terms}}. \tag{22}$$

We have considered the $e_{k,i}$ statistically independent, zero-mean, and Gaussian. This assumption destroys the cooperation between the components, like in Feder and Weinstein (1988) as it was explained in Section 3. Therefore, it can yield a poor local maxima. In this case, the log-likelihood of the complete data is

$$\log f_Z(\mathbf{z}; \theta) = K - \tfrac{1}{2}(\mathbf{z} - \mathbf{s}(\theta))^T \Delta^{-1} (\mathbf{z} - \mathbf{s}(\theta)), \tag{23}$$

where

$$s(\theta) = \begin{bmatrix} \mathbf{g}_1(\theta_1) \\ \mathbf{g}_2(\theta_2) \\ \vdots \\ \mathbf{g_G}(\theta_G) \end{bmatrix}, \qquad \text{with } \mathbf{g_i}(\theta_i) = \begin{bmatrix} g_i(x_1; \theta_i) \\ g_i(x_2; \theta_i) \\ \vdots \\ g_i(x_N; \theta_i) \end{bmatrix}, \tag{24}$$

and

$$\Delta = \begin{bmatrix} Q_1 & & (0) \\ & Q_2 & \\ (0) & & Q_G \end{bmatrix}, \tag{25}$$

is a block-diagonal matrix. In Feder and Weinstein (1988), the Q_i matrix are obtained by

$$Q_i = t_i Q, \tag{26}$$

where Q is the covariance matrix of the error e_k, and t_i are arbitrary constants constrained to sum 1. In this paper, we propose to use the following decomposition

$$Q_i = T_i Q, \tag{27}$$

where T_i is the following diagonal matrix

$$T_i = \begin{bmatrix} t_{1,i} & & (0) \\ & \ddots & \\ (0) & & t_{N,i} \end{bmatrix}. \tag{28}$$

With this decomposition, the conditional expectation of the log-likelihood function, given the observations and the current value of the parameters, can be written as

$$E\{\log f_Z(\mathbf{z};\theta)|Y=\mathbf{y};\theta^{(n)}\} = K - \tfrac{1}{2}(\hat{\mathbf{z}}-\mathbf{s}(\theta))^T \Delta^{-1}(\hat{\mathbf{z}}-\mathbf{s}(\theta)), \tag{29}$$

where $\hat{\mathbf{z}}$ is the expectation of $\mathbf{z}$ given the observations and the current value of the parameters, $\theta^{(n)}$. As it is shown in Feder and Weinstein (1988), assuming that $\mathbf{z}$ and $\mathbf{y}$ are jointly Gaussian and related by the linear transformation (21), this expectation is given by

$$\hat{\mathbf{z}} = \mathbf{s}(\theta^{(n)}) + \Delta\mathbf{H}^T[\mathbf{H}\Delta\mathbf{H}^T]^{-1}[\mathbf{y}-\mathbf{H}\mathbf{s}(\theta^{(n)})]. \tag{30}$$

It is straightforward to see that Eq. (29) can be decoupled into a sum of G independent terms and, therefore, the EM algorithm is reduced to estimate the expectation of the complete data for each component. Taking into account Eq. (30), the EM algorithm can be stated as follows

E-Step: for $i = 1,\ldots,G$ compute

$$\hat{z}_{k,i} = g_i(x_k;\theta_i) + t_{k,i}\left[y_k - \sum_{j=1}^{G} g_j(x_k;\theta_j)\right]. \tag{31}$$

M-Step: for $i = 1,\ldots,G$ evaluate

$$\theta_i = \arg\min_{\theta_i} [\hat{z}_{k,i} - g_i(x_k;\theta_i)]^T Q_i^{-1} [\hat{z}_{k,i} - g_i(x_k;\theta_i)]. \tag{32}$$

Taking into account the nature of the Q_i matrix, the M-step states to minimize a weighted squared error. Intuitively, a better way to find the θ_i parameters, in the case of an RBF network, is to minimize the squared error. This intuition has been validated by experimental results. Then, the M-step used in the simulations is:

M-Step (Modified): for $i = 1,\ldots,G$ evaluate

$$\theta_i = \arg\min_{\theta_i} \sum_{k=1}^{N} (\hat{z}_{k,i} - g_i(x_k;\theta_i))^2. \tag{33}$$

4.2. Selection of the decoupling variables $t_{k,i}$

To take advantage of the local nature of the activation function of an RBF network, in this paper we propose to use as decoupling variables the following posterior probabilities

$$t_{k,i} = P(\psi_k = i | x_k, z_{k,i}), \tag{34}$$

where $\{z_{k,i}\}$ is the complete data set used in the E-step, and $\psi_k \in \{1,\ldots,G\}$ is an indicator variable: the event $\psi_k = i$ indicates that the kth input–output pattern $(x_k, z_{k,i})$ has been generated by the ith RBF unit. Using Eq. (34), the algorithm performs a soft adaptive decomposition of the residual taking into account the local nature of the activation functions. We denote this modification as soft-EM as opposed to the classical EM versions using constant decoupling variables (Feder & Weinstein, 1988; Ma et al., 1997).

Now we consider the estimation of Eq. (34): applying Bayes, the posterior probabilities can be estimated as

$$t_{k,i} = \frac{P(x_k, z_{k,i}|\psi_k = i)}{\sum_j P(x_k, z_{k,i}|\psi_k = j)}, \tag{35}$$

and the probabilities $P(x_k, z_{k,i}|\psi_k = j)$ can be obtained through

$$P(x_k, z_{k,i}|\psi_k = j) = P(z_{k,i}|\psi_k = j, x_k) P(x_k|\psi_k = j). \tag{36}$$

It is interesting to remark that the probabilities $P(x_k|\psi_k = j)$ are the key variables responsible for introducing the local character of the RBF units, since they can be estimated as

$$P(x_k|\psi_k = j) = \frac{o_j(x_k)}{\int o_j(x_k)\mathrm{d}x_k}. \tag{37}$$

In order to estimate $P(z_{k,i}|\psi_k = j, x_k)$, we have several possibilities depending on the assumed model for the data. We assume the following Gaussian model

$$P(z_{k,i}|\psi_k = j, x_k) = \frac{1}{\sqrt{2\pi}\sigma_{k,i}} \exp - \frac{(z_{k,i} - g_j(x_k))^2}{2\sigma_{k,i}^2}, \tag{38}$$

where the variance $\sigma_{k,i}^2$ can be estimated at each iteration as

$$\sigma_{k,i}^2 = \hat{t}_{k,i}\sigma^2, \tag{39}$$

where σ^2 is the variance of the error, and $\hat{t}_{k,i}$ are the decoupling variables obtained in the previous EM iteration. These variables are initialized proportionally to the output of their

Table 1
Functions used to generate the two-dimensional data sets

Name	Function	Domain
Func 1	$y = \sin(x_1 x_2)$	$[-2,2]$
Func 2	$y = \exp(x_1 \sin(\pi x_2))$	$[-1,1]$
Func 3	$y = \frac{40 \exp(8((x_1-0.5)^2+(x_2-0.5)^2))}{\exp(8((x_1-0.2)^2+(x_2-0.7)^2))+\exp(8((x_1-0.7^2+(x_2-0.2)^2))}$	$[0,1]$
Func 4	$y = (1+\sin(2x_1+3x_2))/(3.5+\sin(x_1-x_2))$	$[-2,2]$
Func 5	$y = 42.659(0.1+x_1(0.05+x_1^4-10x_1^2x_2^2+5x_2^4))$	$[-0.5,0.5]$
Func 6	$y = 1.3356[1.5(1-x_1)+\exp(2x_1-1)\sin(3\pi(x_1-0.6)^2)+\exp(3(x_2-0.5))\sin(4\pi(x_2-0.9)^2)]$	$[0,1]$
Func 7	$y = 1.9[1.35+\exp(x_1)\sin(13(x_1-0.6)^2]+\exp(3(x_2-0.5))\sin(4\pi(x_2-0.9)^2)]$	$[0,1]$
Func 8	$y = \sin(2\pi\sqrt{x_1^2+x_2^2})$	$[-1,1]$

corresponding RBF unit

$$\hat{t}_{k,i}^{(\mathrm{ini})} = \frac{g_i(x_k;\theta_i)}{\sum_j g_j(x_k;\theta_j)}. \tag{40}$$

Taking into account the transformation (21), this model is consistent with the overall noise model (9), since all the models are Gaussian and the decoupling variables are constrained to sum 1.

To summarize, using Eqs. (36)–(38) we estimate the decoupling variables as in Eq. (35): these estimates are then used in the E-step.

In the maximization step, new RBF parameters need to be obtained through minimizing Eq. (14) for each component. The amplitude of each RBF unit can be obtained solving a linear least squares problem, while its center and variance can be updated using a gradient descent procedure (Santamaría et al., 1999). To accomplish with the EM requirements, the gradient algorithm must be iterated at each M-step until convergence is reached. However, another possibility is to carry out only a limited number of gradient iterations. In this way, Eq. (14) is not minimized, but only decreased. In this case, the algorithm become a Generalized EM (GEM) algorithm, a variant of the EM algorithm that instead of requiring the maximization of the expectation of the log-likelihood of the complete data at each E-step, it only requires that the expectation be improved (Dempster et al., 1977).

Let us point out that each RBF unit is adapted separately therefore simplifying the global optimization problem and allowing an easy parallelization. The extension to multi-dimensional input spaces is straightforward.

5. Experimental results

In this experiment we consider the set of eight 2-D functions used in Cherkassky, Gehring, and Mulier (1996) to compare the performance of several adaptive methods. These functions, which form a suitable test set, are described in Table 1. We use a generalized radial basis function (GRBF) allowing a different variance along each input dimension.

First, we compare the performance of the proposed soft-EM approach with the classical EM alternatives (Feder & Weinstein, 1988; Ma et al., 1997), denoted as EM-1 and EM-2, respectively. A GRBF network with 15 neurons is considered. The network is initialized as follows: the position of the centers is obtained using the orthogonal least squares (OLS) algorithm (Chen, Cowan, & Grant, 1991), with different initial values of σ^2. The λ_i parameters are then obtained by least squares. A single iteration of gradient is used in the E-step of all methods. Several values of parameters α_1 and α_2 in Ma et al. (1997) have been tested. The best results have been obtained for $\alpha_1 = 1, \alpha_2 = 5$.

Fig. 1 shows the evolution of the normalized MSE as a function of the EM iterations. In this case, the MSE corresponds to the mean value obtained in the eight functions. In any case, a similar behavior has been observed for each individual function. It can be seen how the soft-EM approach provides a fast convergence.

In order to ensure a fair comparison, we want to remark that the initialization procedure meets the requirements

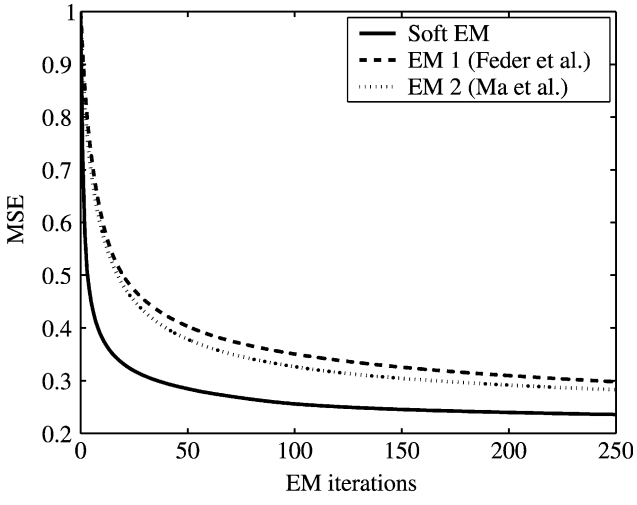

Fig. 1. Mean evolution of the normalized MSE for the different EM alternatives in the eight 2D test functions.

Table 2
Mean SER (dB) for the different test functions

	Func 1	Func 2	Func 3	Func 4	Func 5	Func 6	Func 7	Func 8
Gradient	20.4	40.5	32.2	19.0	26.1	28.7	26.3	16.2
EM-1	17.9	30.6	34.7	14.1	27.4	29.4	30.5	14.7
EM-2	18.6	29.8	34.8	16.3	27.8	28.9	30.5	15.0
Soft-EM	20.0	39.3	42.9	19.5	31.8	33.6	34.3	16.3

discussed in Ma et al. (1997), since the components tend to be uniformly placed in the entire input domain and behave differently from the beginning of the algorithm. Moreover, other initialization strategies have been tested providing similar results: for instance, the centers in a uniform distribution drawing a grid in the input space.

The soft-EM has also been compared with a global gradient descent approach (Santamaría et al., 1999). Again, a GRBF network with 15 neurons is employed and 300 iterations of the gradient method and the EM approaches are carried out. We apply a single gradient iteration at each M-step. In this way both alternatives have a similar computational cost. The same initialization procedure of the previous experiment is employed, with 10 different initial values of σ^2 for each function. In this case, the gradient based method and the soft-EM approach have a similar behavior in the speed of convergence. However, the final results are better using the soft-EM approach. Table 2 shows the mean signal to error ratio (SER) in dB obtained for each function with the gradient based approach and with the EM based methods.

It can be seen that, for most of the functions, the soft-EM provides better results than the global gradient and the conventional EM approaches.

6. Nonlinear small-signal modeling of a MESFET for intermodulation distortion characterization

In this section, a GRBF network trained with the proposed soft-EM procedure is used to reproduce the small-signal intermodulation behavior of a microwave MESFET transistor. Fig. 2 shows the most widely accepted equivalent nonlinear circuit of a MESFET in its saturated region. The predominant nonlinearity in this model is the drain to source current I_{ds}, which depends on both the drain to source, V_{ds}, and gate to source, V_{gs}, voltages. Here we are going to model this static nonlinearity.

Reproducing the small-signal third order intermodulation distortion (IMD) behavior is quite a common and difficult task: amplifiers working below the 1 dB compression point and mixers excited by small RF signals when compared with the local oscillator are typical examples. In this case, to be able to predict the IMD behavior, the model must describe not only the nonlinear current–voltage (I/V) characteristic, but also its respective derivatives up to the same order (Maas & Neilson, 1990). In particular, the drain to source current I_{ds} can be represented in a small interval around the bias point, (V_{ds0},V_{gs0}), by the following two-dimensional truncated Taylor series expansion

$$I_{ds}=I_{ds0}+G_m v_{gs}+G_d v_{ds}+G_{m2}v_{gs}^2+G_{md}v_{gs}v_{ds}+G_{d2}v_{ds}^2 +G_{m3}v_{gs}^3+G_{m2d}v_{gs}^2v_{ds}+G_{md2}v_{gs}v_{ds}^2+G_{d3}v_{ds}^3, \quad (41)$$

where I_{ds0} is the dc drain current, v_{ds} and v_{gs} are the incremental drain-to-source and gate-to-source voltages, respectively; and $(G_m,\ldots,G_{d3})$ are coefficients related to the nth-order derivatives of the I/V characteristic evaluated at the bias point. For example, G_{md2} would be given by

$$G_{md2}=\frac{1}{2}\frac{\partial^3 I_{ds}}{\partial v_{gs}\partial v_{ds}^2}\bigg|_{(V_{gs0},V_{ds0})}, \quad \text{in } v_{ds}=v_{gs}=0. \quad (42)$$

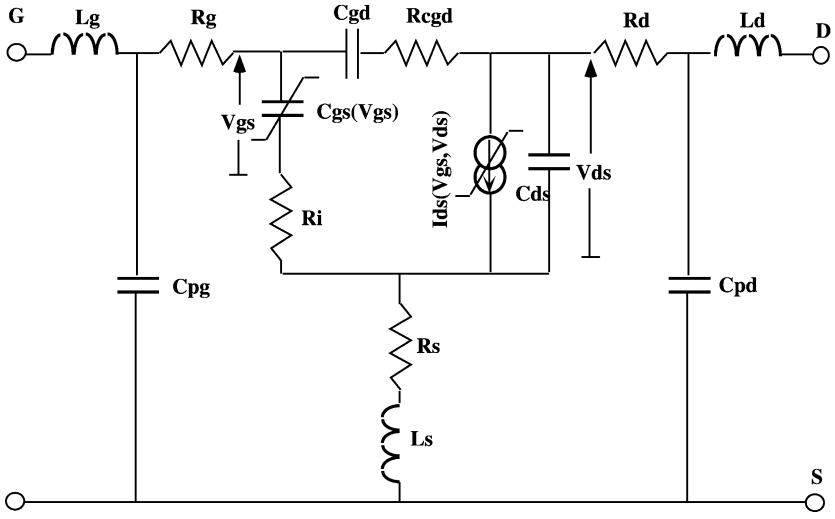

Fig. 2. Nonlinear equivalent circuit of a MESFET transistor.

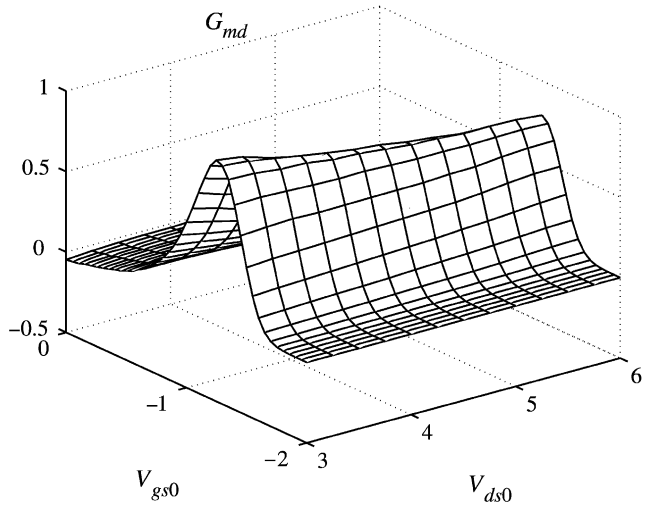

Fig. 3. Normalized value of the parameter G_{md} as a function of the polarization voltages.

These parameters can be extracted from experimental measurements. The first order coefficients are determined from scattering parameters and the rest from intermodulation power measurements for two tones excitation (Pedro & Pérez, 1994). With this model for I_{ds}, it is possible to approximate, in the neighborhood of the bias point, the (I/V) characteristic and its derivatives.

To summarize, the modeling problem we are facing consists in approximating the nonlinear dependence of I_{ds} with the bias and instantaneous drain to source and gate to source voltages

$$\hat{I}_{ds} = f(V_{ds0}, V_{gs0}, v_{ds}, v_{gs}). \tag{43}$$

After obtaining a set of real measurements of the coefficients in the Taylor series (41), the problem can be stated as the $\mathbf{G}: R^2 \rightarrow R^{10}$ multidimensional function approximation problem that implements the nonlinear mapping from the input space of bias voltages $\mathbf{V} = (V_{ds0}, V_{gs0})$, to the output space composed by the coefficients of the Taylor series

$$\mathbf{G}(\mathbf{V}) = (I_{ds0}, G_m, G_{ds}, G_{m2}, G_{md}, G_{d2}, G_{m3}, G_{m2d}, G_{md2}, G_{d3}). \tag{44}$$

Then, $\hat{I}_{ds}$ is estimated by Eq. (41).

The coefficients related with the derivatives of the I/V characteristic have been measured for a microwave NE 72084 MESFET in 533 different bias points (for instance, the normalized shape of G_{md} is shown in Fig. 3). Therefore, our GRBF model receives as input these bias voltages (V_{ds0}, V_{gs0}) and gives at its output the coefficients (44). In this case, because of the reduced number of patterns and the general reduced level of noise of the samples, we have not considered separated sets of training and test. The network is trained with the whole available measurements. The network is initialized by selecting an initial variance and applying the OLS algorithm (Chen et al., 1991).

In Table 3 we compare the results obtained with a GRBF network with 8 units (the models has 112 parameters), trained with the proposed soft-EM algorithm (GRBF-EM) and with a global gradient algorithm (GRBF-GRD) (Santamaría et al., 1999), and the results provided by a MLP network with a similar number of parameters (114). It can be seen that, even with this reduced number of parameters, a suitable approximation of all the coefficients is obtained. The soft-EM provides better results than the gradient based approach using the GRBF network. Moreover, let us note that to provide similar results a MLP model requires approximately twice as many parameters.

Fig. 4 shows the evolution of the normalized MSE with the number of iterations for the soft-EM method and compares it with the conventional EM approaches. For EM-2, we have used the parameters $\alpha_1 = 1$ and $\alpha_2 = 5$. The averaged results, starting from 50 different initial conditions (different initial variances), are presented. It can be seen that the proposed soft-EM approach improves the convergence of the training algorithm also in this application.

7. Conclusions and future work

The decoupling variables used in the E-step of EM-based learning algorithms can be selected to control the rate of convergence of the algorithm. We have studied in this paper a suitable selection of these variables for feedforward networks with local activation functions (mainly, RBF networks). Specifically, these variables are estimated as the posterior probability of each RBF unit given each pattern of the selected complete data. By

Table 3
Results (SER in dB) of the GRBF model with eight neurons for a NE 72084 MESFET using the soft-EM (GRBF-EM) and a gradient-based training algorithm (GRBF-GRD). These results are compared with the obtained with a MLP with 8 and 16 neurons, respectively

Model	N_{par}	I_{ds0}	G_{d2}	G_{d3}	G_{ds}	G_{m1}	G_{m2S}	G_{m2d}	G_{m3}	G_{md}	G_{mds}
GRBF-EM	112	25.0	18.5	15.2	26.0	25.5	24.1	21.3	20.9	24.2	21.2
GRBF-GRD	112	22.8	19.1	15.7	25.4	27.2	17.0	18.0	18.5	17.9	16.1
MLP (8)	114	18.7	13.5	14.0	24.6	30.0	13.5	9.9	8.9	13.6	10.7
MLP (16)	218	20.0	14.4	16.7	29.1	31.7	16.3	15.4	14.3	16.3	14.2

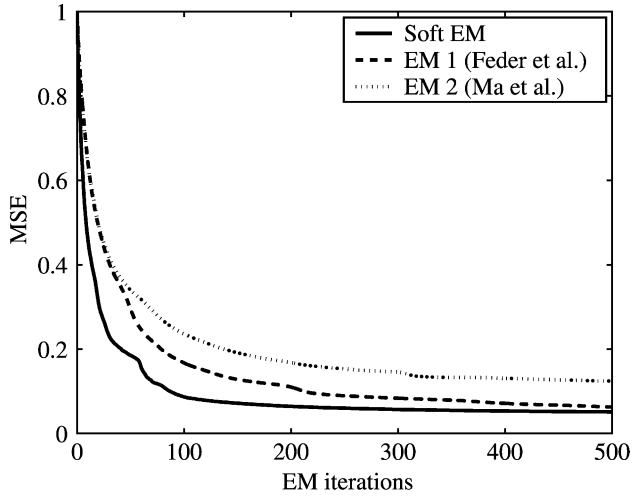

Fig. 4. Evolution of the normalized MSE for the different EM alternatives in the training of a MESFET model for intermodulation behavior.

means of several simulation examples, it has been shown that this modification accelerates the convergence of the algorithm.

The proposed soft-EM approach has been applied to model the small-signal behavior of a MESFET transistor to reproduce the IMD, providing suitable results.

There are several lines of investigation to carry out in the future. In particular, an efficient training method for a single neuron of an RBF network will help to reduce the computational burden introduced by the actual gradient based strategy. We also consider interesting to analyze the decoupling variables of the algorithm as a tool that could be employed in a pruning strategy in order to find the optimal size of the network. Since these variables carry information about the contribution of each RBF unit, they could be used to select the more important ones.

Acknowledgements

This work has been partially supported by the European Community and the Spanish Government through FEDER project 1FD97-1863-C02-01. The authors also thank the reviewers for careful reading the manuscript and for many helpful comments.

References

Bishop, C. (1997). *Neural networks for pattern recognition*. Oxford: Clarendon Press.

Broomhead, D. S., & Lowe, D. (1988). Multivariable functional interpolation and adaptive networks. *Complex Systems*, *2*, 321–355.

Chen, S., Cowan, C. F. N., & Grant, P. M. (1991). Orthogonal least squares learning algorithm for radial basis functions. *IEEE Transactions on Neural Networks*, *2*(2), 302–309.

Cherkassky, V., Gehring, D., & Mulier, F. (1996). Comparison of adaptive methods for function estimation from samples. *IEEE Transactions on Neural Networks*, *7*(4), 969–984.

Dempster, A. P., Laird, N., & Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journals of Royal Statistics Society B*, *39*, 1–38.

Feder, M., & Weinstein, E. (1988). Parameter estimation of superimposed signals using the EM algorithm. *IEEE Transactions on Acoustics, Speech and Signal Processing*, *36*, 477–489.

Ghahramani, Z., & Jordan, M. I. (1994). In J. D. Cowan, G. Tesauro, & J. Alspector (Eds.), *Supervised learning from incomplete data via an EM approach. Advances in NIPS VI*, San Mateo, CA: Morgan Kaufmann.

Haykin, S. (1994). *Neural networks: A comprehensive foundation*. Englewoods Cliffs, NJ: Macmillan.

Jordan, M. I., & Jacobs, R. A. (1994). Hierarchical mixtures of experts and the EM algorithm. *Neural Computation*, *6*, 181–214.

Karayanis, N (1997). Gradient descent learning of radial basis neural networks. *Proceedings of the IEEE International Conference on Neural Networks, Houston, TX* (pp. 1825–1830).

Lowe, D (1989). Adaptive radial basis functions nonlinearities and the problem of generalization. *First IEEE Conference on Artificial Neural Networks, London, UK* (pp. 171–175).

Ma, S., & Ji, C. (1998). Fast training of recurrent networks based on the EM algorithm. *IEEE Transactions on Neural Networks*, *9*, 11–26.

Ma, S., Ji, C., & Farmer, J. (1997). An efficient EM-based training algorithm for feedforward neural networks. *Neural Networks*, *10*, 243–256.

Maas, S. A., & Neilson, D. (1990). Modeling MESFETs for intermodulation analysis of mixers and amplifiers. *IEEE Transactions Microwave Theory and Techniques*, *38*(12), 1964–1971.

Moody, J. E., & Darken, C. J. (1989). Fast learning in networks of locally-tuned processing units. *Neural Computation*, *1*, 281–294.

Pedro, J. C., & Pérez, J. (1994). Accurate simulation of GaAs MESFETs intermodulation using a new drain-source current model. *IEEE Transactions Microwave Theory and Techniques*, *42*(1), 25–33.

Santamaría, I., Lázaro, M., Pantaleón, C. J., García, J. A., Tazón, A., & Mediavilla, A. (1999). A nonlinear MESFET model for intermodulation analysis using a generalized radial basis function network. *Neurocomputing*, *25*, 1–18.

Wu, C. F. (1983). On the convergence properties of the EM algorithm. *Annals of Statistics*, *11*, 95–103.