Contributed article

Generalized radial basis function networks for classification and novelty detection: self-organization of optimal Bayesian decision

S. Albrecht, J. Busch, M. Kloppenburg, F. Metze, P. Tavan*

Institut für Medizinische Optik, Theoretische Biophysik, Ludwig Maximilians Universität München, Oettingenstraße 67, D-80538 München, Germany

Received 16 March 1999; accepted 24 May 2000

Abstract

By adding reverse connections from the output layer to the central layer it is shown how a generalized radial basis functions (GRBF) network can self-organize to form a Bayesian classifier, which is also capable of novelty detection. For this purpose, three stochastic sequential learning rules are introduced from biological considerations which pertain to the centers, the shapes, and the widths of the receptive fields of the neurons and allow a joint optimization of all network parameters. The rules are shown to generate maximum-likelihood estimates of the class-conditional probability density functions of labeled data in terms of multivariate normal mixtures. Upon combination with a hierarchy of deterministic annealing procedures, which implement a multiple-scale approach, the learning process can avoid the convergence problems hampering conventional expectation-maximization algorithms. Using an example from the field of speech recognition, the stages of the learning process and the capabilities of the self-organizing GRBF classifier are illustrated. © 2000 Elsevier Science Ltd. All rights reserved.

Keywords: Generalized radial basis functions; Self-organization; Classification; Maximum-likelihood density estimation

1. Introduction

Nowadays so-called radial basis function (RBF) networks have become one of the major tools of neuroinformatics (for a review see, e.g. Bishop (1997)). As stated by Ripley (1996), however, there is a "plethora of methods" which "provides a difficulty in assessing RBF methods; no two workers use the same class of RBFs and method of fitting". We do not want to increase that confusion. Therefore we will first sketch a classification scheme for RBFs which allows us to put our work into proper perspective and to introduce our notation.

1.1. Basic notions and notations

Fig. 1 shows the general structure of an RBF network. An RBF is fed with D-dimensional vectors $x \in \mathbb{R}^D$ whose components $x_i$ are interpreted as activities of $D$ input neurons $i$. Through synaptic connections with weights $c_{ri}$ the input neurons activate $M$ neurons $r$ of the central layer. The induced activities are calculated from non-linear activation functions $a_r(x)$ and parametrically depend on the weights $c_{ri}$ converging at the neurons $r$, which make up dendritic tree vectors $c_r \in \mathbb{R}^{D}$. The activities $a_r$ in the central layer are transferred through synaptic connections $f_{jr}$ to $K$ linear output neurons $j$. The activities $\hat{f}_j(x)$ induced at these neurons are collected into the vector $\hat{f}(x) \in \mathbb{R}^{K}$ representing the output of the net. Collecting the synaptic weights $f_{jr}$ diverging from the neurons $r$ into axonal tree vectors $f_r$, the output is given by

$$\hat{f}(x) = \sum_{r=1}^{M} f_r a_r(x).$$

Thus, an RBF represents a non-linear mapping

$$\hat{f} : x \in \mathbb{R}^D \rightarrow \hat{f}(x) \in \mathbb{R}^{K}.$$ (2)

It is the choice of a radial response function for the neurons of the central layer, which distinguishes RBFs from other three-layer perceptrons.

1.2. Specification of RBF and GRBF networks

We employ response functions $a_r(x)$, which are derived from Gaussians in their most general multivariate form. Thus, we associate to each neuron $r$ of the central layer a
multivariate normal distribution given by
\[ \hat{p}(x|r, \theta_r) = \frac{\exp\left[-(x - c_r)^T \Sigma_r^{-1} (x - c_r)/2\right]}{(2\pi)^{D/2} |\Sigma_r|^{1/2}}. \] (3)

Each of these distributions depends on a set of local parameters \( \theta_r \) covering the centers \( c_r \) and covariance matrices \( \Sigma_r \). Using the spectral decompositions of the \( \Sigma_r \) into eigenvectors \( \mathbf{w}_r \) and eigenvalues \( \sigma_{ir}^2 \), \( i = 1, \ldots, D \), the inverse matrices \( \Sigma_r^{-1} \) can be conveniently expressed in terms of the orthonormal diagonalizing transformations \( \mathbf{W}_r = (\mathbf{w}_{1r}, \ldots, \mathbf{w}_{Dr}) \) and of the diagonal matrices \( \Sigma_r \) of eigenvalues through
\[ \Sigma_r^{-1} = \mathbf{W}_r \Sigma_r^{-1} \mathbf{W}_r^T. \] (4)

For the determinants one has \( \det \Sigma_r = \det \hat{\Sigma}_r \). Thus, the parameters of the Gaussians are the sets \( \theta_r = \{c_r, \hat{\Sigma}_r, \mathbf{W}_r\} \).

Biologically the choice of such functions has been motivated by the observation that many neurons in sensory cortical areas exhibit stimulus-response characteristics, which are more or less broadly tuned to feature combinations \( x \) centered around a best stimulus \( c_r \). In that view the Gaussians define the receptive fields of such neurons (see Poggio & Giro, 1989, 1990a,b).

Now the response functions \( a_r \) of the neurons \( r \) can be written as
\[ a_r(x|\Theta_r) = \delta_r(x|\hat{P}_r, \theta_r), \] (5)
where \( \hat{P}_r \) is a positive amplitude. Owing to the \( \hat{P}_r \), the maximal response becomes independent of the normalization factor \( (2\pi)^{D/2} |\Sigma_r|^{1/2} \). Choice (5) has been the predominant one in previous studies of RBF networks (see e.g. Chen & Chen, 1995; Krzyzak & Linder, 1998; Tara senko & Roberts, 1994). However, we prefer a globally normalized alternative first suggested by Moody and Darken (1989). Defining the total activity of the central layer by
\[ A(x|\Theta_r) = \sum_{r=1}^{M} a_r(x|\hat{P}_r, \theta_r), \] (6)
where \( \Theta_r \) denotes the corresponding parameter set \( \{\hat{P}_r, \theta_r\}_{r=1}^{M} \), the globally normalized response function is
\[ a_r(x|\Theta_r) = \begin{cases} a_r(x|\hat{P}_r, \theta_r)/A(x|\Theta_r) & \text{if } A(x|\Theta_r) \geq \epsilon \geq 0, \\ 0 & \text{else.} \end{cases} \] (7)

Here, we have introduced a small cut-off parameter \( \epsilon \) in order to confine non-zero activity responses of the central layer to stimuli \( x \) from a bounded region \( \mathcal{T}_\epsilon \subset \mathbb{R}^D \) within input space. From a biological perspective, the cut-off accounts for the well-known threshold behavior of cortical neurons (cf. Willshaw & Malsburg, 1976); for biological motivations of the activity normalization we refer to Hillermeier, Kunstmann, Rabus, and Tavan (1994), Moody and Darken (1989), Tavan, Grubmüller, and Kühnel (1990), and Willshaw and Malsburg (1976).

Our choice of the normalized activation function (7) has several mathematical reasons. Two of them will be noted now.

1. Consider the special case of one-dimensional input and output spaces \( (D = K = 1) \). Then the covariance matrices \( \Sigma_r \) reduce to the variances \( \sigma_r^2 \) of the Gaussians. Consider furthermore the limit of vanishing variances \( \sigma_r^2 \to 0 \). Then the initial response functions \( a_r \) become delta-distributions. In contrast, the normalized activation functions \( a_r \) become tesselation functions, each of which assumes the value 1 within a certain interval \( I_r \subset \mathcal{T}_\epsilon \) and the value 0 outside. Thus the RBF (1) becomes a step function \( f(x) \) with values \( f \) for \( x \in I_r \). This step function can approximate any integrable function \( f(x) \) on \( \mathcal{T}_\epsilon \) with arbitrary precision, if the discretization of \( \mathcal{T}_\epsilon \) by the points \( c_r \) is chosen sufficiently dense. A similarly reasonable limit does not exist for the response functions \( a_r \). These arguments1 can be generalized to higher-dimensional cases and multivariate distributions.

2. The normalization may be reformulated as
\[ \sum_{r=1}^{M} a_r(x|\Theta_r) = 1 \quad \text{for } x \in \mathcal{T}_\epsilon \] (8)

---

1. We are not aware that these arguments have been given elsewhere.
implying that the activation functions $a_i(x|\Theta_c)$ define a fuzzy partition (Grauel, 1995) of the bounded input region $\mathcal{F}$, which becomes a Voronoi tessellation in the limit of vanishing variances.

The normalized activation functions do not exhibit a simple radial decay characteristics anymore. Although being still centered around the $c_i$, they have more complicated shapes. Therefore, we call these functions GRBFs, where the $G$ stands for generalized, and we suggest reserving the short-cut RBF for networks based on the truly radial response functions $a_i$ given by Eqs. (3) and (5).

1.3. Ideal vs. non-ideal GRBF networks

Our restriction to GRBF networks strongly confines the set of possible models. For a unique classification, however, further criteria are required, some of which will now adopt from the work of Xu, Krzyzak, and Yuille (1994).

In order to derive a classification scheme, these authors have analyzed the various suggestions as to how one should estimate the network parameters $\Theta$ covering the sets $\Theta_c$ given above and $\Theta_r = \{r_i| r = 1, ..., M\}$, i.e.

$$\Theta = \{c_r, \Sigma_r, W_r, \hat{P}_r, \hat{f}_r| r = 1, ..., M\}. \quad (9)$$

Here, they considered GRBF networks $\hat{f}(x|\Theta)$ as tools for general non-linear regression (Specht, 1991; Trâve, 1991), i.e. as tools for the approximation of non-linear target functions

$$f : x \in \mathbb{R}^D \rightarrow f(x) \in \mathbb{R}^K$$

which are given through training sets of noisy samples

$$\mathcal{D} = \{(x_i, y_i) \in \mathbb{R}^D \times \mathbb{R}^K | i, ..., N\}. \quad (10)$$

In that view, procedures for parameter estimation serve the purpose that the GRBF approximation

$$\hat{f}(x_i|\Theta) = y_i, \quad i = 1, ..., N, \quad (12)$$

is as good as possible according to some predefined cost function.

To support and explain the classification scheme of Xu et al. (1994), we note\(^2\) that the approximation problem (12) has a trivial solution for $M = N$. Then the approximations in Eq. (12) immediately become strict equalities for the choice $c_r = x_r$, $f_r = y_r$, $W_r = I$, $\Sigma_r = \sigma^2 I$, and for the limit $\sigma \rightarrow 0$ (here, $I$ is the identity matrix and $r = 1, ..., N$). Thus, the trivial solution is just the step function sketched further above.\(^2\) The trivial case $M = N$, however, is of no interest in large-scale applications, in which huge sample data sets $\mathcal{D}$ are accessible. Here, data compression is required and is implemented by a choice $M \ll N$. In this case learning becomes a hard problem of non-linear optimization (Xu et al., 1994), as long as a joint optimization of all network parameters $\Theta$ is desired.

Xu et al. (1994) classified models, which actually aim at such a joint optimization, as being of the ideal type: They were able to show that such a joint optimization in principle can yield better approximations than other learning strategies. However, they also noted that straightforward applications of gradient descent procedures to the joint optimization in practice also yield sub-optimal solutions. Because of these convergence problems, this approach largely has been avoided in previous studies.

Instead, in order to reduce the complexity of the problem of parameter estimation, various simplifications rendering non-ideal GRBF models have been suggested, which generally amount to: (i) separate optimizations of the parameter sets $\Theta_c$ and $\Theta_r$, respectively; (ii) strong restrictions to the shapes $\Sigma_r$ of the receptive fields such as $\Sigma_r = \sigma^2 I$ (univariate Gaussians) or even $\Sigma_r = \sigma^2 I$ (univariate Gaussians with identical widths); and (iii) various heuristics for the choice of the widths $\sigma_r$.

Using the amount of simplification as a rough measure Xu et al. (1994) suggested a partitioning of these non-ideal GRBF models into further sub-classes. Accordingly, the trivial case $M = N$ is classified as being of the basic type, whereas models with $M \ll N$, which determine the synaptic weights $c_r$ by some clustering algorithm from the sample points $x_i \in \mathbb{R}^D$ (Moody & Darken, 1989), are classified into the clustering-aided type. Since the quality of a GRBF necessarily decreases with increasing simplification of the learning strategy (Xu et al., 1994), our paper aims at the construction of a GRBF model of the ideal type and therefore, will have to address the problem of full non-linear optimization.

The above classification of GRBF networks took the applied learning strategies as criteria. In our view, however, these criteria have to be complemented by the particular purpose for which a GRBF is to be used.

1.4. GRBF networks for classification

If a GRBF is used for the purpose of classification then both the training data set (11) and the target function (10) are of a very special kind:

1. The training set consists of pattern vectors $x_i$ partitioned into $K$ classes $j$. For class labeling the sample function values $y_i$ are taken from the set of unit vectors $\{\tilde{e}_j \in \mathbb{R}^K | j = 1, ..., K\}$, where $y_i = \tilde{e}_{y_i}$ codes the membership of $x_i$ to class $j$. Thus, during training the activity of the output neuron $j$ is set to the value 1 and the activities of the other output neurons are set to 0, if the input pattern $x_i$ belongs to class $j$.

2. The components $f_j(x)$ of the target function $f = \hat{f}(x)$ are 

\(^2\) Equalities in Eq. (12) are also obtained for the choice of a fuzzy partition, i.e. for $\sigma > 0$, if one solves the system of N linear equations $\sum A_{rij} = y_r$ for the output weights $f_r$ through inversion of the $N \times N$-matrix $A_{rr} = \sigma^2 I$. But if, instead, one simply sets the $f_r$ to the values $f_r = y_r$, then the GRBF represents an approximation (12) on $\mathcal{D}$ and reduces to the general regression neural network suggested by Specht (1991).
According to Bayes theorem the association probabilities have not been given yet. To decision. However, in our view a mathematically rigorous review) there are obvious similarities between GRBF concerning training data and target functions sets. According to Basi et al. (1997) like the expectation-maximization (EM) severe convergence problems (Duda & Hart, 1973; Streit & Lugten, 1994) are much simpler because of the restrictions concerning training data and target functions sets. Accordingly, this paper is devoted to the construction of a GRBF classifier of the ideal type.

1.5. GRBFs and Bayesian maximum-likelihood classifiers

As noted earlier by many authors (see Bishop (1997) for a review) there are obvious similarities between GRBF networks for classification and parametric techniques for density estimation (Duda & Hart, 1973) aiming at Bayesian decision. However, in our view a mathematically rigorous analysis of these relations has not been given yet. To provide a typical example for our general critique concerning the treatment of this issue in the literature we consider the arguments of Bishop (1997) in Section 5.7 of his textbook starting with a recapitulation of basic facts.

According to Bayes theorem the association probabilities \(P(j|x)\), which have to be modeled in classification problems (cf. Eq. (13)), derive from the class-conditional probability density functions (PDFs) \(p(x|j)\) and from the statistical weights \(P(j)\) of the \(K\) classes \(j\) through

\[
P(j|x) = \frac{P(j)p(x|j)}{\sum_{l=1}^{K} P(l)p(x|l)}. \tag{14}
\]

The class-conditional PDFs \(p(x|j)\) can be approximated by mixture PDFs

\[
\hat{p}(x|j, \Theta_j) = \sum_{r=1}^{m(j)} \hat{P}(r|j)\hat{p}(x|r, \theta_j), \tag{15}
\]

of \(m(j)\) multivariate normal distributions (3) labeled by the index \(s\) and the class \(j\), where the parameters \(\Theta_j\) should be chosen to satisfy the maximum-likelihood (ML) principle. Also, this approach is hampered by severe convergence problems (Duda & Hart, 1973; Yang & Chen, 1998) if gradient ascent procedures (Trav?n, 1991) like the expectation-maximization (EM) algorithm (Dempster, Laird, & Rubin, 1977; Streit & Lugten, 1994) are used. In line with Duda and Hart (1973) but contrary to Yang and Chen (1998) we attribute these problems to intrinsic instabilities of the non-linear learning dynamics defined by the EM equations. The identification and removal of these problems has one of the main motivations of our work (see also Kloppenburg & Tavan, 1997).

Inserting the class-conditional ML mixture densities (15) together with ML estimates \(\hat{P}(j)\) of the statistical weights \(P(j)\) into Bayes theorem (14) one obtains approximate association probabilities

\[
\hat{P}(j|x, \Theta_j) = \sum_{s=1}^{m(j)} \hat{P}(s|j)\hat{p}(x|s, j, \theta_s)/\hat{p}(x|\Theta_j), \tag{16}
\]

where \(\Theta_j\) covers the class-conditional parameter sets \(\Theta_s\) and the \(\hat{P}(j)\), such that the PDF

\[
\hat{p}(x|\Theta_j) = \sum_{s=1}^{K} \sum_{r=1}^{m(j)} \hat{P}(r|j)\hat{p}(x|r, \theta_s)/\hat{p}(x|\Theta_j). \tag{17}
\]

is an ML estimate for the distribution \(p(x)\) of all patterns in the training set. We call classification tools constructed in this statistically sound way Bayesian maximum-likelihood classifiers (BMLC). The quality of such a classifier solely depends on the quality of density estimation (measured by the value of the likelihood on the training set).

Considering the special case of the BMLC (16), which is restricted to only one normal distribution per class \(m(j) = 1\), Bishop notes that it can be viewed “as a simple form” of a GRBF. This interpretation also holds (see Section 2) for the general case \(m(j) > 1\), if the GRBF response functions (7) are given by

\[
a_{s,j}(x|\Theta_s) = \hat{P}(s|j)\hat{p}(x|s, j, \theta_s)/\hat{p}(x|\Theta_s). \tag{18}
\]

As indicated by the neuron label \((s, j) = r\), this “simple GRBF” is characterized by the property that each neuron \(r\) of the central layer is uniquely associated to one of the classes \(j\). We will denote this model in the following as GRBF/BMLC.

Although Bishop mentions the possibility of using “a separate mixture model to represent each of the conditional densities”, he discards this approach as computationally inefficient and proposes to “use a common pool” of \(M\) basis functions \(r\) to represent all class-conditional densities. Accordingly he suggests writing

\[
\hat{p}(x|j, \Theta_s) = \sum_{r=1}^{M} \hat{P}(r|j)\hat{p}(x|r, \theta_r) \tag{19}
\]

instead of Eq. (15), but fails to note (cf. p. 180) that this seemingly plausible expression contains an additional uncontrolled approximation. As follows from the exact decomposition \(p(x, r|j) = P(r|j)p(x|r, j)\) by summing over \(r\), the components of the mixture (17) have to depend on the class \(j\). This dependence is correctly contained in the BMLC expression (17). From Eq. (19) Bishop derives a GRBF classifier of the clustering-aided type, which we will denote
by GRBF/CATC and for which he discusses various training algorithms. However, his claim “that the outputs of this network also have a precise interpretation as the posterior probabilities of class membership” (p. 182) is wrong: (i) because of the uncontrolled approximation sketched above, and (ii) because it remains to be proven that the applied training procedures yield network parameters compatible with the envisaged statistical interpretation (we will address this issue in Section 3). On the other hand, we agree that “the ability to interpret network outputs in this way” (cf. (13)) is of “central importance”.

1.6. Outline and scope of the paper

We have provided this detailed discussion of Bishop’s review on GRBF classifiers because it enabled us to work out an uncontrolled approximation, which, as we will show, is inherent to all GRBF/CATC. As will be explained in Section 3 we have detected this approximation in an attempt to assign a precise statistical meaning to the constituent parts of a GRBF/CATC. The inevitable failure of this attempt leads us to conclude that the “simple GRBF” discarded by Bishop actually is the only statistically correct GRBF classifier.

That “simple GRBF” has been introduced above solely as a formal reinterpretation of the BMLC (16). However, one of the central paradigms of neuroinformatics says that an interpretation of mathematical models in terms of neural networks can provide heuristics for finding simple solutions of complicated algorithmic problems. Correspondingly, one may ask whether the convergence problems hampering parameter optimization in the case of the BMLC can be solved through the neural approach and whether the “simple GRBF” could even represent a GRBF classifier of the ideal type.

It is the purpose of this paper to provide answers to these questions. In particular we will prove that:

1. a coupled set of sequential stochastic learning algorithms implementing principles of neural self-organization performs a safely converging ML estimate for multivariate normal mixtures (this proof extends results of Kloppenburg and Tavan (1997));
2. an application to the network depicted in Fig. 1 solely can lead to a GRBF/CATC equivalent to Bishop’s model;
3. an application to a GRBF extended by additional reverse connections yields a GRBF of the ideal type, which self-organizes into a BMLC;
4. the resulting GRBF/BMLC is
   (a) capable of novelty detection (see e.g. Kohonen, 1984); and
   (b) computationally efficient in large-scale applications.

The paper is organized as follows. We first formally work out the reinterpretation of the BMLC as a “simple” GRBF and prove point 4b. In Section 3 we prove points 1 and 2, in Section 4 point 3, and in Section 5 point 4a. In Section 6 we illustrate the computational procedures and the performance of the GRBF/BMLC using an example from speech recognition.

2. A simple GRBF?

We have indicated above that the BMLC (16) can be interpreted as a GRBF (1), whose response functions \( a_r(x|\Theta_r) \) are defined through Eq. (18). We will now prove that a GRBF with such response functions actually becomes identical with the BMLC (16), if this GRBF obeys the following additional conditions:

1. Each neuron \( r \) of the central layer is uniquely associated to one of the classes \( k \). To formally express this...
association, we replace each neuron number \( r \) by an
index pair
\[
(s, k) = r,
\]
where \( s \) counts all neurons associated to class \( k \). Thus, the
original set \( \{ r | r = 1, \ldots, M \} \) of neuron labels is decom-
posed into disjoint subsets \( \{ (s, k) | s = 1, \ldots, m(k) \} \), whose
sizes \( m(k) \) obey
\[
\sum_{k=1}^{K} m(k) = M.
\]

2. The axonal trees \( f_{r,k} \) of all neurons associated to class \( k \)
are given by the unit vector \( e_k \in \mathbb{R}^k \), i.e.
\[
f_{r,k} = e_k \quad \text{or} \quad f_{j,(s,k)} = \delta_{jk}, \quad j = 1, \ldots, K,
\]
where \( \delta_{jk} \) is the Kronecker symbol.

For a proof we insert the decomposition (20) and the
synaptic weights (22) into the definition (1) of a GRBF,
sum over all classes \( k \), and use the well-known properties of
the Kronecker symbol to obtain the expression
\[
\hat{f}_j(x) = \sum_{s=1}^{m(j)} a_{s,j}(x|\Theta_c)
\]
for the \( j \)th component of the network output. Thus, the
activity of the \( j \)th output neuron simply is the sum of the
normalized activities of all those neurons \( (s,j) \) in the central
layer, which are associated to class \( j \). Now we insert the
definition (18) of the GRBF response functions into Eq.
(23) and compare the resulting expression with the defini-
tion (16) of the BMLC. We immediately find that the two
models are identical, i.e.
\[
\hat{f}_j(x) = \hat{P}(j|x, \Theta_c).
\]

Fig. 2 shows the special connectivity of the GRBF/BMLC
obtained in this way. Because of Eq. (22) the output weights
\( f_j \) do not represent free parameters in this model, but are
fixed, once the neurons \( r \) have been associated to the classes
\( k \). Thus, the model actually exhibits a simpler structure and
less parameters than the general GRBF shown in Fig. 1.

2.1. Statistical interpretation of the GRBF/BMLC

By construction the simplified GRBF in Fig. 2 is equiva-
 lent to a statistical standard model. Therefore, all of its
components, which we had originally introduced by associ-
ating biological correlates, have acquired a precise statis-
tical meaning. We will now sketch the corresponding
reinterpretation:

- Each Gaussian receptive field (3) represents a multivari-
ate component \( \hat{p}(x|s,j, \theta_{s,j}) \) of one of the class-condi-
tional ML mixture PDFs (15).
- The amplitudes \( \hat{P}_{s,j} \) of the receptive fields entering the
unnormalized RBF response functions (5) are products
\[
\hat{P}_{s,j} = \hat{P}(j)\hat{P}(s|j)
\]
of ML estimates \( \hat{P}(j) \) for the statistical weights of the
classes \( j \) in \( p(x) \) and of ML estimates \( \hat{P}(s|j) \) for the sta-
tistical weights of the components \( \hat{p}(x|s,j, \theta_{s,j}) \) in
the mixtures (15). Therefore, they are estimates of joint
probabilities, i.e. \( \hat{P}_{s,j} = \hat{P}(s,j) \), which are properly
normalized.
- The total activity (6) is the total ML mixture PDF (17),
i.e.
\[
A(x|\Theta_c) = \hat{p}(x|\Theta_c).
\]
Thus, for a given pattern \( x \) the value of \( A(x|\Theta_c) \) is the
likelihood that this pattern is drawn from \( \hat{p}(x|\Theta_c) \).
Similarly, for a data set \( \chi = \{ x_i | i = 1, \ldots, N \} \subset \mathbb{R}^D \) the
logarithm \( l(\chi|\Theta_c) \) of the likelihood to be generated by
\( \hat{p}(x|\Theta_c) \) is
\[
l(\chi|\Theta_c) = Nln[A(\chi|\Theta_c)]_\chi,
\]
where \( (\ldots)_\chi \) denotes the expectation value with respect to
\( \chi \). Consequently, the maximization of the log-likelihood
in statistical parameter estimation corresponds to the
maximization of the average total activity during neural
learning.
- As follows from a comparison with Bayes theorem (14),
each of the normalized response functions (18) measures the
posterior probability
\[
a_{s,j}(x|\Theta_c) = \hat{P}(s,j|x, \Theta_c),
\]
that the data point \( x \) is generated by the multivariate
component \( \hat{p}(x|s,j, \theta_{s,j}) \).

2.2. Comparison of the GRBF/BMLC with Bishop’s GRBF/
CATC

We have quoted in Section 1 that Bishop (1997) mentions
the GRBF/BMLC as a “simple” alternative, which he imme-
diately discards. He prefers a GRBF/CATC as “compu-
tationally more efficient”.

To discuss this claim, we first have to clarify how a
GRBF/CATC must be designed so as to match the
statistical interpretations suggested by Bishop. Concern-
ing the central layer, we note that Bishop’s inter-
pretation is identical to the one given above for the
GRBF/BMLC, if one drops all references to the class
structure:

- Equivalent to Eq. (26) Bishop interprets the total
activity (6) of his GRBF/CATC as a mixture model for
\( p(x) \), but instead of constructing it through a
superposition (15) of class-local mixtures (15) he
directly writes
\[ \hat{p}(x|\Theta_e) = \sum_{r=1}^{M} \hat{P}(r)\hat{p}(x|r, \theta_r). \] (29)

- As in Eq. (28) the response functions (7) are identified with posterior probabilities
\[ a_r(x|\Theta_e) = \hat{P}(r|x, \Theta_e) = \hat{P}(r|\hat{p}(x|r, \theta_r))\hat{p}(x|\Theta_e). \] (30)

Correspondingly, the parameters \( \Theta_e \) of the central layer, which cover the amplitudes \( \hat{P}_r = \hat{P}(r) \) and the parameters \( \theta_r \) of the multivariate components \( \hat{p}(x|r, \theta_r) \), should also be estimated by the maximization of the log-likelihood (27), if one wants to stick to the statistical interpretation. As a result, we arrive at the following important conclusion:

- In both cases, the parameters \( \Theta_e \) of the central layer can and, in fact, should be optimized by identical algorithms, i.e. by methods accomplishing an ML estimate for multivariate normal mixtures (15) and (29), respectively.

A corollary of this result is that, contrary to Bishop’s claim, the GRBF/BMLC can be trained with much smaller computational effort than a GRBF/CATC.

**Proof.** For a fair comparison we have to assume that the sizes \( M \) and free parameters \( \Theta_e \) of the central layer are identical for the two models. In line with Bishop’s arguments we assume for the GRBF/CATC that first the \( \Theta_e \) are estimated and, subsequently, the axonal trees \( f \). For the GRBF/BMLC solely the \( \Theta_e \) have to be estimated. Here, this set is decomposed into disjoint subsets \( \Theta_e(j) \), which parameterize the class-conditional mixture models (15) and comprise only \( m(j) \) components each. The subsets are sequentially optimized in \( K \) separate ML training processes. As database for this class-conditional training serves a disjoint decomposition of the training set \( \chi \) into class-conditional subsets \( \chi(j) \) containing \( m(j) \) samples each (the estimates \( \hat{P}_j \) additionally required for the construction of \( \hat{p}(x|\Theta_e) \) are easily obtained from the relative frequencies \( n(j)/N \)). In contrast, for a GRBF/CATC all parameters \( \Theta_e \) of all \( M \) components have to be determined at once from \( \chi \). However, all known algorithms for ML parameter estimation of mixture models require evaluations of all Gaussian components at each training data point \( x_r \). Since each of the \( M \) components of the GRBF/BMLC is trained using only \( m(j) \) data points, whereas in the case of the GRBF/CATC all \( N \) data points have to be used, the computational effort of class-conditional training is much smaller. The corresponding gain is a factor of \( 1/K \), if all classes are of equal weight. \( \square \)

Hence, the GRBF/BMLC actually is a simplified GRBF with respect to the computational effort for the optimization of \( \Theta_e \), but it is by no means “simple”, since none of the algorithmic difficulties hampering ML density estimation of mixture models (cf. the discussion of Eq. (15)) has been solved yet. Furthermore, for the complete specification of a GRBF/CATC we have to determine \( K \times M \) additional parameters \( f_{jr} \in \Theta_f \), whereas in the case of a GRBF/BMLC we have to choose only \( K \) parameters \( m(k) \). Now the questions arise, which criteria should guide these choices and how well the resulting GRBFs will perform. For an answer we have to address the problem of parameter estimation.

3. Learning rules

We have shown above that the parameters \( \Theta_e \) of the central layers of two prototypical GRBF classifiers are identical to those of associated normal mixtures and that their optimization requires a safely converging ML algorithm. Recently, such an algorithm has been suggested by Kloppenburg and Tavan (1997). This algorithm avoids the instabilities of the EM algorithm (Dempster et al., 1977) by introducing regularizing constraints and procedures of deterministic annealing. The algorithm has been originally formulated as an EM-type iteration and will now be reformulated in terms of sequential stochastic learning, since this form: (i) is more stable in actual implementations, and (ii) clarifies the connections to neural concepts. Subsequently we will discuss learning algorithms for the parameters \( \Theta_f \) of the output layer of a GRBF/CATC. As a result we will gain further insights into the limitations of the latter approach.

3.1. Learning rules for the central layer

For the central layer of the general GRBF depicted in Fig. 1, a sequential stochastic optimization of a parameter \( \theta \in \Theta_e \) has the form of an iterated update process
\[ \theta_{new} = \theta_{old} + \eta \Delta \theta(x), \] (31)
where \( \eta \) is a small learning parameter, \( x \) is an input signal randomly drawn from the training set \( \chi \), and \( \Delta \theta(x) \) is the corresponding learning rule. In the limit of a small learning parameter \( \eta \), such an update process is a stochastic formation of average values \( \langle \Delta \theta \rangle_{\chi} \), which becomes stationary at \( \langle \Delta \theta \rangle_{\chi} = 0 \). The set \( \Theta_e \) is given by
\[ \Theta_e = \{ c, \hat{\Sigma}, W_r, \hat{P}_r | r = 1,...,M \}, \] (32)
such that for each of these different parameter types a learning rule has to be given.

3.1.1. Fuzzy clustering

For the self-organization of the dendritic trees \( c \), we use a variant of the well-known fuzzy clustering rule (Kohonen,
\begin{equation}
\Delta c_i(x) = a_i(x|\Theta_c)[x - c_i] + \nu, \quad r = 1, \ldots, M \tag{33}
\end{equation}

where \( a_i(x|\Theta_c) \) is the response function (7) and \( \nu \) is a small uncorrelated Gaussian noise vector. Rule (33) represents competitive Hebbian learning and, with the use of normalized activities (cf. Eq. (8)), has been first suggested by Willshaw and Malsburg (1976) from biological considerations (see Dersch (1995) for a discussion).

For an investigation of Eq. (33), we first consider the codebook \( C = \{c_i| r = 1, \ldots, M\} \) as the set of the only free parameters in \( \Theta_c \) and fix the remaining parameters to the values \( \hat{\Sigma} = \sigma I, \ W_r = I \), and \( \hat{P}_r = 1/M \). Then the associated model density (29) is a mixture

\begin{equation}
p(x|C, \sigma) = \frac{1}{M} \sum_{r=1}^{M} \hat{p}(x|r, c_r, \sigma) \tag{34}
\end{equation}

of \( M \) univariate normal distributions

\begin{equation}
\hat{p}(x|r, c_r, \sigma) = \frac{\exp\left(-\frac{(x - c_r)^2}{2\sigma^2}\right)}{(2\pi\sigma^2)^{D/2}} \tag{35}
\end{equation}

with identical widths \( \sigma \) and weights \( 1/M \). In this univariate case the clustering rule (33) reduces exactly to that of the so-called RGF algorithm derived by Rose et al. (1990) through statistical mechanics arguments. In that view the scale parameter \( \sigma \) has the role of a temperature and, thus, enables a process of deterministic annealing (Heskes, Slijpen, & Kappen, 1993; Peterson & Anderson, 1987). During annealing, \( \sigma \) is slowly reduced from some large initial value \( \sigma_0 \) to some final value \( \sigma_f \). This hierarchical multiple-scale approach guarantees safe convergence of the codebook to a highly resolved and density oriented discretization of the input space (Dersch & Tavan, 1994a; Rose et al., 1990), which remains fuzzy as long as \( \sigma_f > 0 \).

The notion density orientation can be formulated as the property of load balance

\begin{equation}
\langle a_i(x|C, \sigma)\rangle_x = \frac{1}{M}, \quad r = 1, \ldots, M \tag{36}
\end{equation}

which implies that all cluster centers \( c_r \) represent equal amounts of data points. In the limit of very large data sets \( N \rightarrow \infty \) and dense discretizations \( M \ll N, M \rightarrow \infty \) this property holds exactly for optimal solutions\(^4\) in the RGF case (Dersch & Tavan, 1994b), whereas other neural clustering algorithms providing annealing procedures (e.g. Kohonen, 1982; Martinetz & Schulten, 1991) exhibit systematic deviations (Dersch & Tavan, 1995; Martinetz, Berkovich, & Schulten, 1993). Thus, the RGF algorithm is the method of choice, if the distribution of the \( c_r \) is supposed to reflect the statistics of the input data.

However, one can even claim: (a) that fuzzy clustering acquires a substantial statistical meaning solely within the framework of RGF, and (b) that only this approach can be complemented by a criterion for the optimal choice of the parameter \( \sigma_f \) at which the annealing process should be stopped. These statements are key results of Kloppenborg and Tavan (1997) and will be shortly explained now.

(a) RGF clustering maximizes the likelihood. For the RGF choice \( \Sigma = \sigma I, \ P_r = 1/M \) of the parameters \( \Theta_c \), we consider the log-likelihood (27) that is generated by the associated univariate normal mixture (34). Using the usual ML approach, we take derivatives of that log-likelihood \( l(x|C, \sigma) \) with respect to the \( c_r \) and find a set of necessary conditions (Duda & Hart, 1973). According to these stationarity conditions, the expectation value

\begin{equation}
0 = \langle \hat{P}_r(x|C, \sigma)[x - c_r]\rangle_x, \tag{37}
\end{equation}

where \( \hat{P}(x|C, \sigma) \) is given by Eq. (30), should vanish. Now a comparison with the clustering learning rule (33) and use of identity (30) reveals that the stationary states of RGF clustering and of ML density estimation for model (34) are identical. Thus, RGF clustering actually is sequential stochastic ML density estimation for model (34) with respect to the codebook \( C \) at a fixed value of \( \sigma \).

(b) Gaussian width \( \sigma_f \) from maximum-likelihood. In the original RGF scheme, \( \sigma \) is an annealing parameter, which exclusively serves to ensure safe convergence independent of initial conditions. Therefore, no optimal value \( \sigma_f \) can be determined at which annealing should be stopped.\(^5\)

However, if one considers RGF clustering as a tool for ML density estimation, one simply has to monitor\(^6\) the log-likelihood \( l(x|C, \sigma) \) during annealing of \( \sigma \) in order to determine an optimal \( \sigma_f \). One can show that \( l(x|C, \sigma) \) monotonously increases during \( \sigma \)-annealing for a wide class of data sets and reaches a maximum at a value \( \sigma_f \) just above the critical value \( \sigma_c \) marking the transition to the hard clustering regime.

Therefore we conclude that RGF clustering, extended by monitoring of \( l(x|C, \sigma) \) during \( \sigma \)-annealing, represents a safely converging algorithm for ML density estimation in the special case of univariate normal mixtures (34). Correspondingly we will call this optimization procedure from now on the Univar algorithm.

\(^1\) Below \( \sigma_0 > 0 \), the hard clustering limit is obtained in a sharp transition (Dersch, 1995) and the RGF algorithm reduces to \( k \)-means clustering (Lloyd, 1982; MacQueen, 1967).

\(^2\) For finite \( M \) and \( N \), Eq. (36) still applies to a very good approximation (see Dersch (1995) for details); in order to actually find such solutions, we use a variant of the RGF algorithm imposing Eq. (36) as a soft constraint during annealing (Dersch & Tavan, 1994a).

\(^3\) Below a starting value \( \sigma_0 \) is known; \( \sigma_0 \) should be slightly larger than the maximal variance \( \sigma_0 \) of the data set (Rose et al., 1990).

\(^4\) Within stochastic sequential learning the required monitoring of \( l(x|C, \sigma) \) can be achieved through calculation of a moving average of the logarithm \( \ln[l(x|\Theta_c)] \) of the total activity (cf. Eqs. (27) and (34)).
Here the question arises, how the advantages of the Univar algorithm can be transferred to the multivariate case (29) with its much larger parameter set (32). Adopting from Univar, for reasons of algorithmic stability, the choice of identical weights \( \hat{\theta}_r = 1/M \), we thus have to specify suitable learning rules for the eigenvalues \( \sigma_{i,r}^2 \), \( i = 1, \ldots, D \), and eigenvectors \( \mathbf{w}_r \) defining the local covariance matrices \( \Sigma_r \) (cf. Eq. (4)).

3.1.2. Shapes of receptive fields by self-organization of local feature detectors

To formulate such a learning rule for the eigenvectors \( \mathbf{w}_r \), we recall that they define the shapes of the receptive fields (3) of the neurons \( r \). Collecting the parameters \( \{ \mathbf{c}_r, \mathbf{w}_r \} \) into the sets \( \theta_r \), these receptive fields can be written as products of one-dimensional normal distributions

\[
\hat{p}(s_r(x|\theta_r)) = \frac{\exp(-s_r^2(x|\theta_r)/2)}{(2\pi\sigma_r^2)^{1/2}}
\]

whose arguments \( s_r(x|\theta_r) \) are the projections

\[
s_r(x|\theta_r) = \sigma_r^{-1} \mathbf{w}_r^T(x - \mathbf{c}_r)
\]

of the oriented distance \( x - \mathbf{c}_r \) between a feature combination \( x \) and a receptive field center \( \mathbf{c}_r \) on the vectors \( \sigma_r^{-1} \mathbf{w}_r \in \mathbb{R}^D \). Therefore, each of these vectors represents a local feature detector with orientation \( \mathbf{w}_r \) and weight \( \sigma_r^{-1} \).

Learning rules for the self-organization of feature detector orientations \( \mathbf{w}_r \) are well known in neuroinformatics (see, e.g. Hornik & Kuan, 1992 or Rubner & Tavan, 1989) and have been applied, e.g. to model receptive fields in vertebrate visual cortices (Rubner, 1989; Rubner & Schulten, 1990; Rubner, Schulten, & Tavan, 1990). The rules have been shown to perform a principal component analysis (PCA) on the global distribution of the input data. As a result, the \( \mathbf{w}_r \) converge to the eigenvectors of the associated covariance matrix.

Thus, we adopt a variant of these rules for the orientations \( \mathbf{w}_r \) of the local feature detectors. Using Eq. (31) we first update the \( \mathbf{w}_r \) by applying the rule

\[
\Delta \mathbf{w}_r(x) = a_r(x|\Theta_c)s_r(x|\theta_r)(x - \mathbf{c}_r)/\sigma_r, \quad i = 1, \ldots, D'.
\]

and subsequently restore orthonormalization of the \( \mathbf{w}_r \) by the well-known Schmidt procedure (Hornik & Kuan, 1992). The first step of this procedure is of the pure Hebbian type if one conceives the product \( a_r(x|\Theta_c)s_r(x|\theta_r)/\sigma_r \) in Eq. (39) as the effective activity of a postsynaptic neuron \( (r,i) \) associated to the feature detector \( \sigma_r^{-1} \mathbf{w}_r \) and the components of \( x - \mathbf{c}_r \) as the activities of \( D' \) presynaptic neurons.\(^7\) The second step (orthonormalization) introduces a competition required for self-organization (cf. Hillermeier et al. (1994)).

\[\text{For an investigation of Eq. (39) we identify by Eq. (30) the activation functions } a_r(x|\Theta_c) \text{ with the posterior probabilities } \hat{P}(r|x, \Theta_c) \text{ and define local expectation values by}
\]

\[
\langle g(x) \rangle_{r,\theta_r} = \frac{\langle \hat{P}(r|x, \Theta_c) q(x) \rangle_x}{\langle \hat{P}(r|x, \Theta_c) \rangle_x}
\]

Then we define local \( D \times D \) covariance matrices by

\[
C_{r,\theta_r} = \langle (x - \mathbf{c}_r)(x - \mathbf{c}_r)^T \rangle_{r,\theta_r},
\]

where the centers \( \mathbf{c}_r \) are assumed to be the local expectation values

\[
\mathbf{c}_r = \langle x \rangle_{r,\theta_r}
\]

Inserting the definition (40) of the local expectation values, we see that Eq. (42) is a trivial reformulation of the ML stationarity conditions (37) such that it holds for \( \mathbf{c}_r \), resulting from Univar. Therefore, Eq. (41) also is an exact expression.

Because of the reiterated renormalizations of the vectors \( \mathbf{w}_r \) during learning according to Eq. (39) the associated stationarity conditions are \( \langle \Delta \mathbf{w}_r \rangle_{r,\theta_r} = \lambda_r \mathbf{w}_r \), where the \( \lambda_r \) are scalar parameters. Inserting definition (38) into Eq. (39) in order to rewrite \( \Delta \mathbf{w}_r \) we find after a little algebra the equivalent condition

\[
C_{r,\theta_r} \mathbf{w}_r = \lambda_r \mathbf{w}_r, \quad i = 1, \ldots, D',
\]

where the new parameters are \( \lambda_r = \lambda_r \sigma_r^2 / \langle a_r(x|\Theta_c) \rangle_x \).

Thus, in contrast to the original model (Hornik & Kuan, 1992; Rubner & Tavan, 1989), which diagonalizes the global covariance matrix of the input data, the modified rule (39) becomes stationary as soon as the \( \mathbf{w}_r \) represent eigenvectors of the local covariance matrices (41). Since the two algorithms are otherwise identical, however, the fast convergence of the original algorithm is inherited by the modified one. For sufficiently slow annealing schedules the fast convergence of rule (39) guarantees that the \( \mathbf{w}_r \) closely approximate the eigenvectors of the local covariance matrices (41) at all stages of optimization. Note that not all eigenvectors are obtained, if one chooses the number \( D' \) of feature detectors in Eq. (39) smaller than the dimension \( D \) of the input space. Then the resulting \( \mathbf{w}_r \) correspond to the \( D' \) largest eigenvalues \( \lambda_r \) (Hornik & Kuan, 1992; Rubner & Tavan, 1989).

The above discussion has demonstrated that rule (39) performs a local principal component analysis (LPCA), which may be partial if \( D' < D \). As a result, the projections \( s_r(x|\theta_r) \) become locally decorrelated, implying that

\[
\langle s_r(x|\theta_r)s_{r'}(x|\theta_{r'}) \rangle_{r,\theta_r} = \frac{\lambda_{r'} \delta_{rr'}}{\sigma_r\sigma_{r'}}
\]

where \( \delta_{rr} \) is the Kronecker symbol. This expression follows from definitions (38) and (40) upon the assumption that the \( \mathbf{w}_r \) solve the eigenvalue problem (43), which was justified above.

\(^7\) A network model, which explicitly adds such neurons and their connections to the GRBF structure depicted in Fig. 1, would be very complex and will not be discussed here.
3.1.3. Functional balance of local feature detectors

Next we turn to a learning rule for the eigenvalues $\sigma^2_{r}$ defining the weights $\sigma^{-1}_{r}$ of the local feature detectors. Using the projections defined by Eq. (38) we suggest the expression

$$\Delta \sigma^2_{r} = a_{r}(\mathbf{x}, \theta_{r}) \left( \sigma^2_{r}[\mathbf{x}|\theta_{r}] - 1 \right) + \mu \frac{\sigma^2_{r} - \sigma^2_{0}}{\langle a_{r}(\mathbf{x}|\theta_{r}) \rangle_{x}} \tag{45}$$

where $\mu$ is a positive coupling parameter and $\sigma_{r}$ is the ML Gaussian width of a corresponding univariate mixture model (34). We assume here, that $\sigma_{r}$ and an ML codebook $C_{r}$ have been determined by Univar as starting values for the joint optimization of the multivariate parameters $c_{r}$, $w_{ir}$, and $\sigma^2_{r}$ through Eqs. (33), (39) and (45), respectively.

Reduction of the coupling parameter $\mu$ in Eq. (45) from a large initial value $\mu_{0}$ to a vanishing final value $\mu_{f} = 0$ enables a smooth transition from a Univar training to a joint optimization. (a) At $\mu = \mu_{0}$ the first term in Eq. (45) can be neglected and the $\sigma^2_{r}$ are strongly bound to the value $\sigma^2_{f}$ by the second term, i.e. one has $\sigma^2_{r} = \sigma^2_{f}$. Therefore, one has $\Sigma_{0} = \sigma_{f} \mathbf{I}$ to a very good approximation, and by Eq. (4) one even has $\Sigma_{0} = \sigma_{f} \mathbf{I}$ for arbitrary orthonormal matrices $W_{r}$. Thus, the multivariate Gaussians (3) become univariate in that limit and the clustering (33) of the $c_{r}$ becomes effectively decoupled from the local PCA (33). (b) At $\mu = 0$ all three rules are strongly coupled. Then the stationarity condition of rule (45) is given by

$$\langle \sigma^2_{r}[\mathbf{x}|\theta_{r}] \rangle_{x} = 1, \tag{46}$$

where we have used the local expectation value (40) again for short-cut notation. We call property (46) of the projections (38) on the local feature detectors functional balance.\footnote{Condition (46) introduces a local Mahalanobis metric within each receptive field; it amounts to the biologically appealing requirement that all local feature detectors are of equal functional importance.}

Now a comparison of Eq. (46) with Eq. (44) shows that the parameters $\sigma^2_{r}$ are equal to the eigenvalues $\lambda_{r}^{\prime}$ of the local covariance matrices (41) if all three rules (33), (39) and (45) have become stationary. Thus, their stationary point is characterized by Eq. (36) and by

$$\Sigma_{r} = C_{r}.\theta_{r}, \tag{47}$$

where we have reconstructed the covariance matrices $\Sigma_{r}$ characterizing the components of the mixture density $\hat{p}(\mathbf{x} | \theta_{r})$ from the $w_{ir}$ and $\sigma^2_{r}$ by inverting the spectral decomposition (4). As shown by Kloppenburg and Tavan (1997), these are just the stationarity conditions of ML density estimation for multivariate normal mixtures (29) or (15) with identical weights $P_{r} = 1/M$.

Hence, we have proven that rules (33), (39) and (45) are sequential stochastic versions of the EM algorithm of Kloppenburg and Tavan (1997).\footnote{For an instructive sample application, we refer to this paper.} We will call the learning process, which smoothly switches from the results of Univar to a joint optimization of the multivariate parameters by annealing of the coupling parameter $\mu$, the Multivar algorithm. As before the progress of the optimization can be monitored by the log-likelihood $l(\chi, \theta_{r})$.

3.1.3.1. Partial LPCA and $\sigma_{r}$-annealing. In high-dimensional applications of Multivar, only a few local feature detectors ($D' \ll D$) should be chosen for reasons of algorithmic stability. In this case the components of the mixture are multivariate only into the directions with the $D'$ largest variances $\sigma^2_{r}$ and are spherical into the directions of the remaining smaller variances. This partial LPCA approach corresponds to a modified decomposition of the covariance matrices

$$\Sigma_{r}^{-1} = W_{r} \Sigma_{r}^{-1} W_{r}^{\prime} + \frac{1}{\sigma_{r}^{2}}(I - W_{r} W_{r}^{\prime}), \tag{48}$$

with the $D \times D'$ matrices $W_{r} = (w_{1r}, \ldots, w_{Dr'})$ of eigenvectors and the diagonal $D' \times D'$ matrices $\Sigma_{r}^{\prime} = \text{diag}(\sigma^2_{1r}, \ldots, \sigma^2_{Dr'})$ of the largest eigenvalues. In Eq. (48) the variance of the spherical part is kept at the value $\sigma_{r}$ obtained by Univar. After Multivar one may further maximize the log-likelihood by additional annealing of this parameter (Multivar-$\sigma$).

3.1.3.2. Discussion of stability. Apart from quoting studies, which prove the stability of Univar annealing (Dersch & Tavan, 1994a; Rose et al., 1990), the fast convergence of Eq. (39) (Hornik & Kuan, 1992; Rubner & Tavan, 1989), and the regularizing role of the soft constraint in Eq. (45) (Kloppenburg & Tavan, 1997), we have not yet clarified why our multiple stage annealing procedure and our sequential stochastic learning algorithms can avoid the stability problems hampering previous EM-type algorithms. Owing to the lack of space, a sketch of two additional reasons must suffice here:

1. A fuzzy partitioning of the input space by the activation functions (7) is maintained at all annealing stages. Hard clustering is known to represent an NP-complete optimization problem, which is prone to get stuck in local extrema. We avoid this limiting case and stick to fuzzy partitions during Univar and Multivar. In high-dimensional cases this is achieved for Multivar by the choice $D' \ll D$; after Univar the optimal variance $\sigma^2_{r}$ is the mean of all eigenvalues of the local covariance matrices (41). Thus, the $D'$ variances $\sigma^2_{r}$ obtained through LPCA are larger than $\sigma^2_{r}$ and, therefore, the degree of fuzziness of the partition will increase during Multivar.

2. Property (36) of load balance is always maintained and enforced. In Univar and Multivar property (36) of load balance derives from our fixed choice $P = 1/M$ for the statistical weights. This constraint forces the remaining parameters to adjust in such a way that load balance is maintained. Without this “biological” competition principle the system would inevitably run into singular
solutions. This claim immediately follows from the EM learning rule (Duda & Hart, 1973) for the \( \hat{P}_r \), which for a stochastic update process is \( \Delta \hat{P}_r = a_r(x(\Theta_e)) - \hat{P}_r \). The rule describes the stochastic formation of an average of the activity and becomes stationary at \( \hat{P}_r = (a_r(x(\Theta_e)))x \).

If during learning a node \( r \) is weakly active for a while, its weight \( \hat{P}_r \) will decay and, hence, will continue to decay due to the competition with the other nodes induced by the global normalization (for examples see Duda and Hart (1973)). Therefore one should not try to adjust the weights \( \hat{P}_r \) to the local statistics by that rule during optimization of \( \Theta_e \); however, it may be tried afterwards with \( \Theta_e \) kept fixed.

3.2. Learning of the output weights

For the construction of a GRBF/CATC, which is statistically sound in the Bishop (1997) sense, a suitable learning rule for the output weights \( f \) remains to be given. Because the output layer represents a simple linear perceptron (cf. Eq. (1) and Fig. 1), Bishop (1997), in line with Moody and Darken (1989), suggests the so-called \( \Delta \)-rule, which minimizes least-square deviations. For a stochastic update process

\[
f_{r}^{\text{new}} = f_{r}^{\text{old}} + \eta \Delta f_r(x, y), \quad r = 1, \ldots, M, \tag{49}
\]

in which training samples \((x, y)\) are randomly drawn from the training set \( \mathcal{I} \), the \( \Delta \)-rule reads

\[
\Delta f_r(x, y) = a_r(x)[y - \hat{f}(x)]. \tag{50}
\]

Although mathematically simple and safely converging, this rule cannot be considered as a model for learning in cortical tissue as it correlates the activities \( a_r(x) \) in the central layer with an error signal \( y - \hat{f}(x) \) and, hence, contains non-local information (cf. Crick (1989)). A local and therefore biologically plausible learning rule can be formulated as

\[
\Delta f_r(x, y) = a_r(x)[y - f_r]. \tag{51}
\]

This local rule also represents competitive Hebbian learning (cf. Eq. (33)). To describe synaptic growth it correlates the activities \( a_r(x) \) in the central layer with activities \( y_j \) coding teaching signals in the output layer and accounts for the biologically required limitation of that growth (Hillermeier et al., 1994) by the decay terms \(-a_r(x)f_{jr}\).

Rule (51) does not minimize a least-square error criterion. As one can easily see, it coincides with Eq. (50) only in the limit \( \sigma \to 0 \). But in the context of classification it has distinct advantages:

1. The stationarity conditions \( \langle \Delta f_r \rangle_x = 0 \) of Eq. (51) are

\[
f_{jr} = (y_j)_r, \quad j = 1, \ldots, K, \tag{52}
\]

where we have used definition (40) of the local expectation values and identity (30). Thus, the components \( f_{jr} \) of the stationary axonal tree vectors \( f \), are obtained by taking simple averages.

2. Recalling that in classification the teaching signals \( y \) are chosen from the set of unit vectors in \( \mathbb{R}^K \) and, thus, have the normalization property \( \sum_{j=1}^{K} y_j = 1 \), one finds that the stationary weight vectors \( f \) are also normalized

\[
\sum_{j=1}^{K} f_{jr} = 1, \quad r = 1, \ldots, M. \tag{53}
\]

Since the \( f_{jr} \) are quotients of averages of positive quantities \( (a_r(x), y_j \geq 0) \), they are also positive and, hence, may be interpreted as conditional probabilities \( \hat{P}(j|r) \) that the neuron \( r \) should be associated to class \( j \). Note that \( f_{jr} \) obtained by the \( \Delta \)-rule (50) do not have these properties and, correspondingly, do not allow such an interpretation (for an ad hoc procedure to repair these deficiencies see Miller & Uyar (1998)).

3. Using the statistical interpretation (30) of the activation functions \( a_r(x) \), output (1) of the GRBF/CATC may now be written as

\[
\hat{f}_j(x) = \sum_{r=1}^{M} \hat{P}(j|r)\hat{P}(r|x, \Theta_e), \quad j = 1, \ldots, M. \tag{54}
\]

This expression shows that the activities \( \hat{f}_j(x) \) of the output neurons are positive and normalized to 1, such that they actually can be interpreted as approximate posterior probabilities \( \hat{P}(j|x) \) as required by Eq. (13) for a GRBF classifier.

However, this interpretation contains the same type of uncontrolled approximation, which has been noted earlier in connection with Bishop’s Ansatz (19). In the case at hand, it derives from the fact that the output weights \( \hat{P}(j|r) = f_{jr} \) (54) are solely conditional to \( r \) but not also conditional to \( x \); the latter should be the case as follows from the exact decomposition of the joint probability \( P(r, j|x, \Theta_e) = \hat{P}(j|r, x)\hat{P}(r|x, \Theta_e) \), a summation of this identity over \( r \) and a comparison with Eq. (54). In addition, we immediately see that this defect cannot be repaired by choosing an alternative learning rule for the \( f \), since no such rule can convert these constants into functions of \( x \). Therefore, this defect is a common property of all GRBF/CATC and point 2 raised at the end of Section 1 is proven.

So the question remains, whether this defect can be removed through a joint optimization of all GRBF parameters, i.e. by a GRBF of the ideal type, how such a GRBF must be structured and trained and how it is related to the statistically sound GRBF/BMLC. These questions will be addressed in Section 4.

4. Self-organization of a GRBF/BMLC

A GRBF/CATC differs from a GRBF of the ideal type by the decoupling of the training procedures for the parameters
\( \Theta_c \) of the central layer and \( \Theta_f \) of the output layer, respectively. To obtain a clue as to how one can remedy this drawback, compare the update rules (31) and (49). Rule (31) for \( \Theta_c \) solely refers to the input signals \( x \in \chi \). whereas rule (49) for \( \Theta_f \) additionally contains the teaching signals \( y \).

4.1. Clustering of the joint density

To enable a training of \( \Theta_c \), which includes the teaching signals \( y \), it seems natural to introduce additional reverse connections \( f' \), from the output layer to the central layer as shown in Fig. 3. As a result, a combined signal \( (x,y) \in R^{D+K} \), reaching concomitantly the top and bottom layers of the net, can induce activations \( a_i(x,y|\Theta_c') \) within the central layer by means of the dendritic connections \( (c_i,f_i') \in R^{D+K} \). Here we have added the \( f' \) to \( \Theta_c' \) forming the extended parameter set \( \Theta_c' \).

For the mathematical description of the extended activations \( a_i(x,y|\Theta_c') \) we use a normalized response function analogous to Eq. (7) normalized by a correspondingly modified total activity \( A(x,y|\Theta_c') \). To ensure load balance we assume that the associated initial response functions \( a_i(x,y|\widehat{P}_r,\theta_r,\theta_r') \) have identical weights \( \widehat{P}_r = 1/M \). Without loss of generality and in order to simplify notation we restrict the following discussion to the univariate case.

Hence, the normalizing total activity is the mixture PDF

\[
A(x,y|\Theta_c') = \frac{1}{M} \sum_{i=1}^{M} \hat{p}(x|r,\theta_r)\hat{p}(y|r,\theta_r')
\]  

composed of products of univariate normal distributions \( \hat{p}(x|r,\theta_r) \) and \( \hat{p}(y|r,\theta_r') \) characterized by the parameters \( \theta_r = \{c_r,\sigma\} \) and \( \theta_r' = \{f',\rho\} \), respectively. The former functions are given by Eq. (35) and the latter by

\[
\hat{p}(y|r,\theta_r') = \frac{\exp[-(y-f')^2/2\rho^2]}{(2\pi \rho^2)^{K/2}}.
\]  

The connections \( (c_i,f_i') \) can then self-organize upon stochas-

tic presentation of signals \( (x, y) \) using a correspondingly modified clustering rule (cf. Eq. (3))

\[
\Delta c_i(x,y) = a_i(x,y|\Theta_c')[x - c_i] + v_i,
\]

\[
\Delta f_i(x,y) = a_i(x,y|\Theta_c')[y - f_i'] + v_i',
\]  

where the parameters \( \Theta_c' \) cover the set \( \{c_i,\sigma,f_i',\rho|r = 1,\ldots,M\} \).

4.2. Univariate annealing for the extended GRBF

For the joint clustering (57) of classification data \( (x,y) \in \mathcal{Z} \), the associated Univariate annealing process must be slightly changed, in order to enable a convergence of the model density (55) towards a high-quality ML estimate. The reasons for the required change may be seen as follows:

Consider the data set \( \mathcal{Z} \) as a sample drawn from a joint PDF \( p(x,y) \), which then is the PDF to be modeled by the normal mixture (55). Recall now that the association of a data vector \( x \) to a class \( j \) is coded in \( \mathcal{Z} \) by \( y = e_j \in R^K \). Therefore, the PDF \( p(y) \) of the teaching signals \( y \) is a sum of Dirac \( \delta \)-distributions

\[
p(y) = \sum_{j=1}^{K} P_j \delta(y - e_j) \]  

weighted by the prior probabilities \( P_j \) of the classes \( j \). Using \( p(x,y) = p(y)p(x|y) \) and Eq. (58) one finds for the joint PDF the decomposition

\[
p(x,y) = \sum_{j=1}^{K} P_j p(x,y|j) \]  

into class-local PDFs

\[
p(x,y|j) = \delta(y - e_j)p(x|j),
\]  

which have vanishing variances in the \( y \)-directions and are centered at the points \( e_j \) in the \( y \)-subspace. Therefore, an ML model (55) for the joint PDF (59) must also be composed of components \( a_i(x,y|\hat{P}_r,\theta_r,\theta_r') \) with these properties. A
comparison with Eq. (56) shows that this is the case if 
\[ \rho = 0 \quad \text{and} \quad \mathbf{f}'_r \in \{ \mathbf{e}_j \mid j = 1, \ldots, K \}, \]  
(61)
where we have used the fact that the normal distributions (56) become the \( \delta \)-distributions \( \delta(x - \mathbf{f}'_r) \) for \( \rho \to 0 \). The identification of the optimal value \( \rho = 0 \) by Eq. (61) suggests that one should use the variances \( \sigma \) and \( \rho \) as separate annealing parameters in \textit{Univar} and that the annealing of \( \rho \) has to be continued until the hard clustering limit is reached (\( \rho < \rho_c \)). At \( \rho < \rho_c \), the \textit{Univar} algorithm reliably renders the ML solution (61) for the \( \mathbf{f}'_r \). This has been demonstrated (Dersch, 1995) by a series of applications to data distributions exhibiting \( \delta \)-type cluster structures like the PDFs (58) and (59). As a result each neuron \( r \) of the central layer becomes uniquely associated to one of the classes \( j \) as formally expressed by Eqs. (20) and (21). Consequently, \textit{Univar} also renders an estimate for the GRBF/BMLC parameters \( m(j) \), for which no training procedure had been given yet. One generally finds (Dersch, 1995) that the numbers \( m(j) \) of centers allocated to the classes \( j \) are proportional to the prior statistical weights \( P(j) \) of these classes, i.e.
\[ \frac{m(j)}{M} = P(j). \]  
(62)
This statistically balanced distribution of the \( M \) centers \( \mathbf{f}'_r \) to the \( K \) \( \delta \)-distributions \( j \) of \( \mathcal{Z} \) is an immediate consequence of the property of load balance (36), which distinguishes the \textit{Univar} ML estimates from the results of other clustering procedures. Thus, \textit{Univar} also renders the ML estimates \( \hat{P}(j) = m(j)/M \) for the statistical weights \( P(j) \).

Eqs. (61) and (62) with the above association \( (s,j) \mapsto r \) of the neurons \( r \) to the classes \( j \) completely specify the \textit{Univar} ML estimates concerning the \( \mathbf{y} \)-directions at \( \rho < \rho_c \). In particular the codebook vectors are
\[ (c_{s,j}, \mathbf{f}'_{s,j}) = (c_{s,j}, e_j) \]  
(63)
As a result, the extended activation functions \( a_{s,j}(\mathbf{x}, \mathbf{y}|\Theta'_r) \) become much simpler at \( \rho < \rho_c \). To formulate this simplification consider the total activity (55) in the limit \( \rho \to 0 \). With Eqs. (59), (60) and (62) and with the above analysis it becomes decomposed
\[ A(\mathbf{x}, \mathbf{y}|\Theta'_r) = \hat{p}(\mathbf{x}, \mathbf{y}|\Theta'_r) = \sum_{j=1}^{K} P(j) \delta(\mathbf{y} - \mathbf{e}_j) A_j(\mathbf{x}|\Theta,(j)) \]  
(64)
into class-conditional activation functions
\[ A_j(\mathbf{x}|\Theta,(j)) = \hat{p}(\mathbf{x}|j|, \Theta,(j)) = \frac{1}{m(j)} \sum_{i=1}^{m(j)} \hat{p}(\mathbf{x}|s_i, j, \theta_{s_i}) \]  
(65)
for the neural groups associated to the classes \( j \). Returning to the multivariate case for these functions \( A_j(\mathbf{x}|\Theta,(j)) \), we note that the sets \( \Theta,(j) \) solely comprise the parameters \( \theta_{s,j} = \{ c_{s,j}, \Sigma_{s,j}, W_{s,j} \} \), i.e. that the components of the associated model densities \( \hat{p}(\mathbf{x}|j|, \Theta,(j)) \) have identical statistical weights \( P(s|j) = 1/m(j) \). Thus, the group activities (65) are identical with the mixture models (15) of a BMLC, if the latter is constructed using \textit{Multivar}. Integrating Eq. (64) over \( \mathbf{y} \) and using the ML estimate \( \hat{P}(j) = m(j)/M \) one finds that the PDF of all patterns \( \mathbf{x} \) is the weighted sum
\[ \hat{p}(\mathbf{x}|\Theta_r) = \sum_{j=1}^{K} P(j) \hat{p}(\mathbf{x}|j, \Theta,(j)) \]  
(66)
of the class-conditional PDFs. Now we can state the following theorem:

**Theorem.** As \( \rho \to 0 \) the extended activation functions \( a_{s,j}(\mathbf{x}, \mathbf{y}|\Theta'_r) \) converge to the limits
\[ a_{s,j}(\mathbf{x}, \mathbf{y}|\Theta'_r) = \begin{cases} \hat{P}(j) \hat{p}(\mathbf{x}|s, j, \theta_{s,j}) & \text{if } \mathbf{y} = \mathbf{e}_j \\ 0 & \text{if } \mathbf{y} = \mathbf{e}_j, k \neq j \\ \hat{P}(j) \hat{p}(\mathbf{x}|s, j, \theta_{s,j}) & \text{if } \mathbf{y} = \mathbf{0}, \end{cases} \]  
(67)
where we have used Eqs. (65) and (66) and have omitted the threshold behavior from Eq. (7) for the sake of simplicity.

According to Eq. (67) there are two cases: (a) A pattern vector \( \mathbf{x} \) is accompanied by a non-zero teaching signal \( \mathbf{y} \in \{ \mathbf{e}_j \} \) as is the case during learning. For teaching signals \( \mathbf{y} = \mathbf{e}_j \), the neuron \( s_j \) belongs, its extended activation function \( a_{s,j}(\mathbf{x}, \mathbf{y}|\Theta'_r) \) reduces to a form, which is required for a Multivar training of a class-conditional mixture (15) using a class-conditional training set \( \chi(j) \) (cf. Sections 2 and 3). This is apparent from the fact that here the normalizing activity is the class-conditional group activity (65). For teaching signals from other classes, the response vanishes and, thus, the signal is ignored.

(b) No teaching signal is given \( \mathbf{y} = \mathbf{0} \) as is the case during classification of unlabeled patterns. Then the extended activation function \( a_{s,j}(\mathbf{x}, \mathbf{y}|\Theta'_r) \) exactly reduces to the response function (18) of a GRBF/BMLC.

**Proof.** (a) By Eq. (61) the reverse connections of the neurons \( r \mapsto (s,j) \) associated to the class \( j \) converge to the values \( \mathbf{f}'_{s,j} = \mathbf{e}_j \) at sufficiently small \( \rho < \rho_c \). Owing to Eq. (56) the neurons \( (s,j) \) then are sharply tuned to the teaching signal \( \mathbf{y} = \mathbf{e}_j \). Upon presentation of this signal, only these neurons will acquire substantial initial activities \( \hat{d}_{s,j}(\mathbf{x}, \mathbf{y}) \approx \int \delta(\mathbf{y} - \mathbf{e}_j) \hat{p}(\mathbf{x}|s, j, \theta_{s,j}) \), which add up to the group activity \( m(j)/M \delta(\mathbf{y} - \mathbf{e}_j) \). Upon formation of the normalized response the common factor \( (1/M)\delta(\mathbf{y} - \mathbf{e}_j) \) cancels. As a result the normalizing activity becomes the class-conditional activity (65) for the neurons \( (s,j) \). The initial and group
activities and, due to the threshold in Eq. (7), also the normalized activities of the other neurons \((s, k)\) associated to the remaining classes \(k\) vanish.

(b) According to Eqs. (55) and (56) all neurons \(r\) of the central layer have the initial response \(1/M\hat{p}(x_{r}, \theta_{c})\hat{p}(0|\theta_c')\) with the common damping factor \(\hat{p}(0|\theta_c') = \exp(-1/2\rho^2)/(2\pi\rho^2)^{1/2}\). Upon formation of the normalized response this damping factor cancels and the globally normalized activation function (18) of the feed-forward GRBF/BMLC is recovered, i.e. \(a_r(x, \theta_c') = a_r(x|\theta_c) = a_{s,j}(x|\theta_c)\).

To summarize the results of the modified Univar annealing we state:

- The self-organization of the reverse connections and the shrinking widths \(\rho\) of the receptive fields concerning the \(y\)-directions generate a sharp tuning of a neuron group \((s, j)|s = 1, \ldots, m(j)|\) to the teaching signal \(e_j\) labeling class \(j\). During training with data \((x, y) \in \mathcal{I}\) the tuning automatically selects this group for ML optimization of the class-conditional model density (65). According to Eq. (62) the size \(m(j)\) of that group reflects the statistical weight of class \(j\).
- If a teaching signal is lacking, all neurons of the central layer respond to a pattern \(x\) and their normalized activation function (67) is the activation function (18) of a GRBF/BMLC.

4.3. Network output and discussion of results

In order to determine the output (1) of the extended network depicted in Fig. 3, we have to consider the training of the axonal trees \(f_{r}\). As in the case of the GRBF/CATC, we choose the local learning rule (51) for the \(f_{r}\), replacing\(^{11}\) the feed-forward activation functions \(a_r(x|\theta_c)\) by the extended activation functions \(a_r(x, y|\theta_c')\). Then the local rule (51) is identical with the clustering rule (57) for the reverse connections \(f'_{s,j}\). Therefore, the \(f_{s,j}\) and the \(f'_{r}\) converge during \(\rho\)-annealing to the same values, which by Eq. (63) are given by\(^{12}\)

\[
f_{s,j} = f'_{s,j} = e_j. \tag{68}
\]

Thus, during joint parameter optimization the axonal trees of the extended network converge to the values given in Eq. (22) for the axonal trees of a GRBF/BMLC.

As a result, the feed-forward structure of the extended network in Fig. 3 simplifies to that of the GRBF/BMLC depicted in Fig. 2 by self-organization. If one wants to change Fig. 2 into a sketch of the complete extended network, one simply has to add arrows pointing upwards to the connections between the central neurons \(r\) and the output neurons \(j\) in order to indicate also the \(f'_{s,j}\).

Using the arguments, which in Section 2 lead to Eq. (24) for the output \(\hat{f}_j(x)\) of a GRBF/BMLC, one immediately derives from Eqs. (67) and (68) the final result:

- For an unlabeled pattern \(x\) the output of a converged extended network is the set of ML Bayesian posterior probabilities \(\hat{P}(j|x, \theta_c)\) as given by Eq. (24). Therefore, concerning pattern classification, the self-organizing extended network is identical to the GRBF/BMLC and, due to the joint optimization of all parameters, is of the ideal type. This completes the proof of point 3 raised in Section 1.

Concerning training, the two models differ: transferring the arguments of the proof for the corollary in Section 2 one shows that the GRBF/BMLC is computationally more efficient than the extended GRBF and, therefore, should be used in practical applications. Thus, the extended model is of purely theoretical interest, as it proves that the seemingly artificial GRBF/BMLC, which had been introduced by purely statistical reasoning, is a “natural” neural network model: (a) its simplified feed-forward structure can arise by biologically plausible processes\(^{13}\) of self-organization, and (b) the reciprocal connectivities required for self-organization are abundant in cortical tissue (cf. Tavan et al. (1990)). In contrast, the feed-forward architecture of a GRBF/CATC is neither “natural” nor is that model statistically sound.

5. Novelty detection

Up to now we have not yet checked how the threshold behavior of the neurons \(r\) expressed by the cut-off parameter \(\epsilon\) in the activation function (7) affects the classification properties of the GRBF/BMLC.

For a discussion assume that a pattern \(x\) is presented to the GRBF for classification, which is very different from any other pattern contained in the training set \(\mathcal{I}\). Such a situation is likely to occur in real-world applications and may be due, e.g. to the confrontation of the classifier with unknown or new objects. Such a new pattern is likely to elicit a very small total activity \(A(x|\theta_c)\) within the central layer, because \(A(x|\theta_c)\) is the likelihood that \(x\) has been drawn from the model density \(\hat{p}(x|\theta_c)\) of the training set \(\mathcal{I}\). If \(A(x|\theta_c)\) happens to be smaller than the cut-off parameter \(\epsilon\), then none of the neurons \(r\) will acquire a non-vanishing activation and by Eq. (23) the output of the net will also vanish, i.e.

\[
A(x|\theta_c) < \epsilon \iff \hat{f}(x) = 0. \tag{69}
\]

Conversely we can consider a vanishing response of the

\(^{11}\) The replacement is required, if one wants to maintain the Hebbian interpretation of Eq. (51).

\(^{12}\) Formally, \(f_{s,j} = e_j\) follows from Eq. (52) using Eqs. (63) and (67).

\(^{13}\) Due to lack of space, biological analogs for the annealing processes cannot be given here.
classifier as an indication that the presented pattern \( x \) must be sorted into the complementary class of unknown objects. Therefore, the cut-off can prevent the classification of unknown patterns in terms of the predefined categories and, correspondingly allows novelty detection.

5.1. Statistical analysis

Although the above scheme for a binary classification of all patterns presented to a GRBF into the two complementary categories \( C_0 \) and \( C_1 \) of known and unknown patterns is qualitatively clear, a precise statistical meaning remains to be given.

For this purpose we first recall that the result \( \hat{p}(x|\Theta_c) = A(x|\Theta_c) \) of our maximum-likelihood approach is an estimate \( \hat{p}(x|C_0) = \hat{p}(x|\Theta_c) \) of the known patterns. Concerning the distribution of all other patterns we do not have any information. The distribution accounting for such a lack of information is the uniform distribution. Thus, we may write a best estimate for the distribution \( p(x|C_1) \) of the unknown patterns as

\[
\hat{p}(x|C_1) = V^{-1},
\]

where the constant \( V^{-1} \) serves to normalize this PDF. Since the volume \( V \) of the corresponding feature space is unknown, this constant is also unknown.

Suppose now that we want to decide for a given pattern \( x \) whether it belongs to the class \( C_1 \). According to the Bayes criterion (cf. Section 1) one should classify \( x \) into \( C_1 \) if

\[
\hat{P}_0\hat{p}(x|C_0) < (1 - \hat{P}_0)\hat{p}(x|C_1).
\]

Here, the statistical weight \( \hat{P}_0 \) of the class \( C_0 \) within the distribution of all patterns is again an unknown constant. Using Eqs. (70) and (71), and combining all unknown constants in Eq. (72) into the new constant \( \epsilon = (1 - \hat{P}_0)\hat{P}_0 V \) we find the decision rule

\[
\hat{p}(x|\Theta_c) < \epsilon \Rightarrow x \rightarrow C_1
\]

for the rejection of patterns as unknown. Identification of the model PDF with the total activity \( A(x|\Theta_c) \) now shows that the criterion (73) is identical to Eq. (69). Thus, Eq. (69) provides a Bayesian decision between the class \( C_0 \) of known patterns distributed according to \( \hat{p}(x|\Theta_c) \) and the class \( C_1 \) of uniformly distributed unknown patterns. The decision depends, of course, on the parameter \( \epsilon \).

5.2. Probability of erroneous rejection

To determine the quality of that simple binary classifier one must know the probability \( P(\text{err}_1|\epsilon) = P(x|\Theta_c) < \epsilon | x \in C_0) \) that a pattern from \( C_0 \) is falsely classified through Eq. (73) into \( C_1 \). To the extent that the density estimate \( \hat{p}(x|\Theta_c) \) is correct we have

\[
P(\text{err}_1|\epsilon) = P[l(x|\Theta_c) \leq \ln \epsilon | x \in C_0],
\]

where \( l(x|\Theta_c) \) is the log-likelihood \( \ln[\hat{p}(x|\Theta_c)] \) and where we have used Eq. (70) as well as the fact that the logarithm is a monotonous function. In practice one can calculate \( l(x|\Theta_c) \) after training of \( \hat{p}(x|\Theta_c) \) for every training pattern \( x \in \chi \). Considering \( l(x|\Theta_c) \) as a random variable \( l \), one calculates an estimate for its cumulative distribution function \( P(l \leq \ln \epsilon) \) by coarse-grained statistics. For a given level of significance \( \alpha \) one then determines a value \( \epsilon(\alpha) \) in such a way that \( P(\text{err}_1|\epsilon) = \alpha \).

6. A sample application

In order to show how our GRBF/BMLC performs real-world pattern recognition, we present an example from the...
field of speech recognition. In particular, we will illustrate how the quality of the classifier increases during the sequential Univar and Multivar optimization steps.

6.1. Phoneme classification for TIMIT data

For the example we have selected the problem of speaker-independent single phoneme recognition. Phonetically labeled speech data were taken from the TIMIT data bank (National Institute of Standards and Technology, 1990) covering 6300 sentences uttered by 630 male and female speakers of US-American descent. In TIMIT the speech signals have been phonetically transcribed using 61 different symbols providing a very fine distinction of all kinds of plosives, fricatives, nasals, semi-vowels, vowels and different phases of silence within speech. One of the symbols marks long phases of silence and is excluded. The remaining $K = 60$ phoneme classes $j$ have strongly different statistical weights $P_j$ ranging from a few ten to several thousand samples. The data bank is partitioned into a training set (TIMIT/train) and a test set (TIMIT/test) covering 135,518 and 48,993 sample phonemes, respectively.

In our testing scenario, serving to illustrate the GRBF/BMLC, we demand that the preprocessed speech data are classified into all 60 classes defined by the labeling. The applied preprocessing is sketched in Fig. 4 and codes each phoneme utterance into a 63-dimensional pattern vector $x_i$.

6.2. Steps in the construction of the classifier

1. At the outset, the number $M$ of neurons in the central layer of the GRBF was chosen. To achieve a sizable amount of data compression we represent the $N = 135,518$ data vectors $x_i$ of TIMIT/train by $M = 453$ neurons. For the same purpose we restrict the dimension $D'$ for multivariate density estimation (cf. Eq. (48)) to the value $D' = 14$ implementing, thereby, a partial LPCA. Thus, during Multivar the components of the class-conditional mixtures (65) are univariate with variance $\sigma_j$ into the $D - D' = 46$ directions of small local variances and multivariate into the remaining $D'$ directions. For a statistically balanced representation of the $K = 60$ phoneme classes $j$, the numbers $m(j)$ of associated neurons were chosen such that Eq. (62) approximately holds. Owing to the strongly differing sizes $N_j$ of the $X_j$ the values $m(j)$ were in the range $1 \leq m(j) \leq 25$. For each of the 60 classes $j$ the mean value, the covariance matrix, and its spectrum were calculated to obtain starting values for the Univar and Multivar parameters $\Theta_j(j)$.

2. A Univar training was carried out for each of the 55 data sets $X_j$ with $m(j) > 1$. Here, the average log-likelihood $\bar{l}(\sigma) = (1/N_j)\ln(p(x_i|C_j, \sigma))$ (cf. Eq. (27)) was monitored during $\sigma$-annealing. According to Fig. 5 for each class $j$ the average log-likelihood $\bar{l}(\sigma)$ monotonously increases with decreasing $\sigma$ until a maximum is reached. Thus the Univar algorithm allows a stable ML optimization for the $\hat{p}(x_i|j, \Theta_j(j))$. The location, at which the expectation value $\bar{h}(\sigma) = \sum_j P_j \cdot \bar{l}(\sigma)$ assumes its maximum, fixes the common Gaussian width $\sigma_j$ at which the set of codebooks $C_j = \{C_j(\sigma_j)\}_{j = 1, \ldots, K}$ renders an ML density estimate (17) for the whole data set $X$. As given by Eq. (48) this value $\sigma_j = 0.08$ defines the width of the spherical part of the component densities during Multivar.

3. Using Multivar, the class-local parameters $\Theta_j(j)$ were jointly optimized by annealing of the coupling parameter $\mu$ (cf. Section 3). During that annealing, the $m(j)$
components of the class-conditional mixtures (15) slowly became multivariate into the $D_0 = 14$ directions of the largest local variances. Fig. 6 documents the increase of the average log-likelihood $l_j$ during annealing of $m$.

Beyond a certain small value of $m$ the parameter optimization becomes stationary.

6.3. Classifier performance

At each value of the annealing parameters $\sigma$ and $\mu$, one can extract the corresponding parameters $C(j)$ and $\Theta_{j}(j)$, respectively, construct a GRBF/BMLC and measure its performance. To illustrate how that performance depends on the expectation value $\bar{l}$ of the average log-likelihood, we have evaluated the recognition rate during Multivar. This rate is the percentage of correctly classified patterns among all patterns. We will also give percentages for the average reliability of classification defined by $\frac{\sum_j P_j(N_{c,j} / N_j)}{N_j}$. Here, $N_{c,j}$ counts the patterns from class $j$ correctly classified into $j$ and $N_j$ is the number of all patterns classified into $j$.

Fig. 7 exhibits for the training set a linear increase of the recognition rate with $\bar{l}$. Here, the quality of density estimation is directly proportional to the performance of the classifier. Since the recognition rate at the smallest value of $\bar{l}$ pertains to the Univar result, the figure also demonstrates that the multivariate modeling substantially enhances the performance of the classifier. At the optimal parameter set the recognition rate is 75.7%.

The recognition rates measured on the test set and shown in Fig. 7 are smaller indicating that the capability of generalization slowly decreases with increasing quality of density estimation. Here, that rate maximally reaches a value of 62.1% at an average reliability of 61.9%. The specialization of the GRBF is also apparent from the fact that at every single Multivar parameter set, the values of $\bar{l}$ are sizably smaller for the test set (in the figure the triangles are offset to the left from the corresponding diamonds). Thus, the ML density model is worse for the test set and a larger data bank implying better statistics is required to improve the capability of generalization.

6.4. Novelty detection threshold

Fig. 8 shows the cumulative distribution $P(l \leq \ln(\epsilon))$ of the log-likelihood as a function of $\ln(\epsilon)$ (cf. Eq. (74)) obtained for the training set at the optimal network parameters. At the threshold value $\ln(\epsilon) = 45$ of the cut-off parameter $\epsilon$, corresponding to an essentially vanishing significance level $P(l \leq \ln(\epsilon)) = 0$, all patterns randomly generated from a uniform distribution (71) are safely rejected through Eq. (73) and, thus, identified as novelties. The situation changes if data are presented to the classifier more closely resembling the training data. To explore this issue we have generated a new data set from TIMIT/train referring to different time points of speech flow, i.e. to the time points of phoneme transitions. The corresponding cumulative distribution function of the log-likelihood is also depicted in Fig. 8 and demonstrates that for the phoneme transitions the likelihood $p(x; \Theta_j)$ is on the average by about one order of magnitude smaller than for the phoneme centers. However, since the two distribution functions exhibit a strong overlap, this difference does not allow a safe classification of phoneme transitions as novelties. Therefore, other means have to be taken for the segmentation of continuous speech flow into sequences of phonemes.
7. Summary

Local agents in biological systems strive for activity and growth. However, the limitation of resources enforces their competition. As a result, the systems differentiate into complex dynamical structures, whose components are specialized to different tasks but contribute in a balanced way to the functionality of the whole. Concerning neural tissue, corresponding principles of self-organization may be formulated in terms of competitive Hebbian learning (cf. Eqs. (39), (45) and (57)) or in terms of an activity normalization within a neural layer (cf. Eq. (7)). In addition, the properties of neural tissue such as the threshold behavior of individual neurons and the presence of lateral and reverse connections have to be taken into account.

Using this general view as a guideline we have developed a self-organizing scheme for Bayesian pattern classification and novelty detection by a GRBF network of the ideal type. The quoted learning algorithms provide safely converging ML estimates of class-local data distributions in terms of multivariate normal mixtures. Applying these results we have specified a sequence of training steps for technical applications using speaker-independent single phoneme recognition as an example. Here we have illustrated how the quality of the classifier increases during self-organization.

Our results are not meant as a theory of “how neural tissue self-organizes to perform Bayesian classification”. For this purpose a detailed analysis of the connections between model assumptions and physiological findings would be required. Our results show, however, that one can obtain clues for finding solutions of complicated algorithmic problems by exploiting known aspects of neural behavior.

References


