Neural Networks, Vol. 4, pp. 89–102, 1991
Printed in the USA. All rights reserved.

0893-6080/91 $3.00 + .00

ORIGINAL CONTRIBUTION

Maximum Likelihood Neural Networks for Sensor Fusion and Adaptive Classification

LEONID I. PERLOVSKY AND MARGARET M. MCMANUS

Nichols Research Corporation

(*Received 28 February 1989; revised and accepted 4 June 1990*)

Abstract—*A maximum likelihood artificial neural system* (MLANS) *has been designed for problems which require an adaptive estimation of metrics in classification spaces. Examples of such problems are an* XOR *problem and most classification problems with multiple classes having complicated classifier boundaries. The metric estimation has the capability of achieving flexible classifier boundary shapes using a simple architecture without hidden layers. This neural network learns much more efficiently than other neural networks or classification algorithms, and it approaches the theoretical bounds on adaptive efficiency according to the Cramer–Rao theorem. It also provides for optimal fusing of all the available information, such as a priori and real-time information coming from a variety of sensors of the same or different types, and utilizes fuzzy classification variables to provide for the efficient utilization of incomplete or erroneous data, including numeric and symbolic data.*

This paper describes the neural network and presents examples of its performances in unsupervised, partially supervised, and environment-interrogation modes. We discuss the Cramer–Rao theory as it relates to neural networks, the relevance of the MLANS to biological and other neural networks, and issues for future work.

Keywords—Neural networks, Maximum likelihood, Adaptive efficiency, Cramer–Rao bounds, Sensor fusion, Adaptive classification, Phase transitions, Attention.

I. INTRODUCTION

Recently Widrow (1988) has addressed the question of how much can be learned from a given number of samples, and what principles should be used to design the most efficient-learning neural networks. Throughout this paper such neural networks will be called *efficient*. The partial answer to this question is given by Adaline, which is an efficient neural network for certain classes of problems. This follows from the Cramer–Rao (CR) theorem (see, i.e., Cramer, 1963), which establishes bounds on the accuracy of estimated quantities, and which can be used to establish bounds on learning efficiency (Perlovsky, 1988a). One of the consequences of the CR theorem is that the maximum likelihood (ML) estimation is asymptotically efficient in many problems of practical interest; that is, the ML procedures yield results which are close to the CR bounds for a large number of samples. Examples of problems for which Adaline performs the ML estimation are linear prediction and linear classification.

In this paper the ML artificial neural system (MLANS) is designed for more complicated problems requiring nonlinear classification boundaries (such as Exclusive OR, and more complicated configurations). This neural network has ML neurons, which adaptively estimate the local metric in the classification space (Perlovsky, 1987). This permits the design of flexible classifier shapes using no-hidden-layer architecture and provides orders-of-magnitude improvement in learning efficiency. The learning efficiency of this network approaches the CR bounds with a relatively small number of samples. In addition, the ML approach allows for optimal fusion of all available information, such as a priori and real-time information coming from a variety of sensors of the same or different types.

Section 2 describes the architecture and learning rules of the MLANS. This is followed by performance examples in Section 3. Discussion of CR bounds and their relation to neural networks is provided in Section 4. In Section 5 the biological relevance is

The authors appreciate multiple discussions with William Schoendorf and his encouragement in searching for the ML solution in every aspect of the problem. Also discussions with Stephen Grossberg and John Daugman were helpful and are appreciated. The authors are grateful for helpful suggestions by the referees.

Requests for reprints should be sent to Leonid I. Perlovsky, Nichols Research Corporation, 251 Edgewater Drive, Wakefield, MA 01880.

addressed, including the question: Does the brain estimate metrics?

2. THE ML NEURAL NETWORK ARCHITECTURE

The architecture of the MLANS (Perlovsky, 1987) is shown in Figure 1. It has as its input all the available data including the observations, $\mathbf{X}_n$, prior information, and environmental interrogation feedback, if available. The output nodes contain the estimated parameters of all classes and types of objects, the estimated number of objects of each type

$$N_{km} = \sum_{n=1}^{N} W_{nkm}, \tag{1}$$

the estimated vector of means of each type

$$\mathbf{M}_{km} = \sum_{n=1}^{N} \frac{W_{nkm}\,\mathbf{X}_n}{N_{km}}, \tag{2}$$

and the covariance matrix of each type

$$\mathbf{K}_{km} = \sum_{n=1}^{N} \frac{W_{nkm}\,(\mathbf{X}_n - \mathbf{M}_{km})^T\,(\mathbf{X}_n - \mathbf{M}_{km})}{N_{km}}, \tag{3}$$

In these formulae the index k refers to the class of the object, and the index m refers to the type of the object within the class. The classification of object types into classes can be achieved on the basis of prior information, environment interrogation, or teacher information, depending on what information is available. Various examples of classification are discussed in the next section. The assignment of objects to the types can be done solely on the basis of statistical properties of the objects; if additional information is available on object-to-type assignments it can be accounted for, similar to the classification information.

The weights W_{nkm} in this neural network are the fuzzy classification variables, in the spirit of the fuzzy set theory (i.e., Kaufmann, 1975); the weights assign every object to each type with a fuzzy measure. If all the information available to the network is specified in probabilistic terms, the weights are defined as posteriori Bayesian probabilities for an object n to belong to class k and type m, $P(k, m|n)$ (see, for example, Fukunaga, 1972):

$$W_{nkm} = P(k, m|n) = \text{pdf}\,(\mathbf{X}_n|k, m) \Big/ \sum_{k',m'} \text{pdf}\,(\mathbf{X}_n|k', m'),, \tag{4}$$

where pdf $(\mathbf{X}_n|k, m)$ is a probability distribution function for the observation $\mathbf{X}_n$, given it is of type m and class k.

The explicit formulae for the weights depend on the statistical model assumed for the data, which determines the learning algorithms. In the examples of the next section, the assumed model is a Gaussian mixture (see, for example, Titterington et al., 1985), so that the probability distribution of each object-type is described by a Gaussian distribution. A Gaussian mixture model assumes that observations are statistically independent, so that the total pdf for all observations $\{\mathbf{X}_n, n = 1, \ldots, N\}$ is a product of individual pdf $(\mathbf{X}_n)$, which are assumed to be sums of Gaussian functions:

$$\begin{aligned} \text{pdf}\,\{\mathbf{X}_n, n = 1, \ldots, N\} &= \prod_{n=1}^{N} \text{pdf}\,(\mathbf{X}_n), \\ \text{pdf}\,(\mathbf{X}_n) &= \sum_{k=1}^{K} \sum_{m=1}^{M} \text{pdf}\,(\mathbf{X}_n|k, m), \\ \text{pdf}\,(\mathbf{X}_n|k, m) &= (2\pi)^{-d/2}\,(\det \mathbf{K}_{km})^{-1/2} \\ &\quad \times \exp\left(-\tfrac{1}{2}\mathbf{D}_{nkm}\mathbf{K}_{km}^{-1}\mathbf{D}_{nkm}^T\right), \\ \mathbf{D}_{nkm} &= \mathbf{X}_n - \mathbf{M}_{km}. \end{aligned} \tag{5}$$

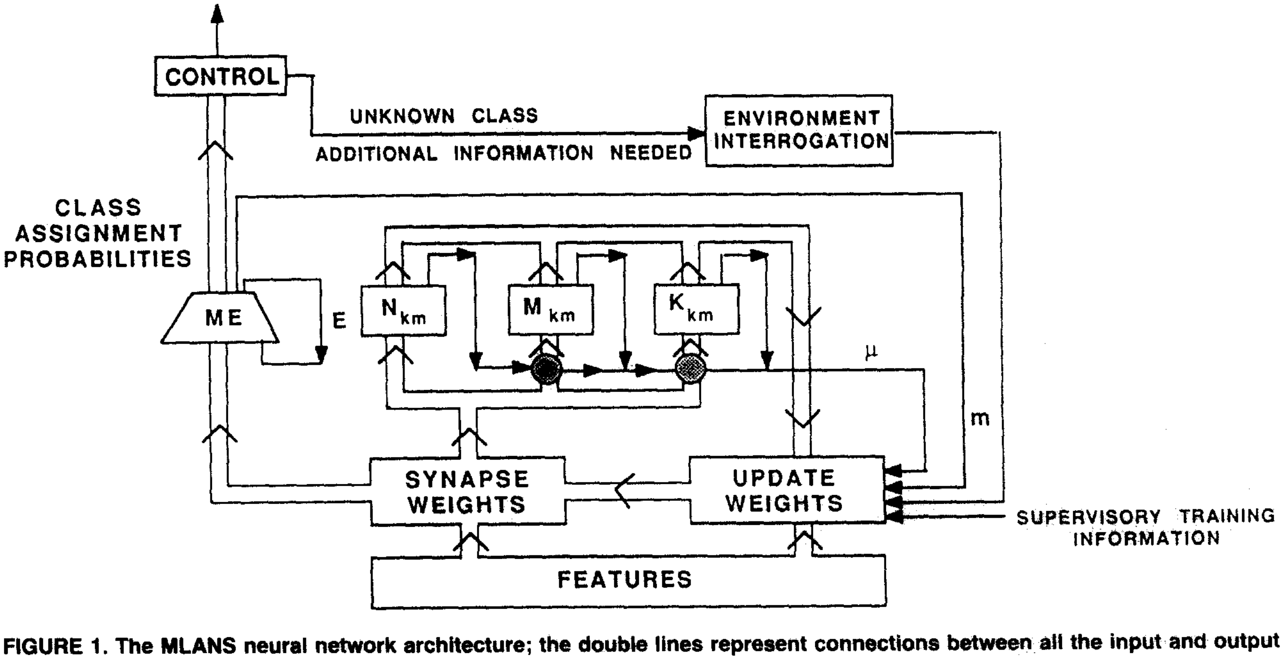
CONTROL
UNKNOWN CLASS
ADDITIONAL INFORMATION NEEDED
ENVIRONMENT INTERROGATION
CLASS ASSIGNMENT PROBABILITIES
ME
E
N_{km}
M_{km}
K_{km}
μ
m
SYNAPSE WEIGHTS
UPDATE WEIGHTS
SUPERVISORY TRAINING INFORMATION
FEATURES

FIGURE 1. The MLANS neural network architecture; the double lines represent connections between all the input and output nodes.

This model is appropriate for a wide variety of application problems for the following reasons. The observations $\mathbf{X}_n$, often include Gaussian noise, which tends to make each object type nearly Gaussian. The neural network is robust with respect to small deviations of class distributions from Gaussian distributions, and large deviations from Gaussian shape are handed by additional object types.

The estimation of the optimal number of object types is a problem that has not been solved in the past clustering research. Many approaches have been suggested (see, for example, Fukunaga, 1989) which are useful for specific problems or rely on prior experience or knowledge. For example, the vigilance parameter of the ART neural network (Carpenter & Grossberg 1987) is not determined by an internal dynamics of an ART neural network and should be specified based on prior experience. A general approach can be based on the ML principle. The Akaike-type modification of the ML principle (Akaike, 1973, Yarman-Vural & Ataman, 1987, Perlovsky, 1986) has an attraction of providing an asymptotically efficient estimate of a number of object types, which entirely relies on the internal dynamics of the MLANS and does not need any prior knowledge or experience. However, none of these methods directly addresses a goal of minimizing classification errors, which is a central goal of classification. For example, the ML approach minimizes a likelihood function, which is a homogeneous quantity, approximately equally distributed over all observations. This is not satisfactory in many practical cases, when of particular interest in finding (with minimal error) a class containing a small number of objects among other classes containing large numbers of objects. The ML approach is, therefore, "biased" toward an accurate description of large classes at the expense of the small class.

It is necessary, therefore, to develop methods which are more directly related to the minimization of classification errors. A straightforward minimization of a classification error for finding the best number of clusters in a supervised case is a possible approach, although a cumbersome one due to the complicated procedures required to estimate a classification error. We propose to utilize the minimum entropy (ME) approach, previously developed by one of the authors (Perlovsky, 1987). This is a fast parallel procedure, which can be easily implemented within MLANS and is suitable in both supervised and unsupervised cases. The ME neuron selects the number of object types that minimizes the entropy of the estimated distributions. This procedure described below is equivalent to maximizing the orderlines of fuzzy classification, or to maximizing class separability, and it closely approximates minimization-of-classification errors.

For the purpose of minimizing a classification error, an entropy, E, of the data set $\{\mathbf{X}_n\}$ with regard to a set of posteriori Bayes classification probabilities $\{P(k|n)\}$,

$$P(k|n) = \sum_{m=1}^{M} P(k, m|n), \tag{6}$$

is defined as a mean value of the logarithm of the probability averaged over all classes, and it is estimated as follows:

$$E = -\sum_{k} P(k|n) \ln P(k|n). \tag{7}$$

This is a nonnegative quantity which reaches its minimum, $E = 0$, when a classification is nonfuzzy, that is, when each object is assigned to a single class with a probability 1, so that all $P(k|n) = 0$ or 1.

Because MLANS weights W_{nkm} estimate posteriori Bayes probabilities $P(k, m|n)$, the ME neuron performs calculations as follows:

$$E = -\sum_{k=1}^{K} \left(\sum_{m=1}^{M} W_{nkm} \right) \ln \left(\sum_{m=1}^{M} W_{nkm} \right). \tag{8}$$

Beginning with a preset value of K and M, the ME neuron calculates entropy (8), then it resets values of K and M and reinitiates the ML subsystem of the MLANS. The resulting new value of the entropy is compared with the previous one and iterating continues until the minimum value of entropy is found.

The learning process of the MLANS can be supervised or unsupervised. It can also be any mixture of unsupervised learning with partial or imperfect supervision. In order to achieve this flexibility the weights are modified to account for all of the available information, including supervisory or teacher's information. A teacher's assignment probability of a target n to a class k, type m will be denoted using a subscript T:

$$P_T(k, m|n). \tag{9}$$

If a teacher provides a perfect classification label, this value is 1 for a particular class and type, and zero for all the rest. In the case when a teacher provides only probabilistic or fuzzy (or tentative) assignments, these values range between 0 and 1. These values are normalized by MLANS according to

$$P_T(k, m|n) \longrightarrow P_T(k, m|n) \Big/ \sum_{k,m} P_T(k, m|n), \tag{10}$$

so that they satisfy the constraint

$$\sum_{k,m} P_T(k, m|n) = 1. \tag{11}$$

This is necessary in order to maintain a probabilistic interpretation of weights, which are modified according to

$$W_{nkm} \longrightarrow W_{nkm} \cdot P_T(k, m|n); \tag{12}$$

this modification results in the ML solution to a classification problem, a solution which optimally fuses all the information from sensors and teachers. The probabilistic interpretation of modified weights (12) also implies that teacher's information is statistically independent from the previously defined quantities $P(k, m|n)$, as is usually the case when a teacher "just knows" the class and type of an object, or derives its information from another sensor. In this case modification (12) corresponds to a classical formula for combining independent probabilties. If the weights (12) only approximate posteriori Bayes probabilities, the MLANS converges to an approximation of the ML solution.

Often a teacher's information is incomplete, for example, when only a few objects are examined by the teacher. In this case it is sufficient to modify weights $W_{nk\hat{m}}$ only for those objects n for which a teacher's information is available. This procedure results in an optimal (the ML) fusion of all of the available teacher's and sensory information; often a teacher's information on only a few objects leads to an improvement of classification of all objects resulting in a high learning efficiency.

Another important case of partial supervision is when a teacher provides information on class assignment only, $P_T(k|n)$, and no information on the type. For example, when a neural network is trained to recognize several different objects viewed from different angles, then during training, the class of each object is known; however, it is desirable that the neural network determines on its own how many types it needs to represent each object adequately for robust and accurate classification, and automatically estimates parameters for these types. It turns out that the weight modification formula (12) still provides an optimal solution which maximizes the likelihood of all the available information. This can be shown as follows. According to the Bayes theorem (see, for example, Fukanaga, 1972), a posteriori Bayes probability $P(k, m|n)$ can be represented as a product of two terms, one term which is the object's class assignment $P(k|n)$, eqn (6), and another term which is a posteriori probability of an object's type, conditioned on an object's class:

$$P(m|k, n) = P(k, m|n)/P(k|n), \tag{13}$$

or

$$P(k, m|n) = P(k|n) \cdot P(m|k, n). \tag{14}$$

A class assignment probability, $P(k|n)$, is modified using a probability supplied by a teacher, $P_T(k|n)$, according to a rule of combining independent probabilities:

$$P(k|n) \longrightarrow P(k|n) \cdot P_T(k|n), \tag{15}$$

which results in weight modification formula (12).

A convergence of the MLANS learning process is determined by comparison of the estimated distributions of object types to those of the previous iteration. The Bhattacharyya distance (see, e.g., Fukunaga, 1972) between successively estimated distributions of each type is used as a measure of convergence. In the perfectly supervised mode the class and type labels are available along with the training objects. This information is used to modify the weights to be 0 or 1 (perfect class and type assignments), and the parameters of classes and types are estimated in a single pass resulting in the standard ML estimates of means and covariances (the minimum two iterations are needed to establish convergence). In order to estimate a mean vector, a single observation is sufficient for each class and type. In order to estimate also a nondegenerate covariance matrix in d dimensions, at least $(d + 1)$ observations are needed for each class and type. If the number of observations exceeds the minimal value, the accuracy of the ML estimation tends quickly to the best possible accuracy determined by the CR bounds, as has been well studied theoretically (Cramer, 1963).

If the number of observations is insufficient to estimate a covariance matrix of a particular object type, several alternatives are possible. The covariance matrix can be set to a predetermined (small) value, such as an observation noise covariance matrix, or to a large value, such as the overall covariance matrix of all the objects, depending on the objective and prior knowledge. An alternative is to use a weighted sum of the noise covariance matrix and of the estimated matrix, in the hope that the resultant matrix is both nonsingular and related to the data. An even more advantageous approach that we have recently developed consists in estimating the Cholesky decomposition of the inverse covariance matrix (Perlovsky & Marzetta, 1989).

In the unsupervised mode the ML neural network learning is an iterative process. Each iteration consists of estimating the object-type parameters, followed by weight update, which is determined by the assumed statistical model, the observed data, and by the estimated values of the parameters at the previous iteration. In our experience the neural network quickly converges to a solution of the ML equation. When only a small number of objects is observed, fixed points might correspond to local maxima of the likelihood function. When the number of objects increases, the neural network finds the global maximum of the likelihood function.

If the nature of an application requires or permits interrogation of the environment, such as manual inspection or engagement of an interactive sensor, this is usually performed after the convergence of unsupervised iterations. The decision of which objects to interrogate is optimized in order to save the interrogative resources (e.g., time and sensor resources). Several ways to achieve this objective are discussed in the next section. The results of interrogation of a few objects are used to improve the previous results of unsupervised classification of all the objects. For this purpose the interrogation results, depending on their nature, are expressed as additional classification features, or as perfect, probabilistic or fuzzy class assignments which are input into the weight-update block similar to a supervisory information. If the information on object types is available from the interrogation, it is also input in the same way. The interrogative information in the weight-update block is used to modify weights for the few interrogated objects similar to the above discussion of supervisory information. This reinitiates the unsupervised learning iteration loop and leads to the new values of parameters and new classification of all the objects. This procedure might lead to only minor modifications of the previous results if the results of interrogation are consistent with the results of previous learning. On the other hand, if the results of interrogation contradict the results of previous learning, a significant change of parameters and class and type assignment probabilities of all objects might occur as a result of only a few interrogations. This high learning efficiency is a result of optimal utilization of all the available information in the MLANS neural network.

3. PERFORMANCE EXAMPLES

Three examples are discussed in this section. In the first one the MLANS neural network performs optimal interrogation of the environment. In the second example the neural network learns a complicated shape of a classifier boundary with imperfect supervision. The third example compares the performance of the MLANS in a clustering mode to a clustering algorithm using standard data sets in eight dimensions.

In the first example a few defective parts have to be identified among hundreds of perfect parts and other objects. The only prior information available is the approximate ratio of 1:10:100 of (defective parts):(other objects):(perfect parts). The specification was to reduce the necessary inspection rate to below 100 per 1000 objects. The classification is performed in a two-dimensional classification space of feature 1 and feature 2 extracted by a preprocessor from sensory data. The actual configuration in the classification space of 816 objects available to the neural network is shown in Figure 2. The 12 defective parts, if not marked, would be impossible to identify by an eye as a separate cluster. The scatter in the distributions is due to variation in aspect angles and due to sensor errors.

The true distributions corresponding to this data are shown in Figure 3a by illustrating the 2-σ Gaussian boundaries of the distributions. The correct classifier boundary is shown by a dotted line. These distributions have been estimated by the network in a supervisory mode using 100 objects of each type. They are shown here for later comparison with the results of unsupervised learning. The prior distributions for unsupervised learning are shown in Fig-

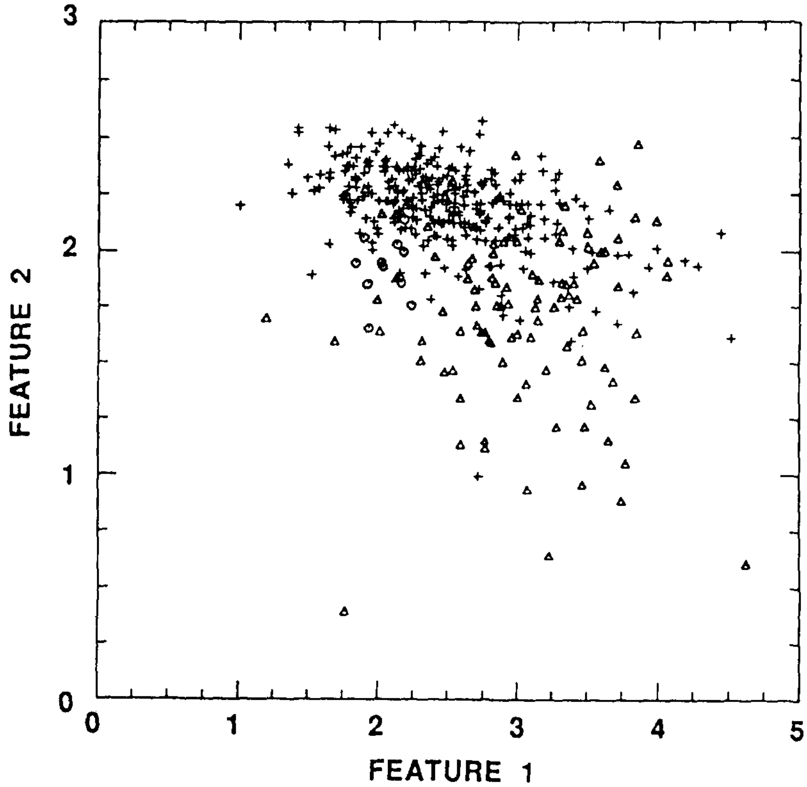

FIGURE 2. The configuration of objects in the classification space, example 1.

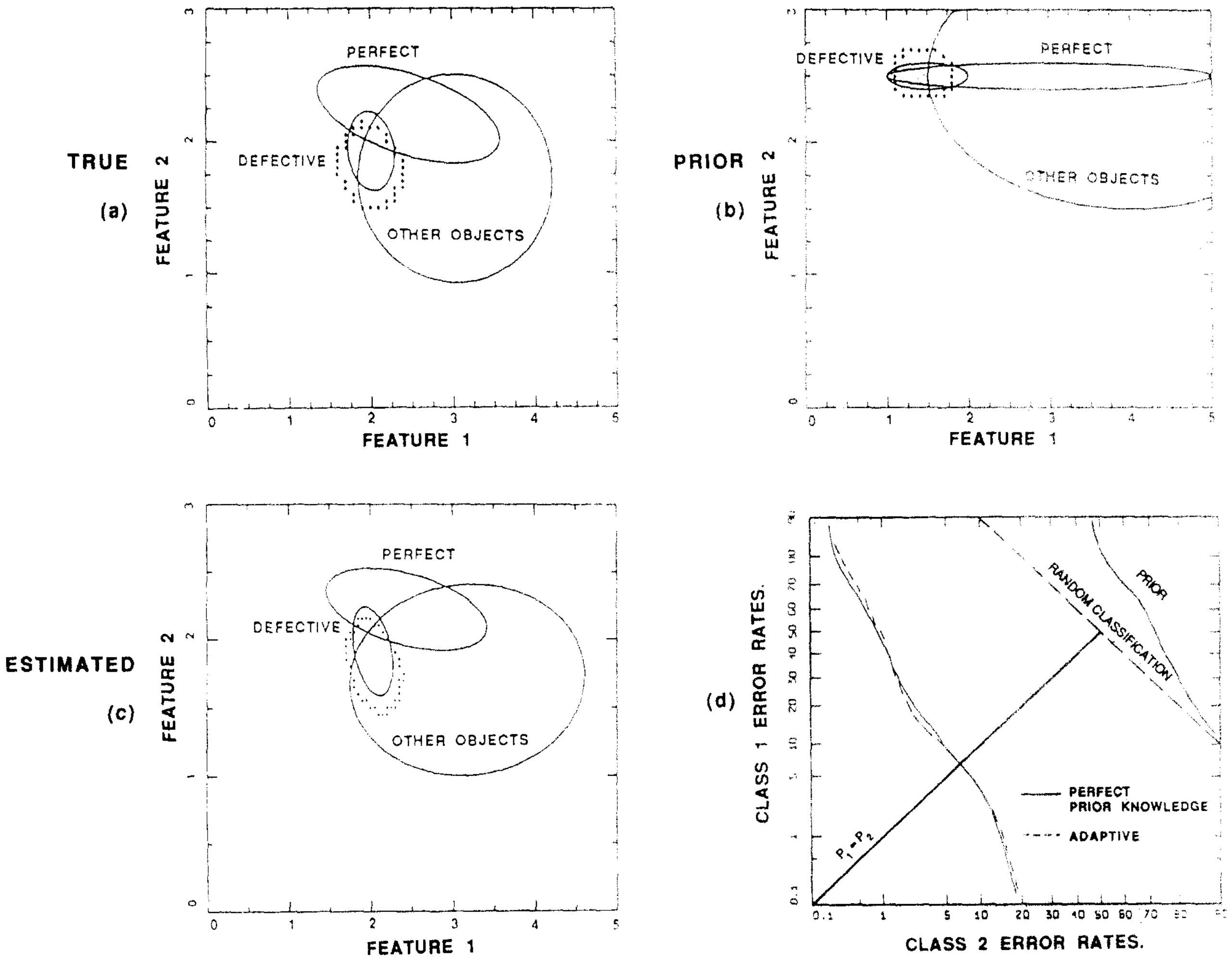

FIGURE 3. Optimal environment interrogation (example 1). (a) True distributions corresponding to the data in Figure 2 and a classifer boundary; (b) prior (guess) distributions have no resemblance to the true ones; (c) neural network estimated distributions are very similar to the true ones; (d) operating characteristics using adaptive classification by the neural network is very similar to the one obtained with the perfect prior knowledge.

ure 3b (the dotted line shows the corresponding classifier); they have been selected based on general considerations and they are far from the truth (Figure 3a). Nevertheless, the network has been able to find the cluster of 12 defective parts. The results of unsupervised learning are shown in Figure 3c. They are seen to be very close to the true distributions in Figure 3a.

The operating characteristics corresponding to classification using perfect prior knowledge and using learned distributions, shown in Figure 3d, are very similar. This indicates that the neural network in this case, without supervision, performs as well as if the perfect prior knowledge of the distributions were available. On the other hand, if the prior guess is used (a classifier in Figure 3b), the results are even worse than a random labeling.

Because there are overlaps between the distributions, the defective parts can not be identified without error. The Bayes risk (class 1 error = class 2 error) in this case, as seen in Figure 3d, is about 5%. Based on the results of unsupervised learning, 80 objects were then selected for inspection and all 12 defective objects were correctly identified.

In this case the number of object types was not known a priori. When the ML principle, modified according to Akaike (Perlovsky, 1986), was used to determine the optimal number of object types, it resulted in four types, of which two types have been used by the neural network to achieve a more accurate description of the perfect-object type. It also resulted in a less-accurate description of the defective-object type. The ML principle, as discussed above, favors an accurate description of a class with a large number of objects at the expense of a less accurate description of a class with a small number of objects, which results in a larger overlap between classes. This is not consistent with the classification objective. Therefore the number of types in this problem should be selected using the ME principle, which favors class separability. In this problem the ME principle results in a more robust description using three types of objects.

This example demonstrates the advantage of estimating the local metric in the classification space, which is achieved by estimating covariance matrices of each type. By so doing, a classifier of complicated shape including isolated regions is designed by a network without hidden layers. It also demonstrates the ability of a network to "conceptualize": it learns the types on its own, and it estimates the optimal number of types for classification.

The second example demonstrates learning of a complicated classifier boundary shape in real time with imperfect supervision. The term "real time" here is used in a sense similar to that of Carpenter (Carpenter, 1989): the supervision in the MLANS is accounted for via internal network dynamics. The true distributions of the six types of objects for two classes are shown in Figure 4a. Vertical and horizontal axes correspond to two discrimination features; the true classifier boundary is shown by a dotted line. The initial information was absent and the supervision provided only information on class assignment and no information on the type. Figure 4b shows the estimated distributions after 50 observations. Of these 50 observations there were only 2 observations of the third type of class 1; therefore, the covariance matrix for this distribution could not be estimated from the data and the network set it equal to the noise covariance matrix. Nevertheless, the classifier boundary is close to the true one and classification errors are close to the true minimum errors for more than 50 observations. This example again demonstrates the advantage of estimating the local metric in the classification space, which enables a network without hidden layers to design a classifier of complicated shape including isolated regions. It demonstrates the ability of the MLANS to "conceptualize": it learns the types on its own. Also, a fast learning with imperfect supervision is achieved in this case due to optimal fusion of incomplete teacher's information with sensory data.

The third example provides the test of the neural network performance using a standard data set and provides comparison of the neural network performance to a state-of-the-art clustering algorithm (a clustering algorithm ISODATA (see Fukunaga, 1972), is relying on nearest-neighbor clustering similar to RCN, Hoppfield, and ART neural networks). In this example unsupervised learning is performed in eight-dimensional classification space.

The results of clustering of two-class data are summarized in Table 1 for 100 objects of each class. The misclassification errors obtained with the neural network are close to the Bayes risk (about 2%). The results obtained with ISODATA clustering algorithm (see Fukunaga, 1972) are significantly worse.

The fact that the classification errors are close to the Bayes risk, which is the minimal possible error given the perfect prior knowledge of class distributions, suggests that the neural network yields an accurate estimation of all parameters of the distributions. This is further confirmed in Figure 5, where some two-dimensional projections of the eight-dimensional results are shown; the means and covariances are estimated close to the true values. Another way to quantify these results is by using the Bhattacharyya distances between the estimated and the true distributions. These are shown in Figure 6 for each class as a function of the number of objects from each class available for the neural network learning process. The initial Bhattacharyya distances between the true distribution and the initial guess for each class are quite large (about 4) due to the absence of prior knowledge. (In fact, the "worst"

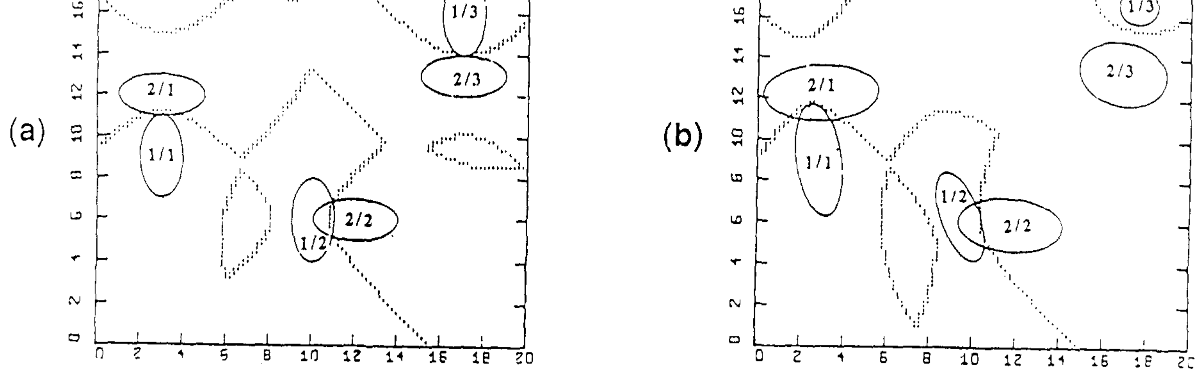

FIGURE 4. Learning a complicated shape of a classifier boundary with imperfect supervision (example 2). (a) True distributions of the six types of objects of two classes; (b) neural network estimated distributions after 50 observations.

TABLE 1
Comparison of the MLANS neural network and ISODATA algorithm using standard data set; two classes, eight-dimensional data.

		MLANS		Isodata algorithm	
		assigned class		assigned class	
		1	2	1	2
Actual class	1	98	2	100	0
	2	3	97	19	81

initial guess was obtained by dividing the total distribution in halves along each axis, and by alternating the class assignments of these halves in such a way that the initial guesses are in the most distant eight-dimensional quadrants from the true distributions). As the neural network starts learning, the Bhattacharyya distance is reduced to a very small number, with only 50 observations per class. There is always some difference between the estimated distribution and the true one; therefore the Bhattacharyya distance never equals zero, which would correspond to perfect estimation. The Bhattacharyya distance therefore is positively biased. The blow-up of this region is shown in Figure 7. The dashed line in Figure 7 shows the bias of the estimated Bhattacharyya distance with perfect knowledge of class assignments of each object, which has been calculated by Fukunaga and Hayes (1988). This line is a best performance bound for an averaged performance of any algorithm or neural network. The fact that the Bhattacharyya distance estimated by the neural network without any class assignment knowledge is close to the theoretical limit, given the perfect knowledge of class assignment, is quite remarkable. This will be further discussed in the next section.

The internal convergence properties of the neural network learning process are illustrated in Figure 8 by plotting the number of internal iteration cycles in the ML subsystem of the network as a function of the number of objects. The initial number of iterations is not very large. Subsequent iterations are initialized when new objects become available to the network. These iterations provide only minor refinements to the previously obtained solutions and their number is the minimal 2. The relatively large number of iterations in the beginning can be interpreted as a long relaxation time near the point of the phase transition (Kryukov, 1988); this interpretation is discussed below in Section 5.

Similarly good results are obtained with three classes in eight dimensions. These results are summarized in Table 2. Again, classification errors obtained with the ML neural network are close to the Bayes risk, which is significantly better than the performance of ISODATA algorithm. Some of the two-dimensional projections of these eight-dimensional results are shown in Figure 9, and the Bhattacharyya distances between the estimated and the true distributions are shown in Figure 10. Again, 50 observations per class are sufficient in order to obtain accurate estimates in this case.

4. CRAMER-RAO BOUNDS

This section describes the CR bounds for the well-known case of perfectly supervised learning, and describes new results obtained by one of the authors (L.I.P.) for unsupervised classification. Then, the application of CR bounds to various aspects of neural network design are discussed.

The CR theorem (see Cramer, 1963) establishes

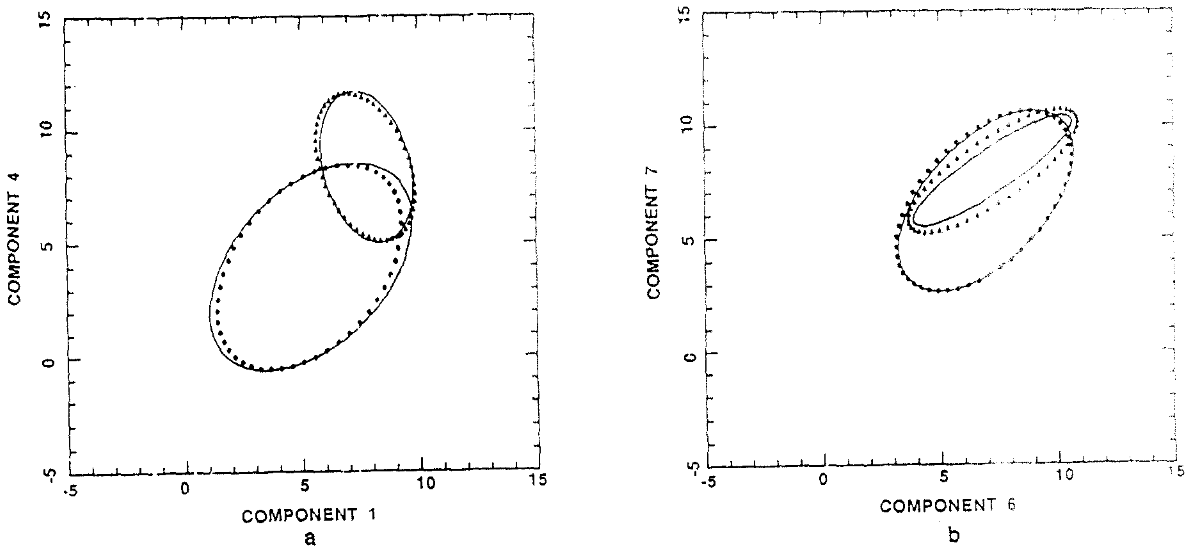

FIGURE 5. Examples of two-dimensional projections of eight-dimensional, two-class data (example 3). Distributions are shown by illustrating 2-σ boundaries; solid lines are used for true distributions and symbols are used for the distributions estimated using the MLANS neural network.

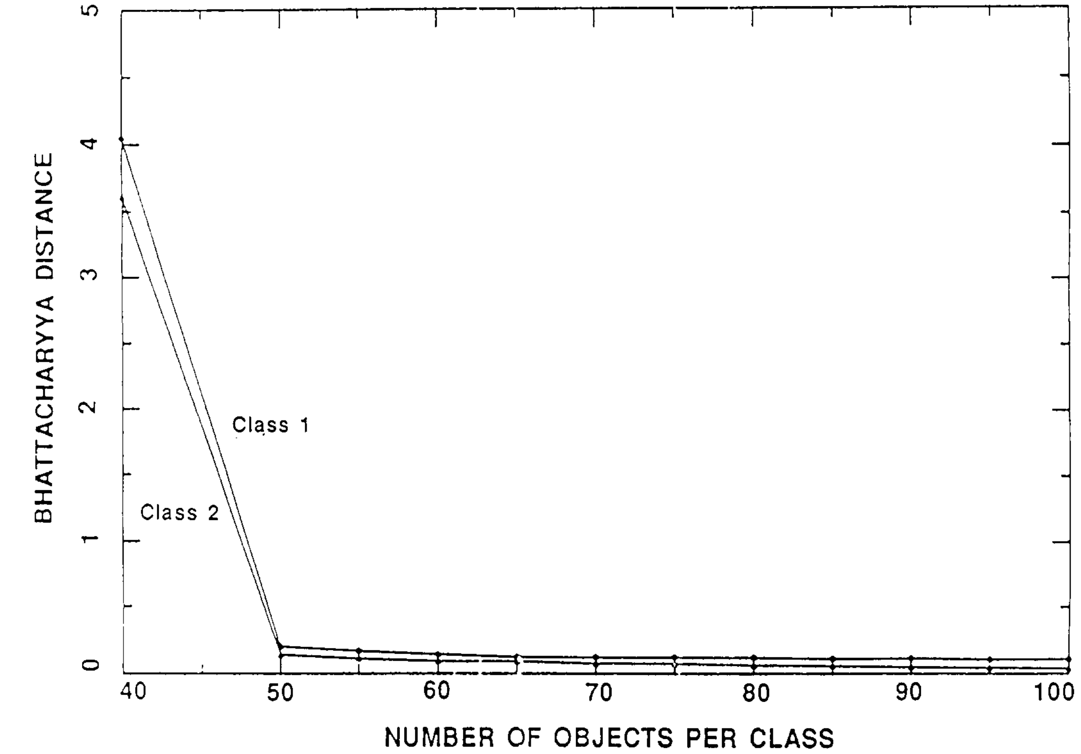

FIGURE 6. Bhattacharyya distances between the true and the estimated distributions for each class; unsupervised learning of two classes, (example 3). The initial guesses are very far from the the true distributions; after adaptation a very close estimate is achieved with as little as 50 objects from each class.

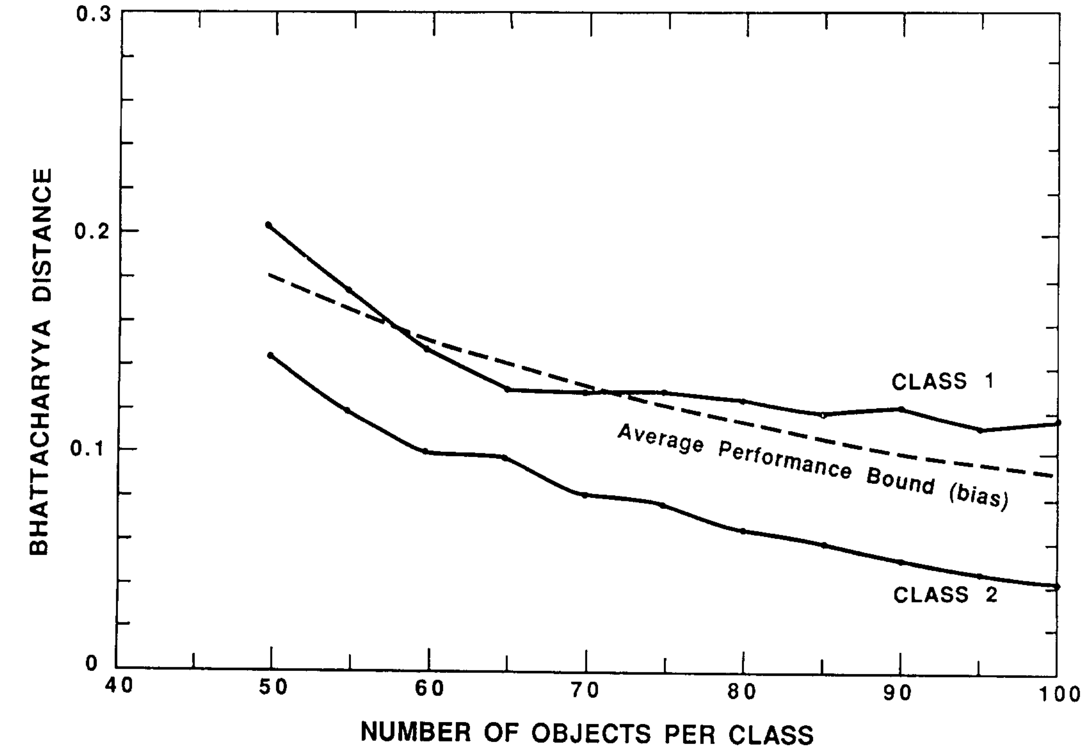

FIGURE 7. Same as in Figure 6, a blow-up of the region $N \geq 50$. The dashed line shows the theoretical bound for an average performance of any algorithm calculated for the perfectly supervised case (same for each class).

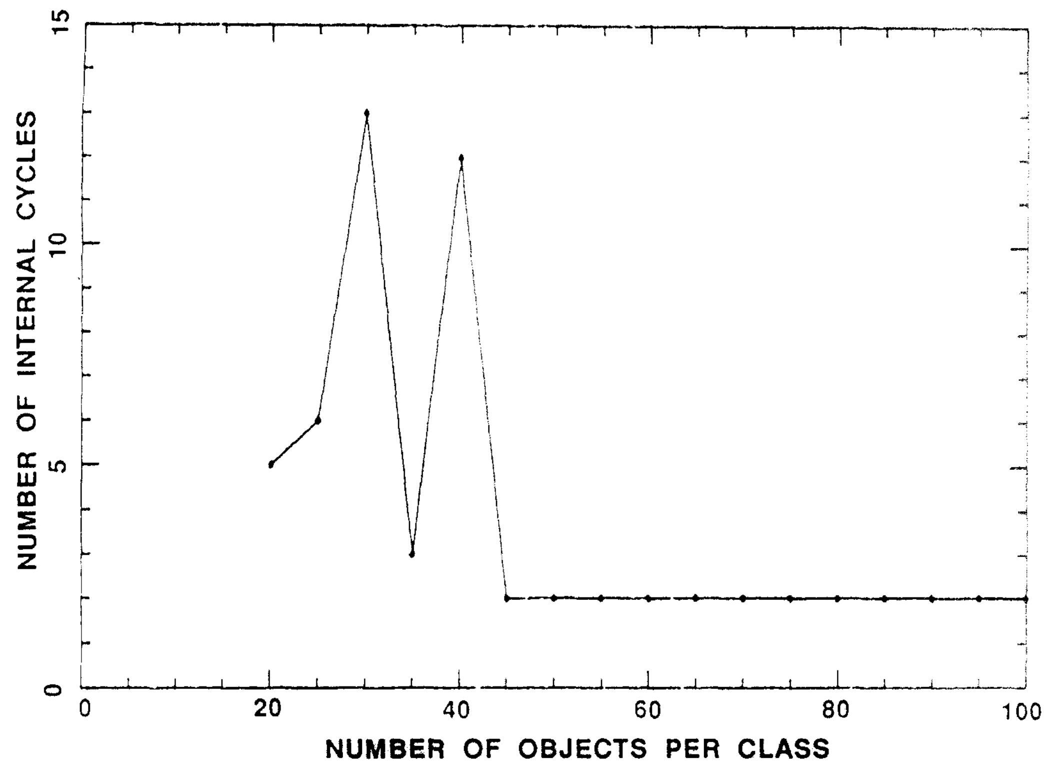

FIGURE 8. Number of internal iteration cycles in example 3 vs. number of objects per class. The network quickly converges with the minimum number (2) of iterations for $N \geq 45$. For a small number of objects, transitions between local maxima are evident, leading to metastable memory states, with the lifetime and order-of-magnitude larger than the neuron cycle time.

lower bounds for the statistical errors of estimated quantities, regardless of the estimation process. For example, consider a simple problem of estimation of the mean of N one-dimensional measurements coming from a single class, which has Gaussian distribution with standard deviation σ. The statistical variance of the conventional ML estimation of the mean is well known to be σ^2/N, which is the CR bound. A neural network implementing this estimation is an efficient learning network, because no other approach can yield a more accurate mean value.

In the multidimensional case the CR bound for the mean is an ellipsoid in the corresponding space; its shape is determined by the covariance matrix and its size is determined by the number of objects in the same way as in an one-dimensional case. If the covariance matrix is known, the conventional mean estimation is the ML estimation, and its accuracy is equal to the CR limit. If the covariance matrix is unknown and is to be estimated from the data, no estimation process can reach the accuracy of the CR bound. However, the standard ML estimation tends quickly to the CR bound with increasing number of observations.

TABLE 2
Comparison of the MLANS neural network and ISODATA algorithm using standard data set; three classes, eight-dimensional data.

	MLANS			Isodata algorithm		
	assigned class			assigned class		
	1	2	3	1	2	3
1	98	2	0	98	2	0
Actual class 2	2	97	1	27	73	0
3	1	1	98	18	0	82

Perfectly supervised multiclass learning is no different than a single-class problem as far as the learning efficiency is concerned. Indeed, in this case the weights in eqns (1)–(3) are all ones or zeroes, resulting in the standard single-class ML estimation for each object type.

Unsupervised learning or imperfectly supervised multiclass learning, however, is essentially different from a single-class problem in that the classification labels for the "training set" are unavailable (or imperfect). The classification should be derived from the estimated parameters of the distributions, and the errors in classification and in the estimated parameters contribute to each other. The CR bounds for unsupervised learning have been derived recently by one of the authors (Perlovsky, 1988c, 1989). These CR bounds are ellipsoids, as in the case of supervised learning. However, their shape and size depend on the geometry of the overlap between classes. This result is intuitively clear. The greater the overlap between classes and types is, the larger is the CR bound. And the accuracy of the distribution esti-

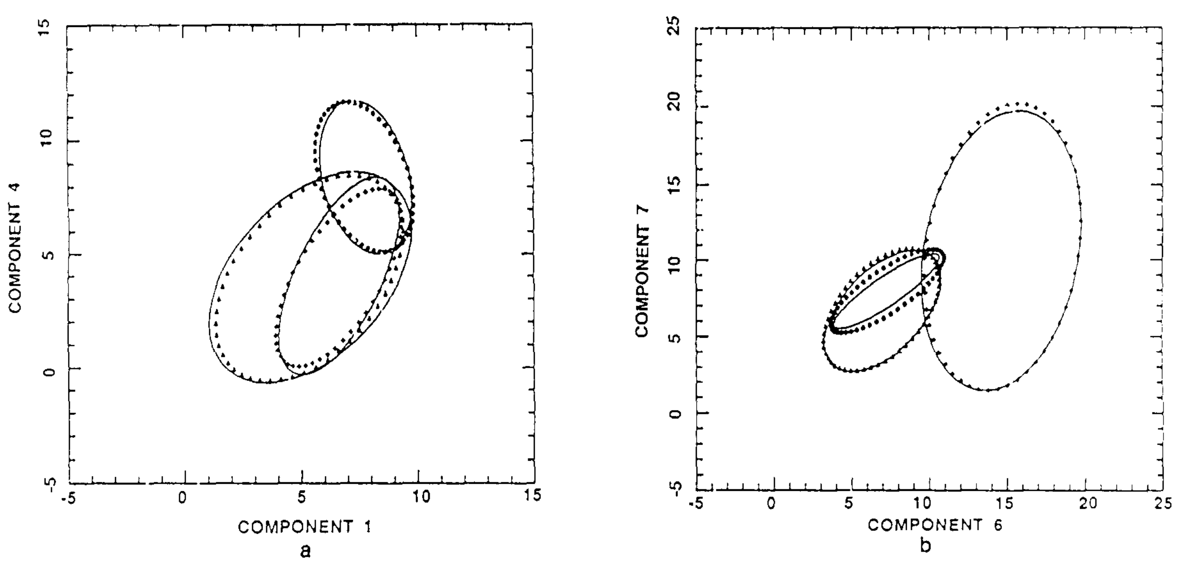

FIGURE 9. Examples of two-dimensional projections of eight-dimensional, three-class data (example 3). Distributions are shown by illustrating 2-σ boundaries; solid lines are used for true distributions and symbols are used for the distributions estimated using the ML neural network.

mates will be poorer for a given number of objects. If a particular class (or type) does not overlap with any other class, the CR bounds for this class are reduced to that of a single-class or a perfectly supervised case.

This last result offers an explanation of the very good neural network performance discussed above in example 3 of the previous section. In this example of unsupervised learning, the neural network performance comes close to the theoretical limit for the supervised case. This can be understood in the following way. First, due to the small overlap between classes, the CR bounds for this unsupervised case are close to the supervised learning CR bounds. And second, the actual learning efficiency of the MLANS has come close to the theoretical limit of the CR bounds.

In this paper CR theory has been exploited as a

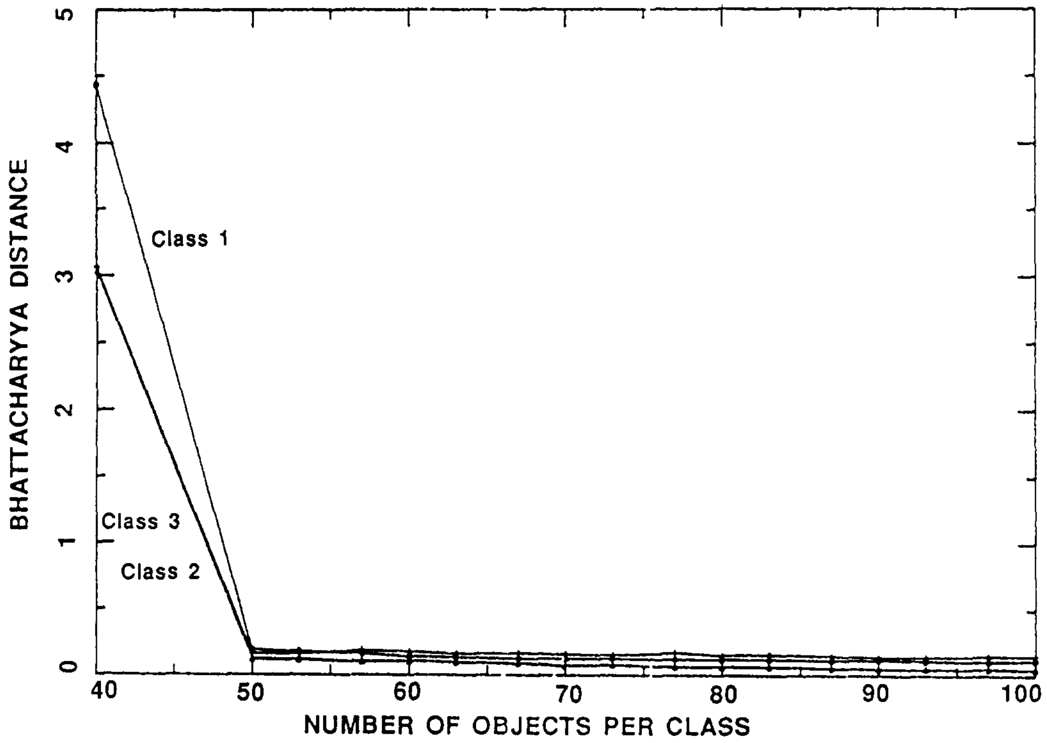

FIGURE 10. Bhattacharyya distances between the true and the estimated distributions for each class; unsupervised learning of three (example 3). The initial gueses are very far from the true distributions; after adaptation very close estimates are achieved with as few as 40 objects from each class.

motivation for the development of the MLANS and, in a somewhat indirect way, as a neural network efficiency-evaluation tool. The future uses of CR bounds in neural network development may include the following. CR bounds are parametrically dependent on various problem parameters, such as the number of classes and types and the dimension of the classification space. Therefore, they can be used for estimating the scaling properties of neural network, when projecting the requirements for large-scale systems. It will be important to understand what can be deduced from the CR bounds for the classification of very-high-dimensional problems such as image classification, where the number of training samples is smaller than the number of pixels, and therefore the CR bounds for the parameters of pixel distribution are infinite if no additional information is used. Which additional information is important, and how should it be used efficiently? The development of CR bounds for learning with partial or imperfect supervision will be necessary to answer this question.

5. RELEVANCE TO BIOLOGICAL AND OTHER NEURAL NETWORKS

Three issues are addressed in this section: Whether the brain estimates metrics in classification spaces; what the temporal dynamics of the learning process at the level of a single neuron and at the whole system level is; and what are the roles of stochastic variability and noise in the MLANS neural network.

Does the brain perform ML estimation and classification? As we have shown above, the ML approach to classification offers competitive advantages in terms of speed of learning and adaptation, optimal information fusion, and simple architecture with local learning mechanisms. Via estimation of covariance matrixes of classes and types of objects, the ML approach achieves an adaptive estimation of local metrics in classification spaces, allowing an adaptive enhancement of even minor differences between the objects that are important for the classification and an adaptive suppression of gross differences that are irrelevant for the classification. There are other mechanisms for achieving the same goals; for example, translational and rotational invariances can be built into the architecture of a neural network. However, adaptive mechanisms are necessary for achieving subtle invariances that are problem dependent, and the ML classification provides an efficient approach for the solution of this problem. Despite these advantages of the ML classification, doubts remain as to how it can be achieved in a biological system. The objection often raised to any biological significance of our results is based on the assumption that biological neurons are not sophisticated enough to calculate covariance matrixes and, even more so, to invert the matrixes. The answer to this question may be related to the results of a recent publication (Perlovsky and Marzetta, 1989) where it is shown that instead of estimating and inverting a covariance matrix it is sufficient to estimate directly the Cholesky decomposition of the inverse covariance matrix. It is shown further that this estimation can be obtained by a linear-estimation procedure similar to the orthogonalization process, which is well suited for biological neural networks, and which have been postulated by many researchers to take place in the brain (see Grossberg, 1983).

Another issue concerning similarity between biological and artificial neural networks is the temporal dynamics of the learning process. The learning mechanism of the MLANS, as specified by eqn (4), is a feedback-competitive learning (a review of learning mechanisms can be found in Carpenter, 1989). Since MLANS weights are probabilities, classes and types "compete" for the assignment probability of each observation. An overall convergence dynamics of the MLANS is similar to the ART neural network (Carpenter & Grossberg, 1987) in that MLANS is converging to a solution by "resonating" between input data and an internal representation. This process accounts for the correlation of a current input with the neuron output at the previous cycle. Such correlation with delay has been considered as a more realistic replacement of the Hebbian learning rule (see Klopf, 1987; Grossberg & Schmajuk, 1989).

The temporal dynamics of the learning process in the ML neural network on a system level exhibit the existence of metastable states with long lifetimes, such as the initial large number of internal cycles shown in Figure 8. Such phenomena have been considered important for an explanation of short-term memory, and they have been explained as long-relaxation-time phenomena near critical points of phase transitions (Kryukov, 1988). In our ML neural network these phase transitions occur when the likelihood function has more than one maximum of approximately the same order of magnitude. When the number of objects available to the MLANS is small, the maximum of a likelihood function may correspond to a different set of parameters than the true one, or even several maxima of approximately equal value may exist.

Such a situation is illustrated in Figure 8, where the transitions between maxima evoke long-living metastable states. The learning process corresponding to Figure 8 can be described as follows: As more information is acquired by the neural network, its initial internal representation of the "world" needs to be adjusted. This need for adjustment is an attentional mechanism that evokes short-term memory (STM here is the metastable state), which is necessary for the modification of the long-term memory (LTM) containing the representation of the world. Similar mechanisms for STM and LTM exist in ATR neural network (Carpenter & Grossberg, 1987).

The further comparison of the ML neural network dynamics to the phase transitions in physical systems calls for the following analogies. The logarithm of the likelihood function is analogous to $-H/T$, where H is the Hamiltonian of the system, and T is the temperature. The degrees of freedom identified with molecules in a physical system are identified here with parameters and neurons that estimate them. That leads to the following identification of the geometry of a phase transition in the ML neural network: The closeness of molecules in a physical system is determined by an interaction between them, which corresponds to inseparable terms in the Hamiltonian. Analogously, parameters of classes, which are inseparable in the classification space, contribute to inseparable terms in the likelihood function. Therefore, the closeness of the parameters of the classes and types of objects is determined by the distance between classes in the classification space (with proper metrics), which is a kind of intuitive result.

The origin of the stochastic properties of the ML neural network is not in the stochastic properties of individual neurons, as the individual neurons do not possess any noisy quality in the current implementation of the MLANS. In that respect the MLANS is different from the Boltzmann machine (Ackley, Hinton, & Sejnowski, 1985), and is similar in spirit to mean field annealing (Bilbro & Snyder, 1988). The role of temperature here is assumed by the covariance matrixes; the large a priori covariance matrixes assure initial randomness, or equal probability of objects to belong to each class, analogous to a high initial temperature in an annealing process. However, no annealing schedule is needed: the iterative estimation of covariance matrices can be compared to local adaptive temperature estimation (Leinbach, 1988).

6. DISCUSSION

The MLANS neural network that we have introduced is an efficiently learning network in that it comes close in its adaptation speed to the theoretical limits of CR bounds for any learning algorithm. This results in very good performance, close to the Bayes risk, with a much smaller number of training samples than is usually required by other neural networks. An issue that we would like to address in the future is optimal classification in very-high-dimensional spaces, when the number of training samples is not sufficient for reliable estimation of classification metrics. This boils down to the old unresolved problem of pattern recognition: How to extract classification features in an optimal way with insufficient information about class distributions. We hope that studying CR bounds for partially or imperfectly supervised classification will help to identify the types of information most useful for solving this problem.

The architecture of the MLANS is much simpler than that of other neural networks. Even though it has no hidden layers, it can still design a classifier of arbitrary shape. The ML neurons and learning mechanism are sophisticated in that they are performing matrix estimation and inversion. This does not represent a problem in digital implementation. For the purpose of future, more detailed comparison to biological systems, and for an analog or optical implementation, if required, these ML neurons can be substituted by subnetworks performing mostly linear operations with fully local learning mechanisms.

The temporal dynamics of the MLANS is intriguing from the biological point of view: Would the phase transitions in this neural network be helpful in understanding such phenomena as short-term memory and attention?

REFERENCES

Ackley, D. H., Hinton, G. E., & Sejnowski, T. J. (1985). A learning algorithm for Boltzmann machines. *Cognitive Science*, **9**, 147–169.

Akaike, H. (1973). Information theory and an extension of the maximum likelihood principle. Petrov and Csaki, Eds. *Proceedings of the Second International Symposium on Information Theory*, Budapest: Akademiai Kido.

Bilbro, G. L., & Snyder, W. E. (1988). Range image restoration using mean field annealing. In *IEEE Conference on Neural Information Processing Systems*, Denver, CO. Piscataway, NJ: IEEE.

Carpenter, G. A., & Grossberg, S. (1987). A massively parallel architecture for a self-organizing neural pattern recognition machine. *Computer Vision, Graphics and Image Processing*, **37**, 54–115.

Carpenter, G. A. (1989). Neural network models for pattern recognition and associative memory. *Neural Networks*, **2**, 243–257.

Cramer, H. (1963). *Mathematical Methods of Statistics*, (10th printing). Princeton, NJ: Princeton University Press.

Fukunaga, K. (1972). *Introduction to pattern recognition*. New York: Academic Press.

Fukunaga, K., & Hayes, R. R. (1988). Statistical classifier design and evaluation. *Purdue University Report TR-EE 88-19*, West Lafayette, IN.

Grossberg, S. (1983). The quantized geometry of visual space: The coherent computation of depth, form, and lightness. *Behavioral and Brain Sciences*, **6**, 625–692.

Grossberg, S., & Schmajuk, N. A. (1989). Neural dynamics of adaptive timing and temporal discrimination during associative learning. *Neural Networks*, **2**, 79–102.

Kaufmann, A. (1975). *Introduction to the theory of fuzzy sets* (Vol. I). New York: Academic Press.

Klopf, A. H. (1987). *A Neuronal Model of Classical Conditioning*, Air Force Wright Aeronautical Laboratories Report AFWAL-TR-87-1139, WPAFB, OH.

Kryukov, V. I. (1988). Short-term memory as a metastable state: "Neurolocator," a model of attention. In *IEEE Conference* *on Neural Information Processing Systems*, Denver, CO. Piscataway, NJ: IEEE.

Leinbach, J. (1988). Automatic local annealing. In *IEEE Conference on Neural Information Processing Systems*, Denver, CO. Piscataway, NJ: IEEE.

Perlovsky, L. I. (1986). Akaike criterion for estimating a number of clusters (unpublished).

Perlovsky, L. I. (1987). Multiple sensor fusion and neural networks. In *DARPA neural network study*. Lexington, MA: MIT/Lincoln Laboratory.

Perlovsky, L. I. (1988a). Neural networks for pattern recognition and Cramer–Rao bounds. *Boston University Engineering Seminar*, Boston, MA.

Perlovsky, L. I. (1988b). Neural networks for sensor fusion and adaptive classification, *First Annual International Neural Network Society Meeting*, Boston, MA.

Perlovsky, L. I. (1988c). Cramer–Rao bounds for the estimation of means in a clustering problem. *Pattern Recognition Letters*, **8**, 1–3.

Perlovsky, L. I. (1989). Cramer–Rao bounds for the estimation of normal mixtures. *Pattern Recognition Letters*, **10**, 141–148.

Perlovsky, L. I., & Marzetta, T. L. (1989). Estimating a covariance matrix in a sensor fusion problem. In *Second Mini Conference on Acoustics Speech and Signal Processing*, Boston, MA.

Titterington, D. M., Smith, A. F. M., & Makov, U. E. (1985). *Statistical analysis of finite mixture distributions*. New York: John Wiley & Sons.

Widrow, B. (1988). Efficiency of adaptive algorithms. In *Twenty-Second Asilomar Conference on Signals, Systems & Computers*, Asilomar, CA.

Yarman-Vural, F., & Ataman, E. (1987). Noise, histogram and cluster validity for Gaussian-mixtured data. *Pattern Recognition*, **20**(4), 385–401.