

Unsupervised learning for nonlinear synthetic discriminant functions

John W. Fisher III and Jose C. Principe

Computational NeuroEngineering Laboratory
University of Florida
Gainesville, FL 32611
fisher@cnel.ufl.edu, principe@cnel.ufl.edu

ABSTRACT

It has been shown in previous work^{5,12} that the family of filters which includes the minimum average correlation energy (MACE) filter⁷ can be formulated as a linear associative memory (LAM)³ preceded by a linear pre-processor which changes depending on the optimization criterion. We have presented a methodology by which the MACE filter and other synthetic discriminant function⁶ (SDF) filters can be extended to nonlinear processing structures⁹ (i.e. nonlinear associative memories) resulting in improved performance with respect to generalization and out-of-class target rejection¹⁰. Our earlier focus was towards developing efficient training algorithms for computing a nonlinear discriminant function without changing the linear pre-processor. In this paper we discuss a nonlinear pre-processing method based on concepts of information theory. We show a simple unsupervised method by which input images can be nonlinearly transformed onto a maximum entropy feature space.

Keywords: correlation filters, neural networks, pattern recognition, unsupervised learning, entropy

1. INTRODUCTION

Many image classification methods can be decomposed, as in figure 1, into two stages: feature extraction followed by discrimination. In some cases, the decomposition is explicit while in others it is a matter of interpretation. As an example, nearly all of the optimized correlation methods⁵ can be decomposed as a pre-whitening filter (feature extraction) followed by an SDF (synthetic discriminant function). The pre-whitening filter is a function of the spectrum of an implicit rejection class⁹ (e.g. MACE⁷ - shifted recognition class exemplars, MVSDF⁸ - random noise process), in which case the rejection class is characterized by its second-order statistics.

In previous work we presented a method by which SDF-type filters could be extended to nonlinear associative memories⁹ resulting in improved generalization and out-of-class target rejection.¹⁰ In that approach the pre-processor remained as a linear pre-whitening filter. We focus now on a nonlinear method of feature extraction. It is based on concepts from information theory, namely mutual information and maximum cross-entropy. The method is unsupervised in the sense that although the feature extraction is extracted via a multi-layer perceptron (MLP), the mapping is determined without assigning an explicit target output, a priori, to each exemplar. Instead the adaptation is driven by a global property of the output: cross entropy.

The motivation for such a method is that we are unable to practically compute a nonlinear discriminant function by fully exploring the input space. Invariably we must project our original input onto a smaller space. In this case it may advantageous to perform the projection such that some measure of information about a specific class is maximized. From our experiments it is evident that the proposed method results in features which appear to be related to the nature of distortions present in the exemplar set.

In section 2 we discuss the concepts upon which our feature extraction method is based. We derive the adaptation method which results in statistically independent features in section 3. Our experimental results are presented in section 4, while our conclusions and observations appear in section 5.

2. BACKGROUND

The method we describe here combines cross entropy maximization with Parzen window probability density function estimation. These concepts are reviewed.

2.1 SELF-ORGANIZATION AND MAXIMUM ENTROPY

Maximum entropy techniques have been applied to a host of problems and scientific disciplines (e.g. parameter estimation, coding theory, etc.). Linsker¹ proposed maximum entropy as a self-organizing principle for neural systems. The basic premise being that any transformation of signals through a neural network should be accomplished so as to maximize the amount of information preserved. Linsker demonstrates this *principle of maximum information preservation* for several problems including a deterministic signal corrupted by gaussian noise.

Entropy (we use the terms *entropy* and *information* interchangeably) has also been proposed as an unsupervised learning method for the generation of topologically ordered feature maps.² The term *topologically ordered feature map* indicates that feature nodes which are “close” in the discrete output space are sensitive to inputs which are “close” in the input space. This work² combines Kohonen’s self-organizing feature map (SOFM)³ with an information theoretic approach to adaptation. The features, in this case, are the excitation of gaussian radial basis functions by an input vector, x ,

$$y_i = \exp\left(-\alpha\|m_i - x\|^2\right) + \sum_{k \in \mathfrak{N}_i} c_{ik} y_k,$$

where m_i , the center of the kernel, is to be adapted. The features are typically mapped onto a discrete structure (e.g. sampled lattice or torus) and local connections in the neighborhood, \mathfrak{N}_i , contribute to the excitation. In this way the topological ordering of the features is maintained. Probabilistic methods are used in determining which kernel center, m_i , is adapted given the current input vector, x . This method is analogous to the method being reported here. Similarities and differences will be described.

As our measure of *information* we use the Kullback-Leibler criterion (also called differential entropy) for continuous RVs (random vector). Given the RV, $Y \in \mathfrak{R}^N$, the Kullback-Leibler criterion is defined as

$$h_Y(u) = - \int_{-\infty}^{\infty} \log(f_Y(u)) f_Y(u) du, \quad (1)$$

where $f_Y(u)$ is the probability density function of the RV, the base of the logarithm is arbitrary, and the integral is N -fold. It is well known that if the random vector is restricted to a finite range in \mathfrak{R}^N the Kullback-Leibler measure is maximized for the *uniform* distribution; a property we will make use of later.

2.2 PROBABILITY SPACE

One difficulty in applying the Kullback-Leibler measure with continuous RVs is that it requires some knowledge of the underlying PDF (probability distribution function). Unless assumptions are made about the form of the density function it is very difficult to use the measure directly. One method by which an estimate of the PDF can be computed is the Parzen window method.⁴ The Parzen window estimate of the probability distribution, $f_Y(u)$, of a random vector $Y \in \mathfrak{R}^{N \times 1}$ at a point u is defined as

$$\hat{f}_Y(u) = \left(\frac{1}{N_y}\right) \sum_{i=1}^{N_y} \kappa(y_i - u). \quad (2)$$

The vectors $y_i \in \mathfrak{R}^N$ are observations of the random vector and $\kappa([\])$ is a kernel function which itself satisfies the properties of PDFs (i.e. $\kappa(u) > 0$ and $\int \kappa(u) du = 1$). Since we wish to make a local estimate of the PDF, the kernel function should also be localized (i.e. uni-modal, decaying to zero). In the technique we describe we will also require that $\kappa([\])$ be differentiable everywhere. In the multidimensional case the form of kernel is typically gaussian or uniform. As a result of the differentiability requirement, only the gaussian kernel will be suitable here.

3. DERIVATION OF GRADIENT DESCENT

As we stated our goal is to find statistically independent features; features that jointly possess minimum mutual information or maximum cross entropy.

Suppose we have a mapping $g: \mathfrak{R}^N \rightarrow \mathfrak{R}^M$ of a random vector $x \in \mathfrak{R}^N$, which is described by the following equation

$$y = g(\alpha, x) \quad (3)$$

How do we adapt the parameters α such that the mapping results in a maximum cross-entropy (which is the same as minimum mutual information) random variable at the output? If we have a desired target distribution then we can use the Parzen windows estimate of the current distribution to minimize the “distance” between the current distribution and the desired distribution. If the mapping has a restricted range (as does the output of an MLP using sigmoidal nonlinearities), the uniform distribution can be used as the target distribution. As we have stated, for a restricted range, the uniform distribution has maximum entropy. It is the only distribution (assuming restricted range) for which the components of a random vector are *statistically independent* (i.e. $f_{Y|X}(y|x) = f_Y(y)$). In our previous work⁹, we relied on features which were orthogonal. It was shown that in a pre-whitened input space, these features were uncorrelated for an implicit rejection class. However, these features are statistically independent only for the *gaussian* random vector case. If we adapt the parameters, α , of our mapping such that the output distribution is uniform, then we will have achieved statistically independent features regardless of the underlying input distribution.

As our minimization criterion we use integrated squared error between our estimate and the desired distribution, which we approximate with a summation.

$$\begin{aligned} J &= \frac{1}{2} \int_{\Omega_Y} \left(f_Y(u) - \hat{f}_Y(u, y) \right)^2 du \\ &= \left(\frac{1}{N_u} \right) \sum_j \frac{1}{2} \left(f_Y(u_j) - \hat{f}_Y(u_j, y) \right)^2 \Delta u \quad y = \{y_1 \dots y_{N_Y}\} \end{aligned} \quad (4)$$

In equation (4), Ω_Y indicates the nonzero region (a hypercube for the uniform distribution) over which the M -fold integration is evaluated. Assuming the output distribution is sampled adequately, we can approximate this integral with a summation in which $u_j \in \mathfrak{R}^M$ are samples in M -space and Δu represents a volume.

The gradient of the criterion function with respect to the mapping parameters is determined via the chain rule as

$$\frac{\partial J}{\partial \alpha} = \left(\frac{\partial J}{\partial \hat{f}} \right) \left(\frac{\partial \hat{f}}{\partial g} \right) \left(\frac{\partial g}{\partial \alpha} \right). \quad (5)$$

It can be shown that

$$\begin{aligned} \left(\frac{\partial J}{\partial \hat{f}} \right) &= \left(\frac{\Delta u}{N_u} \right) \sum_j \left(f_Y(u_j) - \hat{f}_Y(u_j, y) \right) & \left(\frac{\partial \hat{f}}{\partial g} \right) &= \left(\frac{1}{N_Y} \right) \sum_{i=1}^{N_Y} \kappa'(g(\alpha, x_i) - u_j) \\ &= \left(\frac{\Delta u}{N_u} \right) \sum_j \varepsilon_Y(u_j, y) \end{aligned} \quad (6)$$

where $\varepsilon_Y(u_j, y)$ is the computed distribution error over all observations y and $\kappa(\cdot)$ is the kernel used in the Parzen window estimate of the output PDF. The last term in (5), $\partial g / \partial \alpha$, is recognized as the sensitivities of our mapping to the parameters α . Since our mapping is a feed-forward MLP (α represents the weights and bias terms of the neural network), this term can be computed using standard backpropagation. Substituting (6) into (5) yields

$$\frac{\partial J}{\partial \alpha} = \left(\frac{\Delta u}{N_u N_Y} \right) \sum_j \sum_i \varepsilon_Y(u_j, y_i) \kappa'(g(\alpha, x_i) - u_j) \left(\frac{\partial}{\partial \alpha} g(\alpha, x_i) \right) \quad (7)$$

The terms in (7), excluding the MLP sensitivities, become the new error term in our backpropagation algorithm. This adaptation scheme is depicted in figure 2.

Examination of the gaussian kernel and its differential in two dimension illustrate some of the practical issues of implementing this method of feature extraction as well as providing an intuitive understanding of what is happening during the adaptation process. The N-dimensional gaussian kernel evaluated at some u is (simplified for two dimensions)

$$\kappa(y_i - u) = \frac{1}{2\pi\sigma^2} e^{\left(\frac{-1}{2\sigma^2} (y_i - u)^T (y_i - u) \right)} \Bigg|_{\Sigma = \begin{bmatrix} \sigma^2 & 0 \\ 0 & \sigma^2 \end{bmatrix}, N=2} \quad (8)$$

Evaluation of the partial derivative of the kernel (also simplified for the two-dimensional case) with respect to the y_i as observed at the output of the MLP is

$$\begin{aligned} \frac{\partial \kappa}{\partial y_i} &= \kappa(y_i - u) \Sigma^{-1} (u - y_i) \\ &= \left(\frac{e^{\left(\frac{-1}{2\sigma^2} (y_i - u)^T (y_i - u) \right)}}{2\pi\sigma^4} \right) (u - y_i) \Bigg|_{\Sigma = \begin{bmatrix} \sigma^2 & 0 \\ 0 & \sigma^2 \end{bmatrix}, N=2} \end{aligned} \quad (9)$$

These functions are shown in figure 4. The contour of the gaussian kernel is useful in that it shows that output samples, y_i , greater than two standard deviations from the center, u , of the kernel (in the feature space) do not significantly impact the estimate of the output PDF at that sample point. Likewise, the gradient term, is not significant for output samples exceeding two standard deviations from the kernel center. Consequently sample points for the PDF estimate should not exceed a distance of two standard deviations from each other, otherwise, samples caught “in between” do not contribute significantly to the estimate of the PDF. A large number of such samples can cause very slow adaptation.

Recall that the terms in (7) replace the standard error term in the backpropagation algorithm. This term is plotted as a surface in figure 4 minus the PDF error. From this plot we see that the kernels act as either local attractors or repellers depending on whether the computed PDF error is negative (repellor) or positive (attractor). In this way the adaptation procedure operates in the feature space locally from a globally derived measure of the output space (PDF estimate).

The comparison to topologically ordered feature maps can now be made. In previous approaches^{2,3}, the data act as attractors to the kernel centers in the *input* space. The projections inevitably must be linear (although the topological ordering is not necessarily so). Depending on the kernel used, some assumptions must be made about the nature of data clustering in the input space. The method we describe here, makes no such assumptions. The kernels remain fixed in the *output* space. They act as both attractors *and* repellers. The projections can be nonlinear. Finally, this method makes no assumption about clustering of data in the input space.

4. EXPERIMENTAL RESULTS

We have conducted experiments using this method on millimeter-wave ISAR (inverse synthetic aperture radar) data. Two similar vehicles, shown in figure 3, were rotated through an aspect range of 0 to 180 degrees. The depression angle of the radar during data collection was nominally 20 degrees. The resolution in both down range and cross range is approximately 0.22 meters. The fully polarimetric images were processed with a polarimetric whitening filter (PWF)¹³ and then logarithmically scaled to the discrete levels 0 to 255. Training images were extracted at increments of 3.5 degrees for a total of 50 training images from each vehicle. Testing images were extracted at increments of 0.5 degrees (separated from the training images by at least 0.25 degrees and at most 1.75 degrees) for a total of 360 testing images from each vehicle.

The mapping structure we use in our experiments is a multi-layer perceptron with a single hidden layer. There is no reason that additional hidden layers cannot be used, however, we observed that the addition of a second hidden layer increased the training times. In these experiments, the MLP structure we used has 4096 input connections (one for each pixel in the 64 by 64 input images). The input layer is followed by 4 processing elements in the hidden layer, followed by two output elements. It is over the

two output elements that the PDF is estimated using the Parzen window technique. The nonlinearity used was the hyperbolic tangent function, therefore the output range is constrained to $[-1, 1]$.

An important question is over how many samples to estimate the output PDF. As the number of output nodes is increased, the dimensionality of the estimated PDF increases. At more than three output nodes it may become difficult to reliably estimate the output PDF over its entire range. In our experiments the output dimension was two. Several PDF sampling resolutions were used varying from a 21×21 sampling raster at spacings of 0.1 down to a 9×9 sampling raster at spacings of 0.25 (in both dimensions). In general, the results were similar if the standard deviation term of the estimator kernels was set properly. As a rule of thumb we set this term to be twice the value of the spacing (0.2 in the 9×9 raster case). As we can see from figure 4, at one and two standard deviations distance the kernel is reduced to about 50% and 10%, respectively, from its peak value in the center. Using this heuristic for setting the standard deviation term thus enables kernels at the edge of a 5×5 neighborhood to attract samples from the center of the neighborhood. If this parameter is set too small, the condition mentioned earlier occurs; that is, all of the outputs are clustered in a small point between neighboring estimator kernels and gradient information goes to zero.

The main consequence of reducing the sampling raster appeared to be in the sensitivity to the learning rates. Typically the input nodes would be quickly driven to saturation if they were too high. If they were set too low, the convergence rate would slow considerably. This effect was counteracted by using the simple adaptive learning rate rule

$$\mu_k = \begin{cases} (1 + \alpha) \mu_{k-1} & \text{if } \epsilon_k < \epsilon_{k-1} \\ \beta \mu_{k-1} & \text{if } \epsilon_k > \epsilon_{k-1} \end{cases}$$

where μ_k is the learning rate at iteration k , α is the learning growth rate (set in the range 0.01 to 0.05), and β is the learning reduction rate (usually set to 0.5). Furthermore hard limits were also set so that adaptation of the learning rates was over a fixed range. Of course, these setting will generally be problem dependent, but in our experiments going through this process *once* enabled a computational savings of $(21 \times 21) / (9 \times 9)$ PDF estimations, or more than 80% *less* PDF computations for all off the remaining experiments. There are some other areas where we believe computational savings can be obtained which we will be reporting on in the future.

4.1 Single vehicle training

In the first experiment we trained the feature extractor on a single vehicle (upper vehicle in figure 3). We show the mapping of the input images onto the two dimensional output feature space in figures 5, 6, and 7 after 100, 200 and 300 iterations, respectively. The mapping of the images into the feature space are connected in order of increasing aspect. In the latter two plots it is clear the extracted features have begun to fill the output feature space, but have also maintained aspect dependence on the images. We believe that this is evidence that while the method increases the statistical independence of the two output features, it does not decrease the statistical dependency of the output features on the input vehicle class.

4.2 Single vehicle training

In our second experiment we present both vehicles from figure 3 to the network. The training however remains unsupervised, in that the cost function is measured over both classes together. In figure 8 we show the mapping of the vehicles to the output space after 150 (top) and 300 (bottom) iterations, respectively. In the early stages of training, the mapping again exhibits aspect dependence, but no class dependence. However after 300 iterations, class separation is exhibited. In the bottom left plot of figure 8, we have removed the connecting lines in order to show the class separation. It is clear that the separation is far from complete, but we feel it is significant that this was achieved in an unsupervised fashion.

5. CONCLUSIONS/OBSERVATIONS

We have discussed what we believe to be a new method of unsupervised feature extraction. This method unlike the linear method of feature extraction (pre-whitening) is not limited to second-order statistics. In effect, we achieve features which are statistically independent from each other and yet are still, clearly, structurally related to the input. We believe that the experimental results give evidence that this method may be of use in image detection in the presence of distortions. Intuitively, from the results that

we have presented it appears that the features extracted are related to the underlying distortions present in the input imagery (rotation through aspect). We note that in the second experiment the feature was at first aspect dependent and secondly vehicle dependent as the learning progressed. What would happen in general, in the presence of multiple targets cannot be stated as of yet, but we will be reporting our findings in the future.

The method relies on smooth estimation of the output PDF. We would be remiss if we did not acknowledge that PDF estimation is at best an ill-posed problem. However, it is certainly more desirable to attempt such estimation in a reduced space, such as the output of an MLP, than at the input where the problem of dimensionality is much greater. We are also exploring methods by which this process can be simplified or approximated without negatively impacting the overall result: statistically independent features which provide useful information about a class of imagery.

6. Acknowledgments

This research was partially supported by ARPA grant N60921-93-C-A335.

7. References

1. R. Linsker, "Self-organization in a perceptual system.", *Computer*, vol. 21, pp. 105-117, 1988.
2. R. Linsker, "How to generate ordered maps by maximizing the mutual information between input and output signals.", *Neural Computation*, 1, 402-411.
3. T. Kohonen, *Self-Organization and Associative Memory* (1st ed.), Springer Series in Information Sciences, vol. 8, Springer-Verlag, 1988.
4. E. Parzen, "On the estimation of a probability density function and the mode.", *Ann. Math. Stat.*, 33, pp. 1065-1076, 1962.
5. B. V. K. Vijaya Kumar, "Tutorial survey of composite filter designs for optical correlators", *Appl. Opt.* 31 no.23, 4773-4801, 1992.
6. C. F. Hester and D. Casasent, "Multivariate technique for multiclass pattern recognition," *Appl. Opt.* 19, 1758-1761, 1980.
7. A. Mahalanobis, B.V.K. Vijaya Kumar, and D. Casasent, "Minimum average correlation energy filters," *Appl. Opt.* 26 no. 17, 3633-3640, 1987.
8. B. V. K. Vijaya Kumar, "Minimum variance synthetic discriminant functions," *J. Opt. Soc. Am. A* 3 no. 10, 1579-1584, 1986.
9. J. Fisher and J.C. Principe, "A nonlinear extension of the MACE filter," *Neural Networks*, January 1996.
10. J. Fisher and J.C. Principe, "Experimental results using a nonlinear extension of the MACE filter," *Optical Pattern Recognition VI*, vol. 2490, pp. 41-52, Proceedings of SPIE, Orlando, April 1995.
11. J. Fisher and J. C. Principe, "Formulation of the MACE Filter as a Linear Associative Memory", *Proceedings of the IEEE International Conference on Neural Networks*, Vol. 5, p. 2934-2937, 1994.
12. L. M. Novak, M. C. Burl, And W. W. Irving, "Optimal Polarimetric Processing For Enhanced Target Detection," *IEEE Trans. Aerospace And Electronic Systems*, Vol. 29, p. 234, 1993.

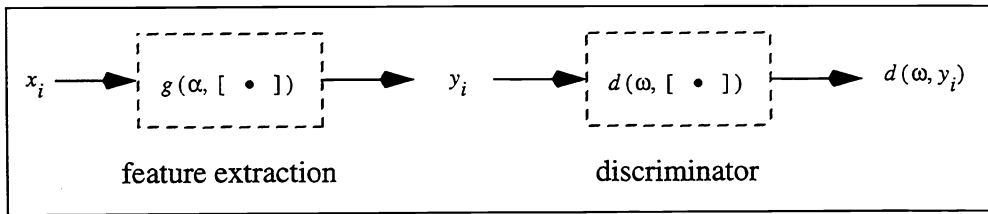


Figure 1. Image classification approach, feature extraction followed by a scalar discriminant function.

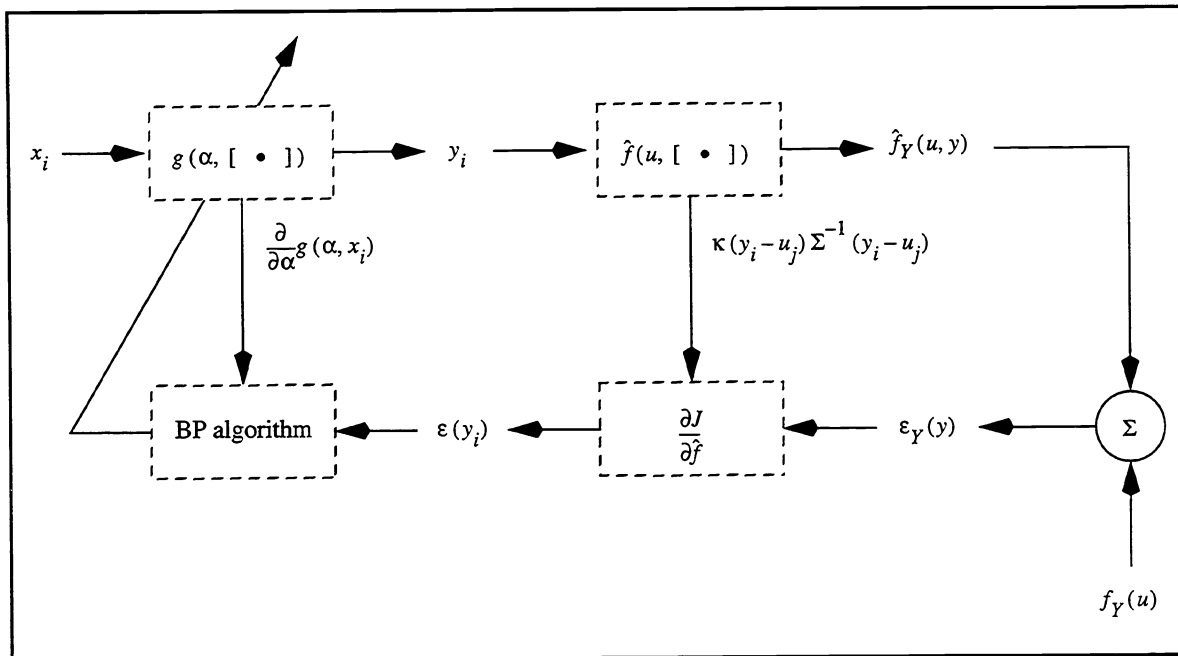


Figure 2. Block diagram of PDF driven adaptation scheme

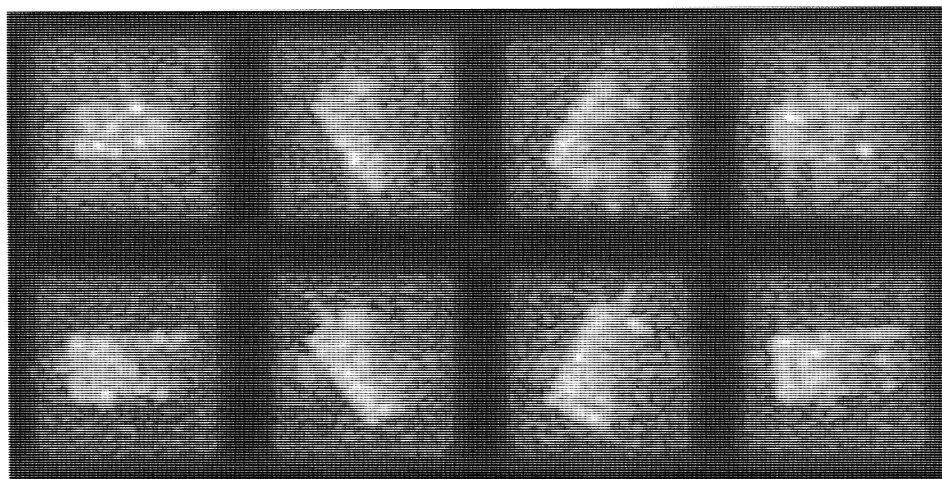


Figure 3. Example ISAR images from two vehicles used for experiments. The vehicles were rotated through an aspect range of 0 to 180 degrees. The top and bottom rows are from different vehicles.

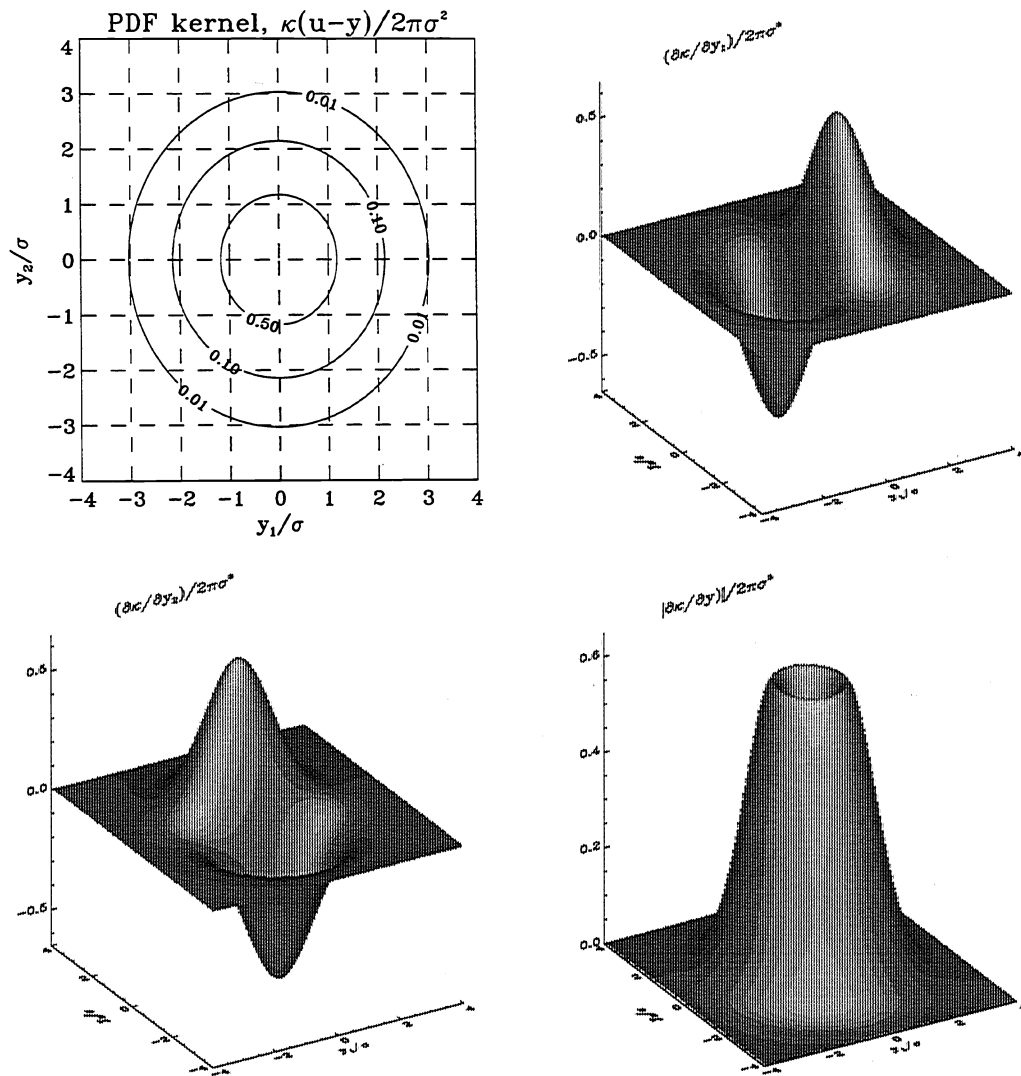
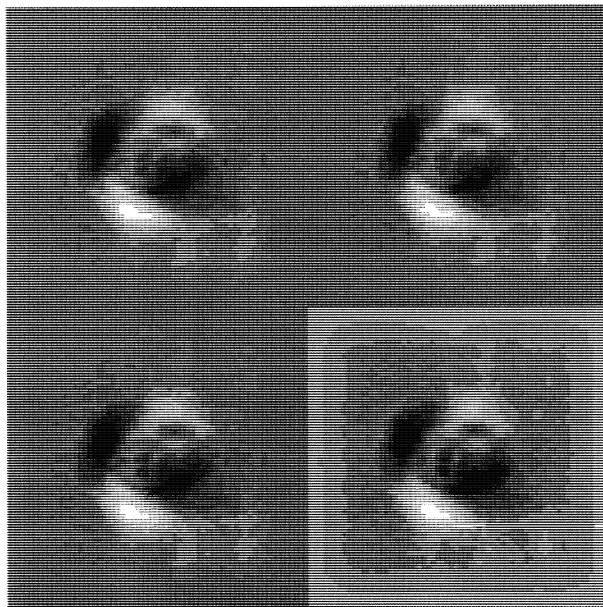
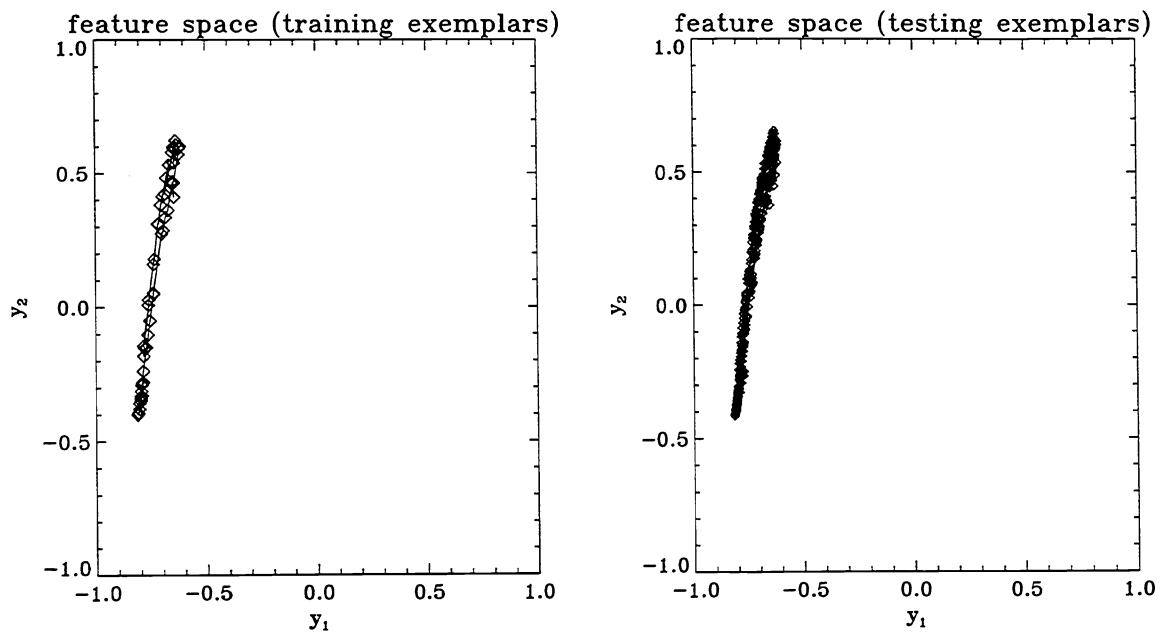


Figure 4. The plots above assume that we are using a two-dimensional gaussian kernel with a diagonal covariance matrix with σ^2 on the diagonals. Contour of the gaussian kernel (top left, normalized by $2\pi\sigma^2$), surface plots of the gradient terms with respect to y_1 (top right), y_2 (bottom left), and magnitude (bottom right) all normalized by $2\pi\sigma^3$. These terms are essentially zero at a distance of two standard deviations.



Hidden layer correlation

f_1	f_2	f_3	f_4
1.00	0.88	0.99	0.99
	1.00	0.87	0.86
		1.00	0.99
			1.00

Figure 5. Projection of training (top left) and testing (top right) images onto feature space for first vehicle after 100 iterations. Images of the correlators are shown at the bottom left, while the correlations of the output of the first hidden layer are shown in the inset table. At this stage of the training the first feature has begun to disperse. We also note the outputs of the first hidden layer remain highly correlated.

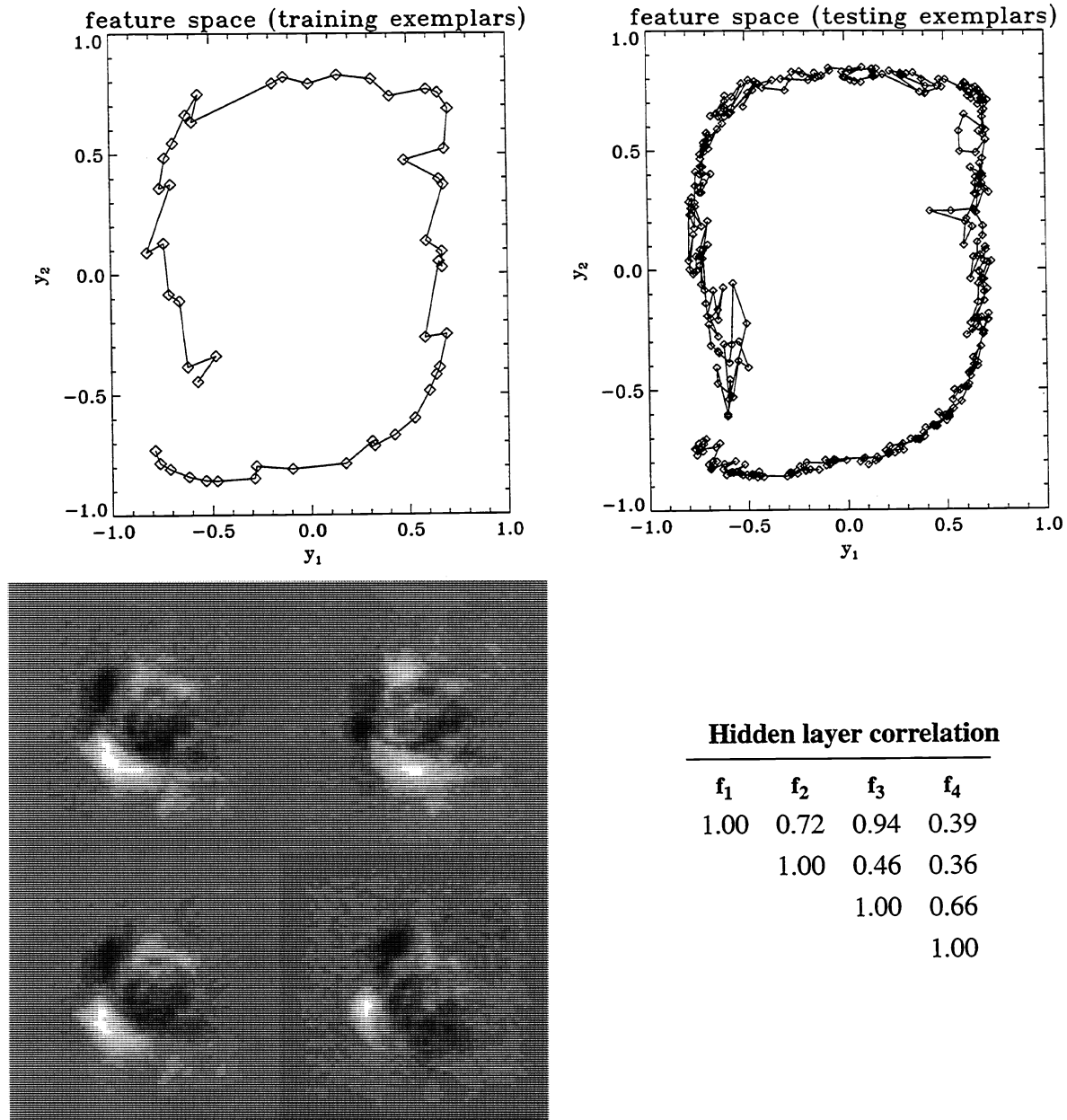
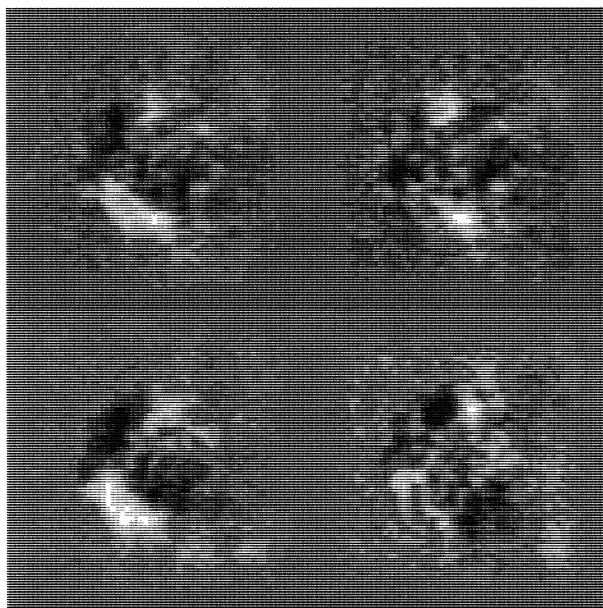
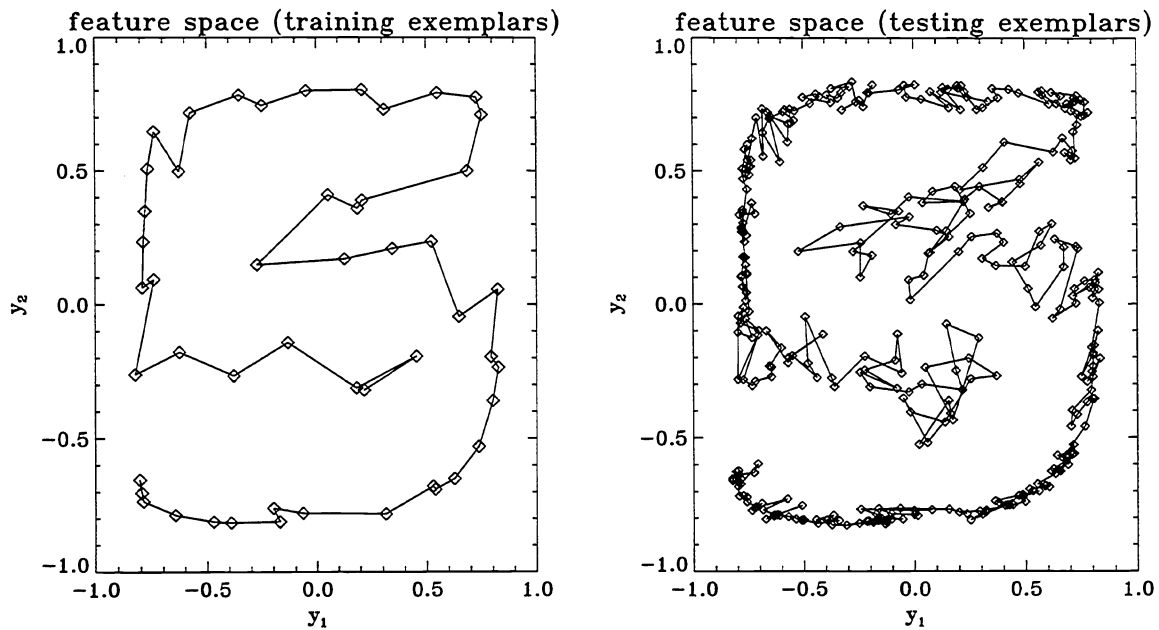


Figure 6. Projection of training (top left) and testing (top right) images onto feature space for first vehicle after 200 iterations of training. Images of the correlators are shown at the bottom left, while the correlations of the output of the first hidden layer are shown in the inset table. The training and testing exemplars are connected by lines in order of increasing aspect angle. This figure seems to show that “closeness” in the input space is maintained in the output space. It is also clear that the feature space generalizes very well for the testing exemplars.



Hidden layer correlation

f_1	f_2	f_3	f_4
1.00	0.75	0.92	0.33
	1.00	0.49	0.29
		1.00	0.66
			1.00

Figure 7. Projection of training (top left) and testing (top right) images onto feature space for first vehicle after 300 iterations. Images of the correlators are shown at the bottom left, while the correlations of the output of the first hidden layer are shown in the inset table. The exemplars continue to increase their coverage in the output space. The features still remain aspect dependent.

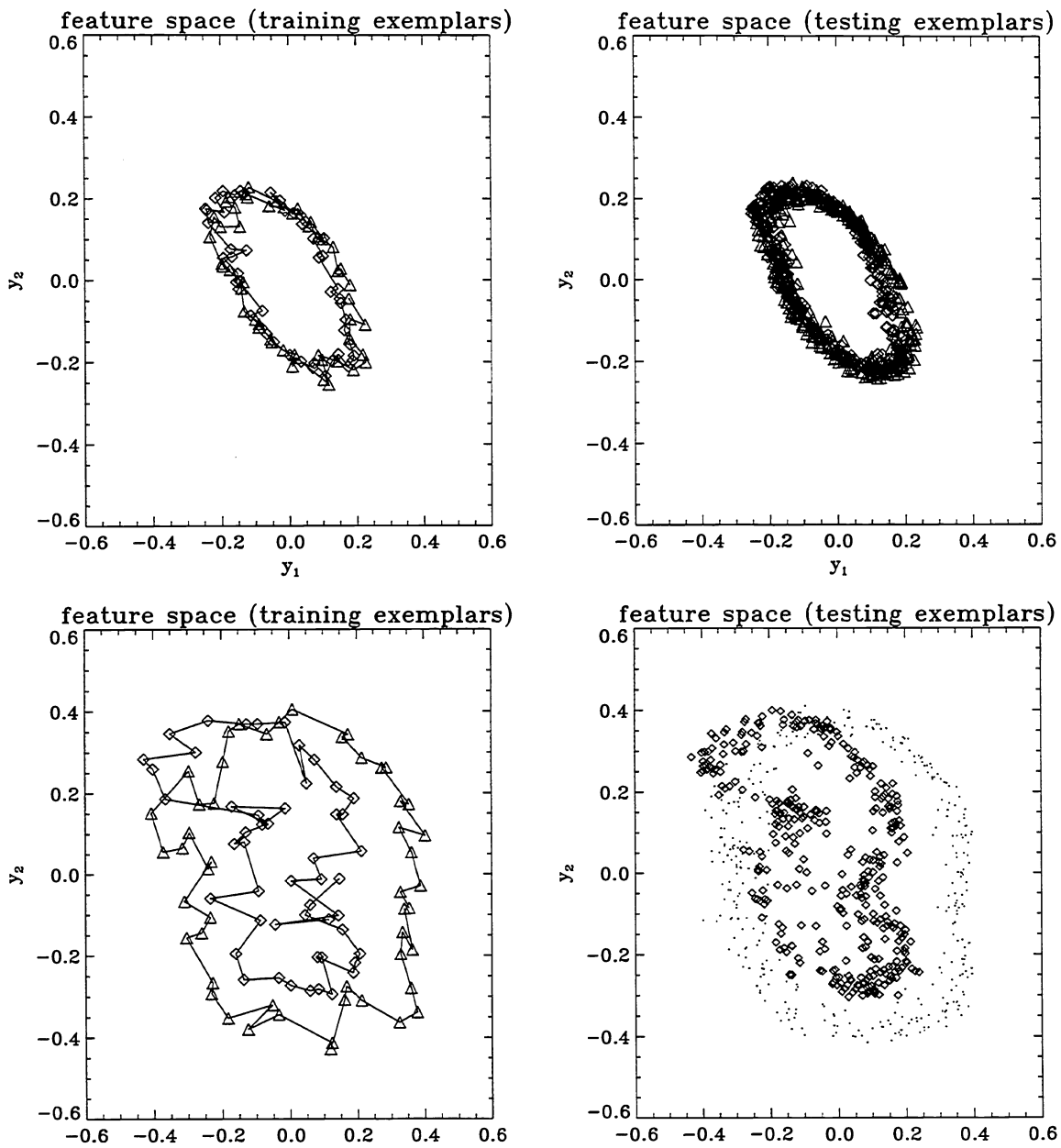


Figure 8. Projection of training (left) and testing (right) images onto feature space after 150 (top) and 300 (bottom) iterations for two vehicle class training. Vehicle 1 is indicated by diamond symbols, while vehicle 2 is indicated by triangles. Each class is connected in order of aspect angle. It appears in these figures that the mapping has maintained aspect dependence for each vehicle. At the 300 iteration point some separation of the vehicles is in evidence. In the bottom left plot, the connecting lines have been removed in order to better show the class separation which has taken place.