INTEGRATION OF NEURAL CLASSIFIERS FOR PASSIVE SONAR SIGNALS

Joydeep Ghosh and Kagan Tumer
Dept. of Elect. and Comp. Engineering,
University of Texas,
Austin, TX 78712-1084

Steven Beck and Larry Deuser
Tracor Applied Sciences
6500 Tracor Lane,
Austin, TX 78725.

Abstract

The identification and classification of underwater acoustic signals is an extremely difficult problem because of low SNRs and a high degree of variability in the signals emanated from the same type of sound source. Since different classification techniques have different inductive biases, a single method cannot give the best results for all signal types. Rather, more accurate and robust classification can obtained by combining the outputs (evidences) of multiple classifiers based on neural network and/or statistical pattern recognition techniques. In this paper, five approaches are compared for integrating the decisions made by networks using sigmoidal activation functions exhibiting global responses with those made by localized basis function networks. These methods are compared using realistic oceanic data. The first method uses an entropy-based weighting of individual classifier outputs. The second is based on combination of confidence factors in a manner similar to that used in MYCIN. The other three methods use simpler techniques of majority voting, averaging, and density estimation with little extra computational overhead. The results indicate that evidence integration provides significant gains when networks are trained on qualitatively different feature sets. Integration techniques also provide a basis for detecting outliers and "false alarms."

1 Introduction

Artificial neural network (ANN) approaches to problems in the field of pattern recognition and signal processing have led to the development of various "neural" classifiers using feed-forward networks [Lip89]. These include the Multi-Layer Perceptron (MLP) as well as kernel-based classifiers such as those employing Radial or Elliptical Basis Functions (EBFs) [BL88, MD89]. These networks can serve as non-parametric, adaptive classifiers that learn through examples [Lip89], without requiring a good *apriori* mathematical model for the underlying signal characteristics.

If the (noisy) input signals representing different classes have substantial overlap in their density distributions, then the statistically optimal expected correct classification achieved by Bayes decision theory can be significantly less than 100% [DH73]. For real-life, difficult signals such as the underwater acoustic signals considered in this paper, the apriori class distributions are not known, and thus Bayesian *a posteriori* probabilities are not immediate. However, recent results show that that multilayer feedforward networks trained by minimizing either the expected mean square error (MSE) or cross-entropy, and by using a 1 of M teaching function, yield network outputs that estimate posterior class probabilities [Gis90, RL91, SCSP91]. These estimations have been observed to be very good for low-dimensional input patterns, at least in regions where there are sufficient training patterns [RL91, SCSP91]. Detailed experiments for high-dimensional patterns have not been reported yet, but they should also give good estimates of aposteriori class probabilities if the cost function is reduced to a sufficiently low value during training, and if the training data reflect true likelihood distributions.

The idea of integrating knowledge from disparate sources has been explored in several fields. Particularly notable are the Dempster-Shaefer theory of evidence and evidence combination in machine learning literature [GLF81], and multisensor fusion in pattern recognition. In the context of feedforward networks, interpretation of network outputs as Bayesian probabilities provides a sound basis for combining the results from multiple classifiers to yield more accurate classification. This idea was first explored by us in [GBC92] wherein we examined several techniques for combining such "evidences", and compared their efficacies for classifying underwater acoustic signals. The results clearly showed that using multiple classifiers provides better and more robust classification decisions. Multiple classifier integration has also been suggested for automated cytology screening [LHDN91], using an approach based on the theory of binary decision trees [Shl90]. The concept of stacked generalization, an inductive approach to combining generalizers, has been recently introduced by Wolpert [Wol92]. Another aspect of getting more robust classification results when using a gradient-descent based network such as the MLP is explored in [KL91]. Their work uses a bank of MLPs that are identical except that they use different initial values of weights, for 3-D object classification and pose estimation. Even the direct use of a second MLP network to combine the outputs of diverse classifiers has shown statistically significant improvement in prediction of protein secondary structure [ZMW92].

In this paper, we focus again on passive sonar signals. This problem domain is of special interest because of the high dimensionality of the input patterns, and the non-trivial decision boundaries demanded. Also, they allow the study of various rejection strategies for countering outliers and false alarms. In the following section, we outline the ANN classifiers used for the experiments. Section 3 describes the database of underwater biologic signals used, and related issues of signal preprocessing and feature-extraction. Experimental results obtained by individual classifiers on this data set are presented in Section 4. The following sections introduce and evaluate various techniques to combine the evidences obtained from distinct classifiers for more accurate and robust results. The work presented in this paper is part of a larger project on the design of a detection and classification system that uses a hybrid of ANN and statistical pattern recognition techniques tailored to recognizing short-duration oceanic signals [Gho91, GBC92, GDB92, BG92].

2 Overview of ANN Classifiers Used

Our experiences, corroborated by those of several other researchers (see [NL91] for example), show that classification error rates are similar across different ANN classifiers when they are powerful enough to form minimum error decision regions, if they are properly tuned, and when sufficient training data is available. Practical characteristics such as training time, classification time and memory requirements, however, can differ by orders of magnitude [GH89]. Also, the classifiers differ in their robustness against noise, effects of small training sets, and in their ability to handle high-dimensional inputs [BG92]. A good review of probabilistic, hyperplane, kernel and exemplar-based classifiers that discusses the relative merit of various schemes within each category, is available in [Lip89, NL91]. Comparisons between these classifiers and conventional techniques such as decision trees, K nearest neighbor, Gaussian mixtures, and CART can be found in [NL91, WK91]. For this study, we chose the MLP and the EBF network, since these classifiers are comparatively insensitive to noise and to irrelevant inputs [BG92].

2.1 MLP.

The standard fully-connected MLP network that adapts weights using "backpropagation" of error is perhaps the most commonly used static ANN classifier [Lip89]. The first network used is an MLP which is trained using the back-propagation rule with momentum. A decaying learning rate is used. Though we have also experimented with various pruning techniques such as "Optimal Brain Damage" [CDS90], the results were not statistically different from a fully connected MLP of appropriate size, and hence are not reported here.

2.2 Localized Basis Function Networks.

The second network used for this study belongs to a class of single hidden-layer feedforward networks in which localized basis functions are used as the activation functions for the hidden layer units. The output of the ith output node, $f_i(\mathbf{x})$ when input vector $\mathbf{x}$ is presented, is given by :

$$f_i(\mathbf{x}) = \sum_j w_{ij} R_j(\mathbf{x}), \tag{1}$$

where $R_j(x)$ is a suitable local basis function that defines the output of the jth hidden node, $\mathbf{x}_j$ is the location of the jth centroid, where each centroid is a kernel/hidden node, and w_{ij} is the weight connecting the jth kernel/hidden node to the ith output node.

A well known example is the Radial Basis Function (RBF) network wherein a radially symmetric function is chosen for $R(x)$, i.e., $R_j(\mathbf{x}) = R(\|\mathbf{x}-\mathbf{x_j}\|)$ [CGDB91, MD89]. If Gaussian functions are chosen as the basis functions, we have

$$R_j(\mathbf{x}) = e^{-\frac{1}{2}\frac{\|\mathbf{x}-\mathbf{x_j}\|^2}{\sigma^2}}$$

where σ determines the width of the receptive field. The j hidden node has a maximum output of 1 when input $\mathbf{x} = \mathbf{x_j}$. Also, contours of constant output value are hyperspheres centered at $\mathbf{x_j}$, as is expected from a radially symmetric function. Since RBFs are equally responsive in all input dimensions, they are not so practical and efficient for high dimensional inputs. This is because a much larger number of kernel nodes are required to cover higher dimensional spaces with adequate resolution [HK91]. Moreover, they are more sensitive to noise and to irrelevant inputs [BG92].

The abovementioned drawbacks can be countered by using Elliptical Basis Function (EBF) networks that employ Gaussian basis functions but with different variance, σ, in different dimensions. Thus, for an EBF network, Eq. 1 is used with

$$R_j(\mathbf{x}_p) = e^{-\frac{1}{2}\Sigma_k \frac{(x_{pk}-x_{jk})^2}{\sigma_{jk}^2}}. \tag{2}$$

The EBF network is a type of Gaussian Potential Function Network [LK88] which involves segmentation of the input domain into several potential fields in form of Gaussians. The Gaussian potential functions of this scheme need not be radially symmetric functions. Instead, the sizes of these potential fields are determined by a correlation matrix. For the EBF, the off-diagonal elements of this matrix are zero, thus reducing the number of free variables used.

The network parameters of EBF are computed by gradient descent. Consider a quadratic error function, $E = \sum_p E_p$ where $E_p = \frac{1}{2}\sum_i (t_i^p - f_i(\mathbf{x_p}))^2$. Here t_i^p is the target function for input $\mathbf{x_p}$ and f_i is as defined in Eq. 1. The mean square error is the expected value of E_p over all patterns. The update rules for various network parameters are obtained using gradient descent on E_p:

$$\Delta w_{ij} = \eta_1(t_i^p - f_i(\mathbf{x_p}))R_j(\mathbf{x_p}) \tag{3}$$

$$\Delta x_{jk} = \eta_2 R_j(\mathbf{x_p})\frac{(x_{pk} - x_{jk})}{\sigma_{jk}^2}(\sum_i (t_i^p - f_i(\mathbf{x_p}))w_{ij}) \tag{4}$$

$$\Delta \sigma_{jk} = \eta_3 R_j(\mathbf{x_p})\frac{(x_{pk} - x_{jk})^2}{\sigma_{jk}^3}(\sum_i (t_i^p - f_i(\mathbf{x_p}))w_{ij}). \tag{5}$$

Classification experiments show that these Elliptical Basis Function Networks require considerably shorter training times compared to RBF networks and typically require a fewer number of kernel nodes. However, training time per epoch is increased since centroids and center widths are also adapted using the generalized delta rule, which is a slower procedure.

How should the widths, $\sigma_{jk}s$ be initialized? For RBF, the initial positions of the centroids are typically obtained by k-means clustering, and the width σ_j for the jth centroid is of the same order as the distance between this unit and the nearest centroid, $\mathbf{x_{j^*}}$. This suggests the initialization

$$\sigma_{jk}(init) = \sigma \times \|\mathbf{x_j} - \mathbf{x_{j^*}}\|, \quad \forall j, k.$$

However, since the spread of data is in general different in different dimensions, an initialization given by:

$$\sigma_{jk}(init) = n^{1/2}\sigma \times \|\mathbf{x_{jk}} - \mathbf{x_{j^*k}}\|, \quad \forall j, k.$$

seems more appropriate, where $\sigma = O(1)$ determines selectivity, and $n^{1/2}$ is a normalization term for n-dimensional inputs so that the average variance is σ_j^2, as before. The latter initialization strategy is used in this paper.

3 Data Description and Representation

The data used for the experiments were based on two sets of feature vectors, both of which were extracted from short–duration passive sonar signals due to four types of naturally occurring oceanic sources. The first feature set, FS1, consisted of 25–dimensional feature vectors and the second set, FS2, consisted of 24–dimensional feature vectors. The signal source type and number of training and test samples are given in Tables 1 and 2.[1]

[1] Note that the test sets are extracted from the same raw data.

Table 1: Description of Data for Feature Set 1 (FS1).

Class	Description	Training	Testing
1	Porpoise Sound	116	284
2	Ice	116	175
3	Whale Sound 1	116	129
4	Whale Sound 2	148	235
Total		496	823

Table 2: Description of Data for Feature Set 2 (FS2).

Class	Description	Training	Testing
1	Porpoise Sound	142	284
2	Ice	175	175
3	Whale Sound 1	129	129
4	Whale Sound 2	118	235
Total		564	823

Signal preprocessing as well as the extraction of good feature vectors is crucial to the performance of the classifiers[BDSW91]. For FS1, the 25-dimensional feature vectors extracted from the raw signals consist of :

- 16 coefficients of Gabor wavelets—a multiscale representation that does not assume signal stationarity[CGE89],
- 1 value denoting signal duration, and
- 8 other temporal descriptors and spectral measurements.

For FS2, the 24–dimensional feature vectors extracted from the raw signals consist of:

- 10 reflection coefficients corresponding to the maximum broadband energy segment, using a short term correlation estimate in the decision time–window,
- 10 reflection coefficients, obtained using a long time window spanning the entire time-window,
- 3 temporal descriptors derived from different noise thresholds, and
- 1 value denoting signal duration.

The preprocessing techniques and choice of the feature vectors reflect the following two goals:

- removal or normalization of all effects that vary, but not as a result of the events of interest, to the greatest possible extent; and
- the presence of each type of signal of interest should result in a measurable difference in the observable features.

4 Performance of Individual Classifiers

For each of the two feature sets, two different classifiers were used. These were

- C1 = MLP with a single hidden layer of 50 nodes.
- C2 = EBF with 60 centroids.

Several different sizes for each classifier were tested before settling on the appropriate ones. Each of the networks were "fully trained," i.e. were trained until the classification rate on the test set reached its maximum. The training samples are indicated in the first column of Tables 1 and 2. The networks were then tested using the samples from the second column in each of those Tables. The classification results and the confidence intervals (CI) for each individual classifiers on both feature sets (averaged over 10 runs, with different random initial weights) are presented in Table 3. These results point to three quick conclusions:

- FS1 represents the actual data better than FS2.
- C1 outperforms C2.
- The FS2-C2 combination has difficulties capturing the essential properties of the data.

Table 3: Results for Individual Classifiers.

Classifier/ Feature set	% Correctly Classified	Standart Deviation	95% Confidence Interval
C1/FS1	92.66	.63	92.21–93.11
C1/FS2	88.60	.73	88.08–89.12
C2/FS1	91.30	.64	90.84–91.76
C2/FS2	82.02	2.22	80.43–83.61

While the overall performance is comparable among the two classifiers for FS1, there is a significant difference for FS2. This can be explained by the fact that FS2 may have less local

Table 4: Confusion Matrix for C1 with FS1.

	decision			
	1	2	3	4
Class				
1	268	5	6	5
2	2	157	3	13
3	2	4	117	6
4	5	0	6	224

Table 5: Confusion Matrix for C1 with FS2.

	decision			
	1	2	3	4
Class				
1	254	18	12	0
2	7	158	5	5
3	3	8	115	3
4	8	12	7	208

information, and therefore not lend itself to a classifier that takes advantage of the local aspects of the data. This type of difficulty may manifest itself during the placement of the centroids, and may prevent the centroids from representing the input space with sufficient precision.

The classification results on the test set are displayed in Tables 4–7 using confusion matrices.[2] For each technique, the correct class is given by the horizontal row label and the misclassifica-

[2]The results in Table 3 are based on averages, whereas the confusion matrices in Tables 4–7 are based on sample runs. Therefore there may be slight differences in classification percentages.

Table 6: Confusion Matrix for C2 with FS1.

	decision			
	1	2	3	4
Class				
1	274	4	3	3
2	2	150	5	18
3	2	5	115	7
4	6	4	11	214

Table 7: Confusion Matrix for C2 with FS2.

	decision			
	1	2	3	4
Class				
1	216	37	17	14
2	5	123	11	36
3	2	7	111	9
4	4	13	6	212

tion (or confused) class assigned is given by the vertical column label. The numerical entries give the number of vectors in each class that is classified as belonging to each decision region. Elements on the diagonal represent correct classifications, whereas elements off the diagonal represent classification errors. For example, in Table 4 there were 13 vectors belonging to class 2 that were erroneously classified as belonging to class 4. Such a display format exposes which signals cause the greatest problems for a given technique. Table 7, for example points to the potential source of the problem for using C2 with FS2: There is a disproportionate amount of patterns from class 1 that are classified as belonging to class 2, and a similar number of patterns from class 2 that are classified as belonging to class 4. That information can be used in directing new training strategies, by either focusing on those centroids or classes that cause the most problems.

5 Evidence Integration and Decision Making

It has been recently shown that training multilayer feedforward networks by minimizing the expected mean square error (MSE) at the outputs and using a 0/1 teaching function yields network outputs that approximate posterior class probabilities [Gis90, RL91, SCSP91]. In particular, the MSE is equivalent to

$$MSE = K_1 + \sum_c \int_x D_c(x)(P(c/x) - f_c(x))^2 dx \tag{6}$$

where K_1 and $D_c(x)$ depend on the class distributions only, $f_c(x)$ is the output of the node representing class c given an input x, $P(c/x)$ denotes the posterior probability, and the summation is over all classes. Thus, minimizing the (expected) MSE corresponds to a weighted least squares fit of the network outputs to the posterior probabilities.

This result gives a sound mathematical basis for interpreting the network outputs as probabilities, and for using an integrator to combine the outputs from multiple classifiers to yield a more accurate classification. For very low values of MSE, $f_c(x)$ approximates $P(c/x)$ according to Eq. 6. Let $f_{c,i}(x)$ be the output of the node that denotes membership in class c in the ith neural classifier. We expect that, for all i, x [Mak91]:

$$\sum_c f_{c,i} \simeq 1. \tag{7}$$

Similarly, if the posteriori estimate is very good, one would expect for all c, i:

$$\frac{1}{N}\sum_{j=1}^{N} f_{c,i}(\mathbf{x}_j) \simeq P(c), \tag{8}$$

where j indexes the N training data samples, and P(c) is obtained by counting the frequency of class c in the training data. Indeed, both Eq. 7 and Eq. 8 are observed to be quite accurate for signals that are classified with high confidence by all the three classifiers.

Note that $f_{c,i}(\mathbf{x}_j)$ can be interpreted as the confidence of membership for a specific input sample, $\mathbf{x}_j$. The *class* confidences for each class can be derived for different classifiers by the following rule [Shl90]:

$$g_{c,i} = \frac{n_{c/c,i} + 1}{(\sum_k n_{k/c,i}) + 2}, \tag{9}$$

where $n_{k/c,i}$ is the number of input samples in the training set that belong to class c but are labeled as class k by the ith classifier. By adding 1 and 2 in the numerator and denominator respectively, we ensure that $0 < g_{c,i} < 1$, and is particularly helpful if the training set is small.

By combining the concept of class confidence with the interpretation of the outputs as *aposterior* class probabilities, an overall classification confidence, $y_{c,i}(x)$ is obtained for each sample and for each classifier:

$$y_{c,i}(x) = g_{c,i} \times f_{c,i}(x) \tag{10}$$

Our goal is now to combine the $y_{c,i}(x)$s from individual classifiers to obtain an overall confidence across all classifiers. Two methods for such evidence combination are proposed:

5.1 Entropy-Based Integrator (ENT).

In this method, a weighted average of the outputs of n different classifiers is first performed, with a larger entropy resulting in a smaller weight. The integrator then selects the class corresponding to the maximum value, providing this value is above a threshold. Otherwise, the input is considered as a false alarm, since there is no strong evidence that it belongs to any of the n classes.

First, for every classifier, the entropy is calculated using normalized outputs $y'_{c,i} = \frac{y_{c,i}}{\sum_c y_{c,i}}$. The weight given to each classifier (normalized) output differs from sample to sample according to the (approximated) entropy at the output of that classifier, as follows:

$$H(c) = \frac{1}{n}\sum_{i=1}^{n} \frac{y_{c,i}}{-\sum_c y'_{c,i}\ ln\ y'_{c,i}};$$

$$\text{assigned class label} = \text{c} : max\ H(c). \tag{11}$$

In this way, the outputs of a classifier with several similar values get a lower weighting as opposed to classifiers which strongly hypothesize a particular class membership.

5.2 Heuristic Combination of Confidence Factors (CF).

This approach is inspired by techniques for parallel combination of rules in expert systems. Certainty factors were introduced in the MYCIN expert system for reasoning in expert systems under uncertainty, and reflect the confidence in a given rule [SB75]. The original method of rule combination in MYCIN was later expressed in a more probabilistic framework by Heckerman [Hec86], and serves as the basis for the method proposed below:

First, the outputs, which are in the range [0,1], are mapped into confidence factors (CFs) in the range [-1,1] using a log transformation. Then, a Mycin-type rule is used to combine the CFs for each class. The advantage of this combination rule is that it makes the result invariant to the sequence in which the different network outputs are combined.

The individual CFs are first obtained using:

$$CF_{c,i} = \log_n((n - 1/n)y_{c,i} + 1/n). \tag{12}$$

For each class c, all positive CFs and all negative CFs are combined separately. The resultant positive and negative CFs are combined in the final step, to obtain a combined confidence CF_c, for each class c. The classification decision is:

$$\text{assigned class label} = \text{c} : max\ CF_c.$$

The equations used for combining the CFs are similar to those used in the original MYCIN [SB75], namely:

$$\begin{aligned} CF(a,b) &= 1-(1-a)(1-b) \text{ if } a,b>0; \\ &= -CF(-a,-b) \quad \text{if} \quad a,b<0; \\ &= a+b, \qquad\qquad \text{otherwise.} \end{aligned} \tag{13}$$

Again, the values of $max\ CF_c$ were much lower for the deviant signals, yielding another metric for detecting false alarms and unknown signals. Indeed, by varying the threshold for the minimum acceptable value for $max\ H(c)$ or $max\ CF_c$, one can obtain a range of classification accuracy versus false alarm rates, and be able to choose a suitable trade-off point.

The other three techniques are straightforward:

5.3 Geometric Mean (GM).

This approach is based on the assumption that the network outputs represent aposteriori probabilities, and that the outputs of the different classifiers are independent. The geometric mean of each classifier is used to determine the class of the vectors. Therefore, the maximum of the geometric means:

$$\overline{y_c} = \sqrt[n]{\prod_i y_{c,i}}$$

is chosen as the correct class, where n is the number of classifiers.

5.4 Averaging (AVE).

This approach simply averages the individual classifier outputs. The maximum of the averaged values:

$$\overline{y_c} = \frac{1}{n} \sum_i y_{c,i}$$

is chosen as the correct class, where n is the number of classifiers.

5.5 Majority Vote (MV).

The last approach is the majority vote, where the class which yields the maximum value in at least two of the three classifiers is chosen. If all three classifiers indicate different classes, then the one with overall maximum output value is used to indicate the correct class.

6 Integration Results

6.1 Combining Fully Trained Classifiers.

First the different classifiers were combined using data from a single feature set, then two feature sets were combined with data from a single classifier and finally both feature sets were combined with both classifiers. All classifier outputs were obtained after the classifiers were fully trained. The combination strategies for partially trained classifiers is addressed in the next subsection.

Table 8: Combining classifiers with a single feature set.

	C1	C2	AVE	CF	ENT	GM	MV
FS1	92.66	91.30	93.03	93.03	92.98	92.88	93.35
FS2	88.60	82.02	89.14	88.70	88.97	88.77	88.60

The combination of two classifiers on a single feature set provides only minimal improvements. Table 8 shows the results of combining C1 and C2 for a single feature set. The results were combined using the arithmetic mean (AVE), confidence factors (CF), entropy (ENT), the geometric mean (GM) and finally a majority vote (MV). [3] The different combining techniques provided almost identical results for FS1, and only slightly varying results for FS2. The averaging method provided a 5% reduction in the error rate for both feature sets, when compared to the results of using C1 alone. In both cases however the improvements were marginal at best, and at the limit of statistical significance (The values were either within, or barely outside the 95% confidence interval).

Table 9: Combining features sets with a single classifier.

	FS1	FS2	AVE	CF	ENT	GM	MV
C1	92.66	88.60	95.24	94.78	95.07	94.63	93.46
C2	91.30	82.02	93.34	93.05	92.59	92.86	91.76

Table 9 shows the results for combining FS1 and FS2 for each classifier. The improvements are much more pronounced when two feature sets are combined, and clearly statistically significant. For C1 there was a 35% reduction in the error rate for the AVE combiner, and for C2 there was a more modest, but still highly significant reduction of 23%.

Finally we combined all four combinations of feature sets and classifiers. The results are presented in Table 10. The error rates were reduced by 39% for the average combiner. Table 11 presents the improvements in the error rates in more detail. The improvements are measured against the best results obtained by a single classifier on a single feature set. For example in the first row of Table 11, improvements are measured against 92.66% correct classification, the result obtained by C1 on FS1.

Tables 12–14 present confusion matrices for the AVE combiner. An important observation here is that the errors are not concentrated in any particular region for the cases where more

[3] For the case with only two combiners, the majority vote is reduced to picking the combiner which provides the highest value.

Table 10: Combining both features sets and classifiers.

AVE	CF	ENT	GM	MV
95.53	94.92	95.12	94.56	94.48

Table 11: Reductions in Error Rates

Combinations	AVE	CF	ENT	GM	MV
C1 and C2 (FS1)	5.0	5.0	4.4	3.0	9.4
C1 and C2 (FS2)	4.7	0.9	3.3	1.5	0.0
FS1 and FS2 (C1)	35.2	28.9	32.5	26.8	10.6
FS1 and FS2 (C2)	23.5	20.1	14.8	17.9	5.3
C1, C2, FS1 and FS2	39.1	30.8	33.5	25.9	24.8

than one feature set is used. Using more than one feature set improves the performance by eliminating systematic errors.

6.2 Combining Partially Trained Classifiers.

The results using fully trained networks confirm the notion that using two or more qualitatively different feature extractors helps to obtain significant improvements on the quality of the solutions. For the application presented in this article, combining two classifiers on a single feature set provided improvements too modest to warrant the increased computation. This situation raises an interesting point, namely the computational overhead of training more than one classifier and then combining the results. This motivates another set of experiments with classifiers trained for only a fifth of the time they were trained previously.

Tables 15–17 show the results on combining such partially trained classifiers. The time

Table 12: Confusion Matrix for C1-C2 combination for FS1 using AVE.

	decision			
	1	2	3	4
Class				
1	277	1	3	3
2	2	164	3	6
3	0	5	124	0
4	3	2	3	227

Table 13: Confusion Matrix for FS1-FS2 combination for C1 using AVE.

	decision			
	1	2	3	4
Class				
1	272	4	4	4
2	2	156	4	13
3	0	5	119	5
4	6	3	7	219

Table 14: Confusion Matrix for FS1-FS2 and C1-C2 combination using AVE.

	decision			
	1	2	3	4
Class				
1	275	2	3	4
2	1	165	3	6
3	0	3	125	1
4	2	3	3	227

required for combining results is negligible compared to the training time of each network. Therefore, the results in Tables 15 and 16 were obtained in only 40 % of the time required for obtaining results in Table 3. We conclude from these results that for a global classifier (C1), and in the presence of more than one feature set, the combination of partially trained networks yields results that are superior to a fully trained network tuned to one feature set. The same however does not hold true for a local classifier, where a partially trained network is not sufficiently "tuned" to the feature vectors to provide data for combinations. Consequently the combination results fall short of the classification rate attained with a single fully trained network. If only a single feature set is available, the accuracy of the combination of partially trained networks does not reach the accuracy of the best fully trained classifier.

The most important result is conveyed by Table 17. Here the total time for training and

Table 15: Combining partially trained classifiers with a single feature set.

	C1	C2	AVE	CF	ENT	GM	MV
FS1	90.88	85.54	91.01	91.13	91.01	91.86	90.89
FS2	85.66	73.50	87.24	87.00	86.39	86.69	87.12

Table 16: Combining features sets with a single partially trained classifier.

	FS1	FS2	AVE	CF	ENT	GM	MV
C1	90.88	85.66	94.65	93.20	94.29	93.92	92.47
C2	85.54	73.50	85.66	85.05	87.97	87.24	83.96

Table 17: Combining both features sets and partially trained classifiers.

AVE	CF	ENT	GM	MV
93.80	93.56	93.68	93.44	92.71

combining is about the same as the average time for "fully" training a single classifier on a single data set, but the results are significantly superior, i.e. are well outside the confidence intervals provided in Table 3.

6.3 Classification Accuracy vs. False Alarms.

For the results reported in sections 6.1 and 6.2, no thresholding was done, so that a classification decision had to be made for every test vector. Since low output values represent low confidences, this information can be used to reject some of the test samples. A simple technique is to check whether the strongest combined confidence for a given sample is greater than a threshold θ. For example, for the AVE technique, we reject a sample if $max(\overline{y_c}) \leq \theta$, while for ENT, a sample is rejected if $max(\overline{H(c)}) \leq \theta$, where $\overline{H(c)}$ is the normalized value of $H(c)$. This type of thresholding can also be performed on the outputs of single classifiers, by simply comparing the largest output to the threshold, and only assigning a class if the output value exceeds this threshold. Otherwise, we say the the pattern has "missed detection."

Table 18: False Alarms: Missed Detection Percentages for AVE combiner.

θ	C1/ FS1	C1/ FS2	C2/ FS1	C2/ FS2	C1/ FS1–FS2	C1–C2/ FS1	C1–C2/ FS1–FS2
0.0	0	0	0	0	0	0	0
0.2	.85	4.98	.49	13.60	.49	.24	0
0.4	1.58	6.68	.97	23.21	1.72	1.81	4.74
0.6	3.28	8.38	3.28	34.51	13.72	7.29	18.11

Tables 18–19 show the percentage of samples rejected and the classification accuracy among the accepted samples for different values of θ, for the AVE combiner. Tables 20–21 show similar results for the ENT combiner. In these tables the second through fifth columns represent the

Table 19: Misclassification Among Detected Vectors for AVE combiner.

θ	C1/ FS1	C1/ FS2	C2/ FS1	C2/ FS2	C1/ FS1–FS2	C1–C2/ FS1	C1–C2/ FS1–FS2
0.0	7.34	11.40	8.70	17.98	4.76	6.97	4.47
0.2	6.74	8.37	8.48	16.32	4.62	6.97	4.47
0.4	6.28	8.11	8.38	14.34	3.74	6.43	2.56
0.6	5.82	8.12	7.71	11.64	1.60	5.13	.69

Table 20: False Alarms: Missed Detection Percentages for ENT combiner.

θ	C1/ FS1	C1/ FS2	C2/ FS1	C2/ FS2	C1/ FS1–FS2	C1–C2/ FS1	C1–C2/ FS1–FS2
0.0	0	0	0	0	0	0	0
0.2	.85	4.98	.49	13.60	1.46	.24	1.74
0.4	1.58	6.68	.97	23.21	2.67	2.19	3.28
0.6	3.28	8.38	3.28	34.51	4.13	5.13	5.47

results for individual classifiers on single data sets. For example, for $\theta = .4$, C1 rejects 1.58% of the samples from FS1, and has an accuracy of 93.72% among detected samples, as opposed to 92.66% for all samples. Similarly combining results from FS1 and FS2 obtained using C1, with a threshold of $\theta = .4$, rejects 1.72% of the samples and increases the classification accuracy of accepted samples to 96.26% (instead of 95.24% for all the samples).

Studying these tables provides an interesting conclusion: One can tune the classifiers to reach a pre–set classification rate, as long as one is allowed to reject certain samples. Therefore, increased classification accuracy can be achieved at the expense of increased rejection rates.

Table 21: Misclassification Among Detected Vectors for ENT combiner.

θ	C1/ FS1	C1/ FS2	C2/ FS1	C2/ FS2	C1/ FS1–FS2	C1–C2/ FS1	C1–C2/ FS1–FS2
0.0	7.34	11.40	8.70	17.98	4.93	7.02	4.88
0.2	6.74	8.37	8.48	16.32	4.43	6.91	4.67
0.4	6.28	8.11	8.38	14.34	3.77	6.42	3.41
0.6	5.82	8.12	7.71	11.64	3.23	5.16	2.88

6.4 Limits on Improvements due to Combination.

An issue that has been so far neglected is the potential limits on the expected improvements. While combining results from different classifiers and/or feature sets, we are combining evidence that stems from a single raw data set. Therefore, certain errors may be too deeply rooted in the data to be eliminated with these methods. To analyze this further, consider the case where a pair of classifiers are used, and thus the classification errors can divided into three categories:[4]

- Only one classifier fails to produces the correct class.
- Both classifiers fail to produce the correct class, but select different (incorrect) classes.
- Both classifiers produce the same incorrect class.

The first type of error is the most common, and can often be corrected by the combiner. The second type of error is more problematic, yet correctable. As long as the output value of the correct class is sufficiently high in both classifiers, a correct decision can be made. It is however, unrealistic to expect the correct class to be have the second highest output value in every case. Therefore, this error count gives a realistic bound on the expected number of errors after combination, i.e. the number of errors that can not be corrected without special conditions. The third type of error is not rectifiable by a combination method, since all the evidence points to a particular erroneous class. Therefore, the number of errors in the third category provides an absolute lower bound on the number of errors after combination—or an upper bound on the correct classification rate.

Table 22: Combination Results and Upper Bounds.

	AVE	CF	ENT	GM	MV	Realistic Bound	Absolute Bound
C1	95.14	94.78	95.07	94.63	93.46	97.21	98.66
C2	93.34	93.05	92.59	92.86	91.76	96.96	98.78
FS1	93.03	93.03	92.98	92.88	93.35	95.02	95.14
FS2	89.14	88.70	88.97	88.77	88.60	92.71	94.78

Table 22 shows the result obtained by the different combiners, and compares them to the mentioned upper bounds [5]. The difference between the AVE combiner and the realistic bound varies between 2–3.6%, even though the actual values are significantly different. Therefore an inspection of errors of different type can provide a guide to the expected combiner error rates.

[4] The method also applies for combining more than two sources, but the specific cases have to be modified.

[5] Here the first column gives the classifier used if two feature sets were combined, and the feature set used if two classifiers were combined.

Furthermore, the number of errors of type 2 can be used in the selection of the feature sets, since significantly different feature sets are expected reduce this type of errors.

7 Concluding Remarks

The results presented in this paper highlight the advantage of using evidence combination techniques for improved classification in a situation when the capabilities of a single classifier are fundamentally limited. They also yield a simple metric for expressing the strength of confidence in a given classification decision. Decisions with low confidence measures indicate either improper feature vector selection, inadequate representation by the networks, or that the corresponding signals are outliers or false alarms.

The performance is relatively similar for all five combination techniques. Simple averaging and using the entropy generally provide slightly better results for the data used in this study. Interestingly, the arithmetic mean provides better results than the geometric mean, even though the geometric mean is qualitatively closer to combining conditional probabilities. One explanation for this discrepancy is the dominating effect of low values. For example, if one network has a very low (say 0.01) value for the correct output, then even if the second network has a high (say 0.8) output, the geometrically combined score is less than 0.09. Thus a class which had a value of .1 in both networks will be erroneously chosen.

The results obtained in this study support the theory that more than one feature set is needed to provide significant gains in the classification accuracy. Using different classifiers on a single feature set did not provide improvements that were statistically significant, and hence the added computational expense was unjustified. The results obtained from partially trained classifiers enforced this view. In the presence of at least two feature sets, partial training and subsequent combination does lead to results superior to that of any single classifier. However, in the absence of different feature sets, the results of combining two partially trained classifiers fall short of that of the best fully trained single classifier.

Finally, the percentage of correct classifiers can be significantly boosted, if a modest percentage of samples are rejected. With the proposed type of thresholding, and in the presence of at least two feature sets, the classification error percentage can be reduced by as much as 30%, by only rejecting 2–3% of the samples. Under similar rejection circumstances, but in the absence of two feature sets, only a 5–10% reduction in the error rate is achieved.

Acknowledgements: This research was supported in part by ONR contracts N00014-89-C-0298 and N00014-92-C-0232. We thank Jim Whiteley, and Russell Still of Tracor Applied Sciences for their contributions at various stages of this project.

References

[BDSW91] S. Beck, L. Deuser, R. Still, and J. Whiteley. A hybrid neural network classifier of short duration acoustic signals. In *Proc. IJCNN*, pages I:119–124, July 1991.

[BG92] S. Beck and J. Ghosh. Noise sensitivity of static neural classifiers. In *SPIE Conf. on Applications of Artificial Neural Networks SPIE Proc. Vol. 1709*, Orlando, Fl., April 1992.

[BL88] D.S. Broomhead and D. Lowe. Multivariable functional interpolation and adaptive networks. *Complex Systems*, 2:321–355, 1988.

[CDS90] Yann Le Cun, J.S. Denker, and S. A. Solla. Optimal brain damage. In *Advances in Neural Information Processing Systems -II*, pages 598–605, 1990.

[CGDB91] S. Chakravarthy, J. Ghosh, L. Deuser, and S. Beck. Efficient training procedures for adaptive kernel classifiers. In *Neural Networks for Signal Processing*, pages 21–29. IEEE Press, 1991.

[CGE89] J.M. Combes, A. Grossman, and Ph. Tchamitchian (Eds.). *Wavelets: Time-Frequency Methods and Phase Space*. Springer-Verlag, 1989.

[DH73] R. O. Duda and P. E. Hart. *Pattern Classification and Scene Analysis*. Wiley, New York, 1973.

[GBC92] J. Ghosh, S. Beck, and C.C. Chu. Evidence combination techniques for robust classification of short-duration oceanic signals. In *SPIE Conf. on Adaptive and Learning Systems, SPIE Proc. Vol. 1706*, pages 266–76, Orlando, Fl., April 1992.

[GDB92] J. Ghosh, L. Deuser, and S. Beck. A neural network based hybrid system for detection, characterization and classification of short-duration oceanic signals. *IEEE Jl. of Ocean Engineering*, 17(4):351–363, october 1992.

[GH89] J. Ghosh and K. Hwang. Mapping neural networks onto message-passing multicomputers. *J. of Parallel and Distributed Computing*, 6:291–330, April, 1989.

[Gho91] J. Ghosh et al. Adaptive kernel classifiers for short-duration oceanic signals. In *IEEE Conf. on Neural Networks for Ocean Engineering*, pages 41–48, August 1991.

[Gis90] H. Gish. A probablistic approach to the understanding and training of neural network classifiers. In *Proc. Int'l Conf. on ASSP, Albuquerque,NM*, pages 1361–1364, April 1990.

[GLF81] T.D. Garvey, J.D. Lowrance, and M.A. Fischler. An inference technique for integrating knowledge from disparate sources. In *Proc. of IJCAI*, pages 319–325, 1981.

[Hec86] D. Heckerman. Probabilistic interpretation for MYCIN's uncertainty factors. In L.N Kanal and J.F. Lemmer, editors, *Uncertainty in Artificial Intelligence*, pages 167–196. North-Holland, 1986.

[HK91] E. Hartman and J.D. Keeler. Predicting the future: Advantages of semi-local units. *Neural Computation*, 3(4):566–78, 1991.

[KL91] A. Khotanzad and J. Liou. Recognition and pose estimation of 3-d objects from a single 2-d perspective view by banks of neural networks. In *Proc. ANNIE 91*, pages 479–484, Nov 1991.

[LHDN91] J. Lee, J.-N. Hwang, D.T. Davis, and A.C. Nelson. Integration of neural networks and decision tree classifiers for automated cytology screening. In *Proc. IJCNN*, pages I:257–262, Seattle, July 1991.

[Lip89] R. P. Lippmann. Pattern classification using neural networks. *IEEE Communications Magazine*, pages 47–64, Nov 1989.

[LK88] Sukhan Lee and Rhee M. Kil. Multilayer feedforward potential function network. In *Proceedings of the Second International Conference on Neural Networks*, pages 161–171, 1988.

[Mak91] J. Makhoul. Pattern recognition properties of neural networks. In *Proc. 1st IEEE Workshop on Neural Networks for Signal Processing*, pages 173–187, Sept 1991.

[MD89] J. Moody and C. J. Darken. Fast learning in networks of locally-tuned processing units. *Neural Computation*, 1(2):281–294, 1989.

[NL91] K. Ng and R.P. Lippmann. Practical characteristics of neural network and conventional pattern classifiers. In *Advances in Neural Information Processing Systems -III*, pages 970–976, 1991.

[RL91] M.D. Richard and R.P. Lippmann. Neural network classifiers estimate bayesian a posteriori probabilities. *Neural Computation*, 3(4):461–483, 1991.

[SB75] E.H. Shortcliffe and B.G. Buchanan. A model of inexact reasoning in medicine. *Mathematical Biosciences*, 23:351–379, 1975.

[SCSP91] P.A. Shoemaker, M.J. Carlin, R.L. Shimabukuro, and C.E. Priebe. Least squares learning and approximation of posterior probabilities on classification problems by neural network models. In *Proc. 2nd Workshop on Neural Networks, WNN-AIND91,Auburn*, pages 187–196, February 1991.

[Shl90] S. Shlien. Multiple binary decision tree classifiers. *Pattern Recognition*, 23(7):757–63, 1990.

[WK91] S. M. Weiss and C.A. Kulikowski. *Computer Systems That Learn.* Morgan Kaufmann, 1991.

[Wol92] D. H. Wolpert. Stacked generalization. *Neural Networks*, 5:241–259, 1992.

[ZMW92] X. Zhang, J.P. Mesirov, and D.L. Waltz. Hybrid system for protein secondary structure prediction. *J. Molecular Biology*, 225:1049–63, 1992.