LETTER Communicated by Sue Becker

Category Learning Through Multimodality Sensing

Virginia R. de Sa
Sloan Center for Theoretical Neurobiology, University of California at San Francisco, San Francisco, CA 94143-0444, U.S.A.

Dana H. Ballard
Department of Computer Science, University of Rochester, Rochester, NY 14627-0226, U.S.A.

Humans and other animals learn to form complex categories without receiving a target output, or teaching signal, with each input pattern. In contrast, most computer algorithms that emulate such performance assume the brain is provided with the correct output at the neuronal level or require grossly unphysiological methods of information propagation. Natural environments do not contain explicit labeling signals, but they do contain important information in the form of temporal correlations between sensations to different sensory modalities, and humans are affected by this correlational structure (Howells, 1944; McGurk & MacDonald, 1976; MacDonald & McGurk, 1978; Zellner & Kautz, 1990; Durgin & Proffitt, 1996). In this article we describe a simple, unsupervised neural network algorithm that also uses this natural structure. Using only the co-occurring patterns of lip motion and sound signals from a human speaker, the network learns separate visual and auditory speech classifiers that perform comparably to supervised networks.

1 Introduction

The ability of humans to form complex categories without explicit supervision has challenged modelers. On the one hand, classification is simpler if more dimensions are available to separate the classes. For example, categorizing cows and horses is simpler if one can also make use of auditory features in addition to visual features. On the other hand, simple clustering of multimodality patterns would prevent adequate performance in the individual modalities, and appropriate density modeling techniques rapidly become infeasible in high dimensions. Also, it is well known that the cerebral cortex competently classifies unimodal stimuli while keeping the different modalities largely separate. Inspired by this, we describe an algorithm that avoids the intractable task of modeling cross-modal associations but uses this useful structure to derive its own internal target signals for classifiers in the individual modalities. The algorithm uses natural and neurophysiolog-

Neural Computation **10, 1097–1117** (1998) © 1998 Massachusetts Institute of Technology

ically plausible one-way connections for information transmission, which distinguishes this approach from backpropagation (Rumelhart, Hinton, & Williams, 1986) and also the unsupervised model of Becker and Hinton (Becker & Hinton, 1992; Becker, 1996). More biologically plausible implementations of the information theoretic approach are given in Phillips, Kay, and Smyth (1995) and Kay, Floreano, and Phillips (1998) but have not been demonstrated on real problems with overlapping classes.

The idea behind the algorithm is to minimize the disagreement between the output decisions of two or more classifiers receiving different forms of input from the same source (see Figure 1). The key insight is that this can be done without directly connecting all the hidden units to each other and without requiring implausible communication of error signals backward along forward connections.

1.1 Classification. A general way of representing sensory inputs is in terms of n-dimensional points, or vectors, groups of which can be represented by prototypes, or codebook vectors. A simple classification border between two such codebook vectors representing different classes is the $(n-1)$-dimensional hyperplane midway between them. With more codebook vectors representing several classes, nonlinear boundaries may be devised by taking the border from the Voronoi tessellation of these prototype points. Each codebook vector is assigned a class label, and patterns are classified as belonging to the class of the closest codebook vector. Class boundaries are then the edges of the Voronoi tessellation that separate codebook vectors of different classes.

In learning algorithms, classification borders are moved indirectly by moving the codebook vectors. Competitive learning (Grossberg, 1976a, b; Kohonen, 1982; Rumelhart & Zipser, 1986) is an unsupervised, biologically plausible (Coultrip, Granger, & Lynch, 1992; Miikkulainen, 1991) way of achieving this for easily separable data clusters, but performs poorly on complicated clusters that are either not well separated or not well approximated by circularly symmetric distributions. More difficult categorization problems can be handled if the correct class of each pattern is known during training. The supervised LVQ2.1 algorithm (Kohonen, 1990) monitors and reduces the number of currently misclassified patterns (but see de Sa & Ballard, 1993a; Diamantini & Spalvieri, 1995). It can be described informally as:

> If the pattern is near a current border, move the codebook vector of the same class toward the pattern and that of the other class away.

The resulting border movement increases the chances of an incorrectly classified pattern being correctly classified on a subsequent trial.

When the labels of the sample patterns are given, the supervised goal (assuming equal costs) is to minimize the probability of misclassified patterns

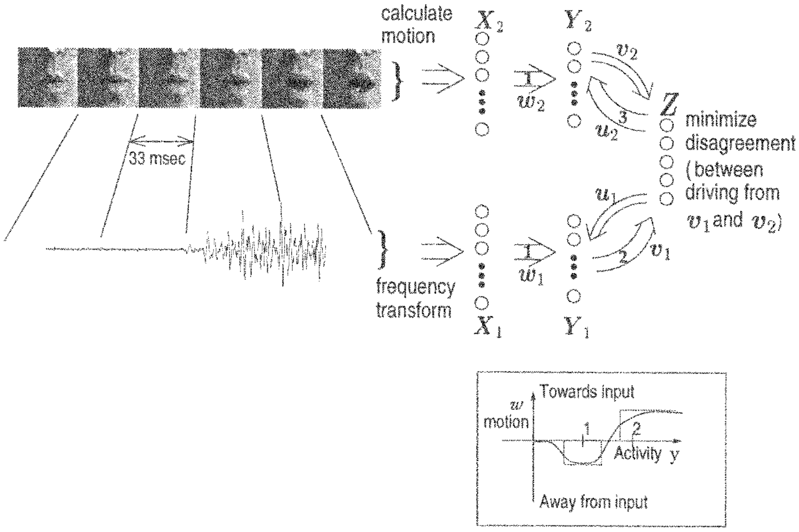

Figure 1: Making use of natural correlational structure. The network learns from the structure inherent in the coherence between the visual and auditory signals. The feature vectors for classification were taken from 10 video frames and 100 msec of auditory signal.

In the network, an arrow between units denotes full connectivity between these banks of units. The variables $\boldsymbol{w}$, $\boldsymbol{u}$, and $\boldsymbol{v}$ represent matrices that store the connection weights. The variables X_i, Y_i, Z store the activity vectors of the various layers. The modalities minimize their disagreement by teaching or "driving" each other. The numbered arrows in the network show the order of propagation of activation when considering the auditory modality driving the visual driven one.

In the driving hidden layer, the hidden unit with the closest weight vector (codebook vector) gets activity 1 (and all others in the layer 0). This 1-of-n activity pattern then serves as the input pattern for the output units, and the output unit with closest weight vector to this activation vector receives activity 1 (and all other output units 0). The two closest weight vectors (to their input pattern) in the driven hidden layer get a forward activity component of 1. The backward weights $\boldsymbol{u}$ are then used to supplement the activity of those driven units receiving forward activation. Due to the binary form of the backward weights $\boldsymbol{u}$, these driven units will have activities of 0, 1, or 2. The inset shows the update direction for the driven codebook vector weights $\boldsymbol{w}$ as a function of their unit's activity level. Positive ordinates represent movement of the weights toward their inputs, and negative values represent movement away. The actual update equations are given in equation 3.1.

for each modality. The goal for each modality is to minimize the number of patterns from each class that fall into Voronoi regions of codebook vectors with other labels. For example, where $P(C_i)$ is the a priori probability for class i and $p(x_j|C_i)$ is the conditional density of the data from modality j from class i, the goal for modality 1 is to minimize (Diamantini & Spalvieri, 1995)

$$E(\{w_{1_i}\}) = \sum_k \sum_i (1 - \delta(L(w_{1_i}), C_k))P(C_k) \int_{V_{1_i}} p(x_1|C_k)\, dx_1. \tag{1.1}$$

Here i is an index over codebook vectors, $L(w_{1_i})$ gives the label of codebook vector w_{1_i}, V_{1_i} represents the Voronoi region around w_{1_i} (the volume closer to w_{1_i} than any other codebook vector), and $\delta(a, b)$ is defined to be 1 when $a = b$ and 0 otherwise. The goal for modality 2 is analogous.

1.2 Cross-Modal Structure. The formulation expressed by equation 1.1 is explicitly supervised in that the estimation of the conditional probabilities depends explicitly on class information. That is, in order to estimate the term $p(x|C_k)$, it is necessary to know which patterns are from class k. An unsupervised error function must depend only on the whole pattern distribution $p(x_1, x_2) = \sum_k P(C_k)p(x_1, x_2|C_k)$. A glance at Figure 2 shows that the structure in the joint feature space is often more informative than that available to either of the individual modalities. One solution this suggests is to perform unsupervised clustering or density estimation in the joint space. However simple k-means clustering, or competitive learning, in the full joint space would require that future patterns for classification contain all feature dimensions; they are not able to marginalize over the missing dimensions. The problem is that while we would like to learn from a joint cross-modal space, we would like, after learning, to be able to act on sensory information from a single modality. Density modeling methods do this and can handle missing features on classification, but they require fitting many parameters, and this is infeasible in high-dimensional spaces.

An architecture that circumvents these problems is shown in Figure 1. The key organizational feature is that each modality has its own processing stream (or classification network) but access to the other's output at a high level.

One way to make use of the cross-modality structure in a network like this is to cluster the codebook vectors (in their individual spaces) but use the joint structure to learn the labels of these codebook vectors. This is a two-stage clustering algorithm. First, the input patterns in each modality are clustered using a competitive learning network. After this, the pattern of activation across the output units of the competitive learning networks (hidden units in Figure 1) can be considered new input patterns for another

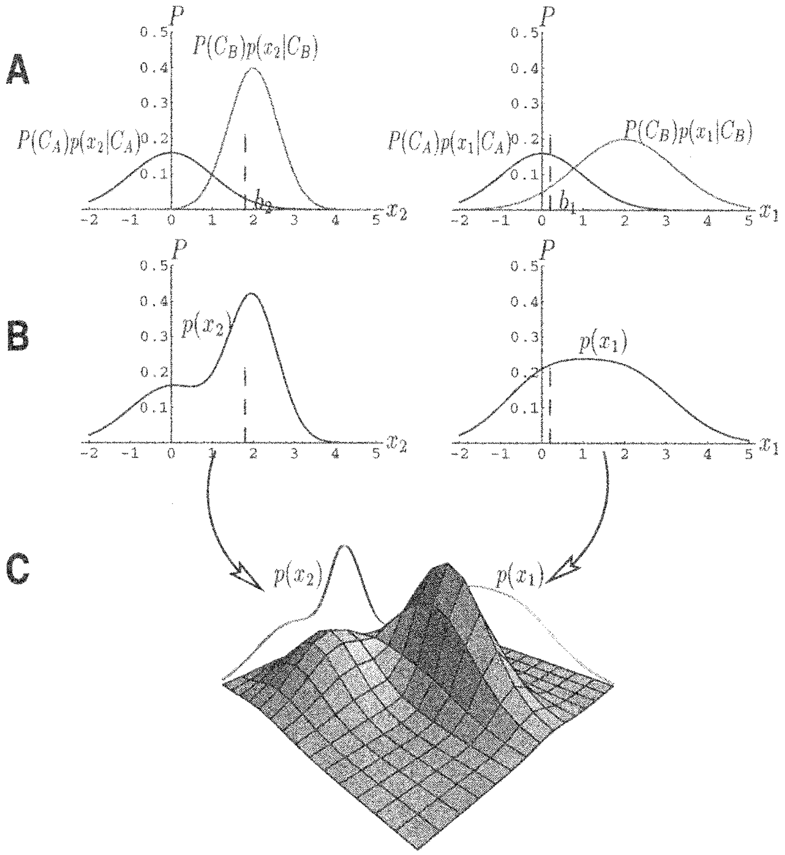

Figure 2: A low-dimensional, two-class example. The graphs on the right side represent the probability distributions of patterns to the first modality; those on the left give the same for the second modality; (A) In the supervised case, the individual density functions associated with each class can be estimated and the appropriate border (at the crossing point of the distributions) found. The darker, left-most curve within each graph represents the distribution of the patterns from class C_A. b_1 and b_2 represent example (but not optimal) classification borders in their respective modalities. (B) In the absence of class information, the computed density function is the sum of the individual class densities. The appropriate border may not be obvious (see, for example, modality 1's density on the right). (C) The higher-dimensional joint distribution $p(x_1, x_2) = P(C_A)p(x_1, x_2|C_A) + P(C_B)p(x_1, x_2|C_B)$ has greater structure and is used to guide the placement of the borders in the individual modalities. The example shows a case where the two variables are conditionally independent, but this is not required.

level of clustering.[1] By assigning labels to the output units of this second clustering stage, each codebook vector is labeled indirectly by the label of the output unit in whose cluster it belongs. This will give the same label to codebook vectors from the different modalities that tend to co-occur.

While this approach is useful, and we use it for initialization (step 2 in the algorithm description in section 2), the constraining structure in the joint distribution can be used more powerfully if it is used for better placement of the codebook vectors themselves.

2 Minimizing Disagreement

The core idea of our codebook placement algorithm is to minimize the disagreement error: the fraction of patterns classified differently by the two networks. The idea is that two modalities, representing different but co-occurring information from the same source, teach each other by finding a local minimum in their output disagreement. This section develops the derivation of the rules for moving the codebook vectors to minimize this error measure. The next shows that one can appropriately move the codebook vectors without directly connecting the codebook vectors to each other and without requiring neurobiologically implausible propagation of information as required in algorithms using backpropagation (Rumelhart et al., 1986) of error signals.

The disagreement error can be written in terms of the codebook vectors as:

$$E(\{w_{1_i}\}, \{w_{2_j}\}) = \sum_i \sum_j (1 - \delta(L(w_{1_i}), L(w_{2_j}))) Pr\{x_1 {\in} V_{1_i}, x_2 {\in} V_{2_j}\}. \quad (2.1)$$

That is,

$$\begin{aligned} E(\{w_{1_i}\}, \{w_{2_j}\}) = {} & \sum_k \sum_i \sum_j (1 - \delta(L(w_{1_i}), L(w_{2_j}))) P(C_k) \\ & \times \int_{V_{1_i}} \int_{V_{2_j}} p(x_1, x_2 | C_k)\, dx_1\, dx_2. \end{aligned} \quad (2.2)$$

Note that equation 2.2 does not depend on the class information but only on the joint density of all inputs (over all classes) and thus can be sampled without labels. It is, however, in the same form as the supervised equation (1.1) dealt with in Diamantini and Spalvieri (1995). Differentiating

[1] While eventually the hidden-layer activation patterns will be of the 1-of-n or winner-take-all form, for this learning stage we activate the k closest units and anneal k to 1 throughout learning.

this equation after their treatment of the supervised version gives

$$\frac{\partial E}{\partial w_{1_p}} = \sum_k \sum_{i, i \neq p} \sum_j \frac{\delta(L(w_{1_p}), L(w_{2_j})) - \delta(L(w_{1_i}), L(w_{2_j}))}{\|w_{1_i} - w_{1_p}\|}, \tag{2.3}$$
$$\times P(C_k) \int_{S_{1_{i,p}}} (w_{1_p} - x_1) \int_{V_{2_j}} p(x_1, x_2 | C_k)\, dx_2\, dx_1$$

where $S_{1_{i,p}}$ is the boundary surface between V_{1_i} and V_{1_p}. Similarly,

$$\frac{\partial E}{\partial w_{2_p}} = \sum_k \sum_{i, i \neq p} \sum_j \frac{\delta(L(w_{2_p}), L(w_{1_j})) - \delta(L(w_{2_i}), L(w_{1_j}))}{\|w_{2_i} - w_{2_p}\|}. \tag{2.4}$$
$$\times P(C_k) \int_{S_{2_{i,p}}} (w_{2_p} - x_2) \int_{V_{1_j}} p(x_1, x_2 | C_k)\, dx_1\, dx_2$$

Using uniform Parzen windows as in Wassel and Sklansky (1972), Sklansky and Wassel (1981), and Diamantini and Spalvieri (1995) to approximate the probability distributions in equations 2.3 and 2.4 and considering for each data sample $X_1(n), X_2(n)$, only the two nearest codebook vectors in each modality give a particularly simple stochastic estimate of the derivatives. If $\|X_1 - X_{1_{p,q}}\| \leq \Delta/2$, where $X_{1_{p,q}}$ is the projection of modality sample's point on the border between its closest codebook vectors w_{1_p} and w_{1_q} (and w_{1_p} belongs to the same class as the modality 2 codebook vector closest to X_2, and w_{1_q} belongs to another class), then a sample estimate of the derivative in equation 2.3 for the codebook vector w_{1_p} is

$$-\frac{X_{1_{p,q}} - w_{1_p}}{\Delta(t) \|w_{1_q} - w_{1_p}\|}. \tag{2.5}$$

This gives the following simple stochastic approximation (Robbins & Monro, 1951) update rules for modality 1. If $\|X_1 - X_{1_{p,q}}\| \leq \Delta(t)/2$,

$$w_{1_p}(n+1) = w_{1_p}(n) + \epsilon(t) \frac{X_{1_{p,q}} - w_{1_p}(n)}{\Delta(t) \|w_{1_q}(n) - w_{1_p}(n)\|}. \tag{2.6}$$

$$w_{1_q}(n+1) = w_{1_q}(n) - \epsilon(t) \frac{X_{1_{p,q}} - w_{1_q}(n)}{\Delta(t) \|w_{1_p}(n) - w_{1_q}(n)\|}. \tag{2.7}$$

If the pattern falls outside a window of width $\Delta(t)$ about the current border of the class output by the other modality, no changes are made. For all other indices, no changes in the codebook vector weights are made. The rules for updating the codebook vectors in modality 2 are exactly analogous.

These rules amount to:

> If the pattern received by a modality is close to a current border, move the codebook vector of the class *that is output by the other modality* toward the pattern and that of the other class away.

This rule moves the borders to increase the local area assigned to the class output by the other modality. The minimizing-disagreement (M-D) algorithm applies this rule after each presentation of multimodal stimuli; it can be summarized as follows:

1. Initialize codebook vectors in each modality (unsupervised clustering).
2. Initialize codebook vectors' labels using unsupervised clustering of the activity patterns across the codebook vector units (as described at the end of section 1).
3. Repeat for each presentation of input patterns $X_1(n)$ and $X_2(n)$ to their respective modalities:
    - Find the two nearest codebook vectors in each modality to their respective input patterns.
    - Find the hypothesized output class in each modality (as given by the label of the closest codebook vector).
    - For each modality, update the codebook vectors according to the rule above.
    - Update the labels (described below).

The algorithm is moderately sensitive to the initial labeling, so improved results are often obtained by repeating steps 2 and 3 with the codebook vectors resulting from one cycle through the algorithm. Because the algorithm results in codebook vectors that better distinguish between classes, they tend to be easier to label appropriately in the initial labeling stage, which often leads to better performance after the third stage. The appendix and Figure 7 show more quantitatively how minimizing the disagreement is related to the classification goal of minimizing the number of misclassified patterns.

The mapping of E with respect to the labels ($\{L(w_i)\}$) is not continuous and thus not differentiable. However, to minimize E with respect to the labels (last point of step 3 above), one should assign the label for w_{1_i} to be the label that labels the most co-occurring patterns in the other modality.

If we define the mapping $W_1(L_1)$ to be the set of codebook vectors in modality 1 for which $L(w_{1_i}) = L_1$ and let

$$colabel_l(w_{1_i}) = \sum_{w_{2_j} \in W_2(L_l)} \sum_k P(C_k) \int_{V_{1_i}} \int_{V_{2_j}} p(x_1, x_2|C_k)\, dx_1\, dx_2,$$

then

$$L(w_{1_i}) = \text{argmax}_l(\{colabel_l(w_{1_i})\}).$$

We use an online algorithm for this. Letting $v_{1_{l,i^*}}$ be the weight from codebook vector $w_{1_{i^*}}$ (the winning codebook vector in modality 1) to output unit l (the winning label picked by modality 2),

$$v_{1_{l,i^*}}(n+1) = v_{1_{l,i^*}}(n) + \alpha(n)$$

where the weights coming into each output unit are kept normalized:

$$v_{1_{l,i}} = \frac{v_{1_{l,i}}}{\|v_{1_l}\|} \qquad \forall i, l.$$

This normalization means that the algorithm is not minimizing the disagreement with respect to the output weights but instead clustering the hidden unit representation using the output class given by the other modality. This objective is better for these weights as it balances the goal of agreement with the desire to avoid the trivial solution of all codebook vectors having the same label. Other forms of extra terms to force the output units to output different classes across the pattern set could also be used. This is analogous to the individual entropy terms $H(Y_1)$, $H(Y_2)$ in the IMAX algorithm (Becker & Hinton, 1992), which force the output units Y_i in each modality ($i = 1, 2$) to output space, preventing the trivial solution of both modalities outputting a constant.

We could also modify the energy function that the codebook vectors are following to prevent the hidden units from coding the input space as one class. However, due to the existence of many reasonable local minima, extra terms were not necessary in the data sets we have encountered, and our foray into adding them with our original data set yielded slightly worse performance (probably because the addition of the terms changes the position of the local minimum; it is no longer minimizing the disagreement). However, it is possible that for problems with more overlap between classes, terms like this might help if the current algorithm does not perform well.

Figure 3 illustrates for an easily visualized two-class problem how, despite the existence of the undesirable global minima in disagreement, for enough segregation in the joint space a local minimum exists between the two classes. An initial border determined by most simple clustering algorithms would start within the basin of attraction of this minimum. The figure shows that an appropriate local minimum exists beyond the case where clusters could be separated given the individual modalities alone, but just short of what could be achieved if one could look for clusters in the joint space. The algorithm is able to extract most of the greater structure in the higher-dimensional joint distribution without requiring the extra parameters for modeling in this large space.

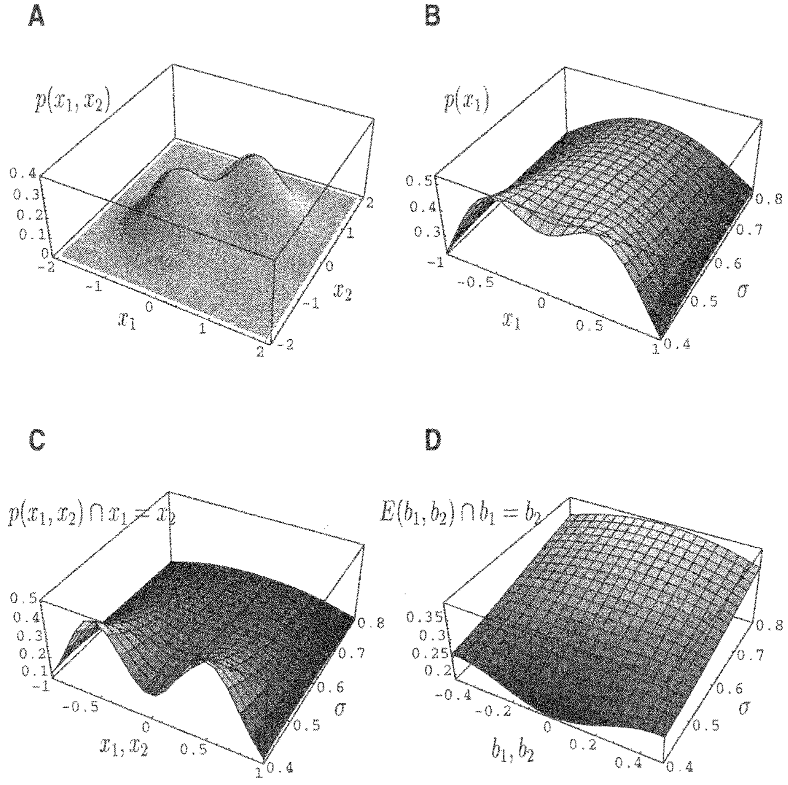

Figure 3: The M-D energy function. (A) The joint probability density for a two-class low dimensional problem. The two modalities are conditionally independent with the individual class distributions in both modalities normally distributed with standard deviation .5 (and means of $-.5, .5$). The dark curve shows the value of the joint distribution for $x_1 = x_2$. For this case, this is the direction that gives the most information on separating the classes. (B) The individual modality distribution for one modality (identical for the other one) for varying standard deviations (σ) of the individual class densities. As the classes become more diffuse (larger σ), the dip between the classes gets smaller, and the classes are harder to separate. (C) The plot of the joint density along the plane $x_1 = x_2$, (the most informative direction) for varying σ. Note that the classes are more separable in the higher-dimensional joint space for a given σ. (D) The minimizing disagreement energy (proportion of misclassified patterns) as a function of border position and σ. Note that for σ up to almost .6, the correct dividing borders ($b_1 = b_2 = 0$) are a local minimum (along $b_1 = b_2$ is the limiting direction); however, for more diffuse classes, there is no appropriate local minimum—only the global minima as $x_1, x_2 \rightarrow \pm\infty$.

3 Network Realization

The M-D algorithm can be realized by the three-layer network shown in Figure 1, where the codebook vectors are represented by the weights $\boldsymbol{w}$ from the input to the hidden units (also called codebook vector units) and the class labels are represented by the output units. The codebook labels are given implicitly by the weights $\boldsymbol{v}$ and $\boldsymbol{u}$. The codebook vector units determine which class is chosen (their label) through the forward weights $\boldsymbol{v}$ and receive agreement information through the complementary back-projecting weights $\boldsymbol{u}$.

Simple feedforward clustering using competitive learning is used to initialize the codebook vectors $\boldsymbol{w}$ in their respective input pattern spaces and subsequently the weights to the output units $\boldsymbol{v}$. During this stage, the back-projecting weights $\boldsymbol{u}$ are kept consistent with the forward weights $\boldsymbol{v}$ by setting the back-projecting weights to the active hidden unit to Z—the 1-of-n activity vector over the output units driven by the forward weights from that activated unit. This results in backward weights of magnitude 1 from the output unit activated by the hidden unit and 0 from the others, and can be considered a form of fast Hebbian learning (Hebb, 1949).

Next, using the M-D rule, both modalities teach each other. For each paired pattern presentation, the output units are driven first by one modality and then by the other. The output units are driven by the forward-projecting weights $\boldsymbol{v}$ of the current driving modality. This output then augments the activity in the nondriving modality through the back-projecting weights $\boldsymbol{u}$, which provide boosted activity to activated units that agree with the output of the driving modality. (For details, see the caption of Figure 1.) Figure 1 shows the case where modality 1 teaches or drives modality 2. The codebook vectors in the driven modality ($\boldsymbol{w_2}$ in the figure) are updated using a simplified version of the M-D rule. (This rule, derived from generalizing the one-dimensional rule, is very slightly different from the rules in equations 2.6 and 2.7 derived from differentiating in the multidimensional space but is simpler and has performed as well or better.) Weights are updated only if the current pattern falls near the middle between two codebook vectors of different classes (The specification of this "window" and the decrease in learning step size are as in Kohonen's (1990) supervised algorithm. The window is decreased with time, as in de Sa & Ballard, 1993a.) In this case:

$$w_{2_i}(n+1) = \begin{cases} w_{2_i}(n) & \text{if } Y_{2_i} < A \\ w_{2_i}(n) - \epsilon(n)\frac{(X_2(n)-w_{2_i}(n-1))}{\|X_2(n)-w_{2_i}(n)\|} & \text{if } A \leq Y_{2_i} < B \\ w_{2_i}(n) & \text{if } B \leq Y_{2_i} < C \\ w_{2_i}(n) + \epsilon(n)\frac{(X_2(n)-w_{2_i}(n-1))}{\|X_2(n)-w_{2_i}(n)\|} & \text{if } Y_{2_i} \geq C. \end{cases} \tag{3.1}$$

This rule for updating the codebook vectors is a discrete version of the ABS rule (Artola & Singer, 1993) shown in graphical form in the inset in Figure 1. (In Figure 1, $A = .4$, $B = 1.4$, $C = 1.7$.) The nonpropagating forward

($\boldsymbol{v_2}$) and backward weights ($\boldsymbol{u_1}$) are also updated at this stage. The backward weights of the winning driving unit ($u_{1^*_i}$) are kept consistent with the forward weights ($\boldsymbol{v_1}$), as in the initialization stage, and the forward weights of the driven modality ($\boldsymbol{v_2}$) are updated to decrease the disagreement error by moving toward the output vector (Z) output by the other (driving) modality (already described). The forward weights to each output unit are kept normalized.

4 Results

The algorithm was demonstrated on the problem of learning to recognize consonant-vowel utterances both visually and acoustically.

A speaker was recorded using an 8 mm camcorder and directional microphone as he spoke 118 repetitions of /ba/, /va/, /da/, /ga/, and /wa/. The first 98 samples of each utterance class formed the training set and the remaining 20 the test set. Each set of 10 utterances (twice through the set) was preceded by a clap using a clapboard arrangement similar to that used in commercial movie production for matching the visual and auditory signals. The camera recorded 30 frames a second and was positioned to view the tip of the nose through the chin of the speaker.

The acoustic data were low-pass filtered, and utterances were detected using threshold crossings of the smoothed time-domain waveform (using the ESPS (Entropic Signal Processing System) from Entropic Research Laboratory). As some of the consonantal information is low amplitude (before the threshold crossing), each utterance was taken from 50 msec before the automatically detected utterance start to 50 msec after. These utterances were then encoded using a 24-channel mel code[2] over 20 msec windows overlapped by 10 msec. This is a coarse, short time frequency encoding, which crudely approximates peripheral auditory processing. Each feature vector was linearly scaled so that all dimensions lie in the range $[-1, 1]$. The final auditory code is a (24×9) 216-dimension vector for each utterance. Example auditory feature vectors are shown in Figure 4.

The visual data were processed using software designed and written by Ramprasad Polana (Polana, 1994). Visual frames were digitized as 64×64 8-bit gray-level images using the Datacube MaxVideo system. The video and auditory tracks were aligned using the clapboard arrangement, and visual detection of the clap was performed manually, which allowed alignment to within one video frame (1/30 second). The frame of the clap was matched to the time of the acoustically detected clap, allowing the automatic segmentation obtained from the acoustic signal to be used to segment the video. Segments were taken as six frames before the acoustically determined utterance offset and four after. The normal flow was computed using

[2] Linear spacing below 1000 Hz and logarithmic above 1000 Hz.

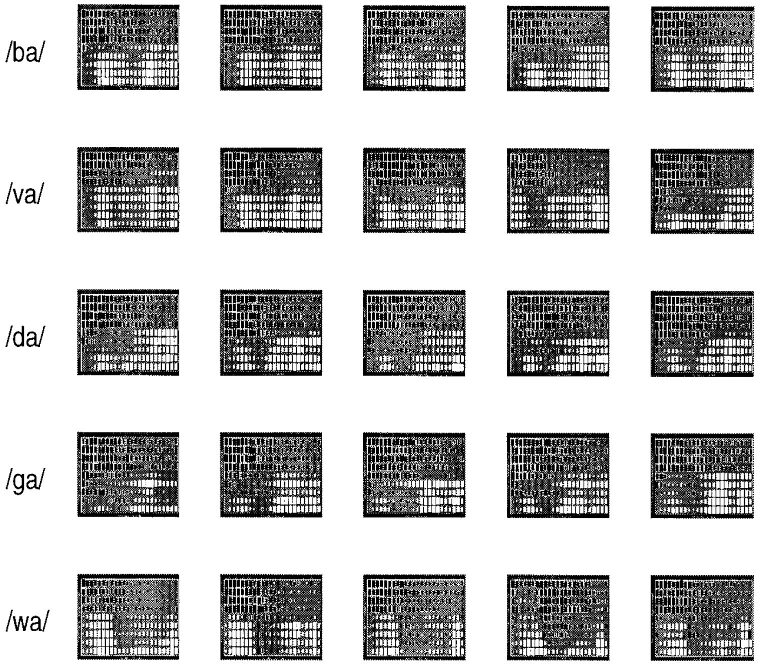

Figure 4: Example auditory patterns. The x-axis within each feature vector represents frequency channels and the y-axis time. The area of the small squares within the feature vector corresponds to the magnitude and the color that of the sign (white positive, black negative) of the feature dimension.

differential techniques between successive frames. Each pair of frames was then averaged, resulting in five frames of motion over the 64 × 64 pixel grid. The frames were divided into 25 equal areas (5 × 5), and the motion magnitudes within each frame were averaged within each area. The final visual feature vector of dimension (5 frames × 25 areas) 125 was linearly normalized as for the auditory vectors. Example visual feature vectors are shown in Figure 5.

The results are shown in Figure 6. After training, the visual network achieved a classification performance of 80% on the test set, while the auditory network had a test set performance of 93%. For comparison, the LVQ2.1 algorithm trained on the auditory data with the same architecture as the auditory subnetwork had a test set classification of 96%, and the supervised visual network, again with the same architecture as the corresponding subnet, 83%. The performance after the initial unsupervised clustering was 56%

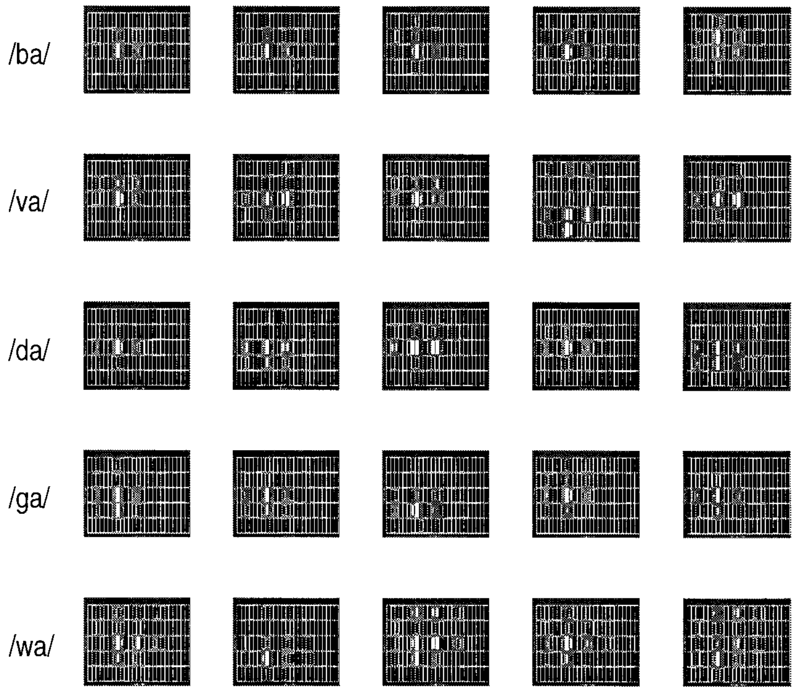

Figure 5: Example visual patterns. These patterns correspond to the auditory patterns in Figure 4. The x-axis represents spatial positions and the y-axis time. The area of the small squares within the feature vector corresponds to the magnitude and the color of the sign (white positive, black negative) of the feature dimension.

and 66% for the auditory and visual subnets, respectively, even though we helped this stage by weighting the auditory activity pattern 50% more than the visual pattern. Slightly better results are obtained with artificially increased pairing. In these experiments, the data set was expanded by matching each auditory pattern of one class of utterances with each visual pattern of that class in the training set (not just the individual pattern with which it co-occurred). These results reflect the expected performance of the M-D algorithm with more data under the assumption that within an utterance class, the exact auditory and visual patterns are independent (and thus each auditory pattern is just as likely to have occurred with each visual pattern in the class).

Results on a preliminary multispeaker task were not as favorable, likely due to the greatly increased difficulty in the visual classification problem

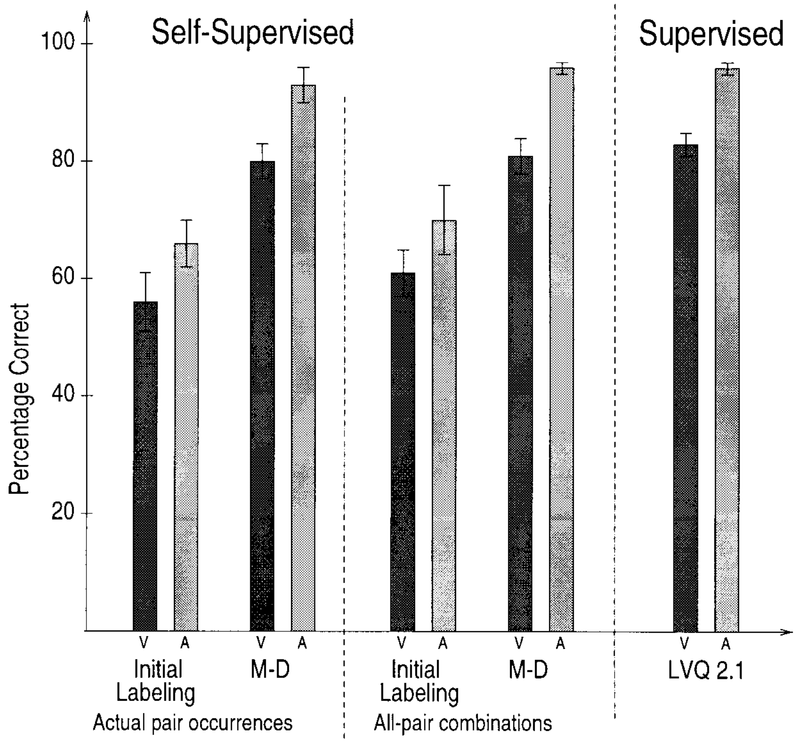

Figure 6: Experimental results in the auditory-visual speech task. The left bar in each set (labeled V) gives the performance of the visual network, and the right bars (labeled A) show the auditory network's performance. All bars represent categorization performance on the test set averaged over 30 experiments from random initial weights. The error bars represent one standard deviation across the runs. The pair of bars labeled "Initial Labeling" represent the performance after the initialization stage of unsupervised clustering in the respective input spaces and output label space. This gives the categorization ability of unsupervised clustering. The bars labeled "All-Pair Combinations" represent the use of an artificially increased data set obtained by matching each auditory pattern of one class of utterances with each visual pattern of that class in the training set (not just the individual pattern with which it co-occurred). Results for the related supervised algorithm (LVQ2.1) (Kohonen, 1990) using the same number of codebook vectors are also shown for comparison.

(supervised test performance on the visual data was about 60%). It is likely that a different visual encoding could improve this result, or possibly more data were required because the artificially increased pairing did give results comparable to supervised performance. (However, the argument that this simulates collection of more data is less compelling in the multispeaker case.)

5 Discussion

The fact that both networks are simultaneously learning makes this problem significantly harder than approaches where one modality trains another (Munro, 1988; Carpenter, Grossberg, & Reynolds, 1991; Tan, 1995) or others that combine two already trained networks (Yuhas, Goldstein, & Sejnowski, 1988; Stork, Wolff, & Levine, 1992). The approach taken in this work and that of Becker and Hinton (1992), Becker (1996), Schmidhuber and Prelinger (1993), Phillips et al. (1995), and Kay et al. (1998) is to use the relationships between inputs to different networks to discover features in the individual networks that could not have been discovered simply by unsupervised learning in the individual spaces. This algorithm is more restricted than these other similar algorithms because it is limited to classification problems and uses 1-of-*n* or winner-take-all output encodings. On the other hand, it easily deals with real problems of many input dimensions. To the best of our knowledge, this is the largest problem attempted with this type of algorithm. It also deals easily, and has improved performance, with the addition of more modalities (for an example of a similar precursor algorithm, see de Sa & Ballard, 1993b).

One limit of the 1-of-*n* output encoding is that as the number of output units is fixed, the number of output classes must be pre-chosen. In our case, five output units were used because we were looking for five classes. For fewer output classes, the algorithm can simply group classes, and we would expect no change or even a decrease in the number of disagreements. For more output classes, we would expect an increase in the number of disagreements. We have done experiments with two, three, four, six, seven, and ten output classes. On the training data, these experiments show a greater increase in the number of disagreements after five classes, though the same curve on the test set gave only a smooth increase in disagreements with class size. Thus, it is possible that the number of classes might be recoverable from the data. However, this will require further development.

The algorithm is currently limited to hard winner-take-all decisions. Incorporating “soft” output decisions would be an easy modification, but making appropriate use of the resulting extra information to provide better teaching signals to the other modality is an interesting problem deserving more research. The uncertainty in classification in the driven hidden layer is reflected in the windowing; only patterns near a current border are able to influence the border. This decision can be made softer by using nonuniform Parzen windows.

The M-D algorithm offers a straightforward computational model for why neurons in one sensory modality also respond to inputs to another sensory modality (Morrell, 1972; Fishman & Michael, 1973; Sams et al., 1991; Buser & Borenstein, 1959; Murata, Cramer, & Rita, 1965; Spinelli, Starr, & Barrett, 1968; Haenny, Maunsell, & Schiller, 1989; Maunsell, Sclar, Nealey, & DePriest, 1991). In fact, the algorithm is not limited to different sensory modalities but can also be used for submodal patterns such as color and motion. The key requirement is that there be some decorrelation of the instances of the different pairs of patterns. The model shows that without the huge cost of connecting all neurons to all sensory input, one can still take advantage of the greater structure available in the higher-dimensional multimodal sensory space. We suggest that cortical back-projections and multisensory integration may be doing more than affecting the properties of developed systems but may play an important role in the learning process itself.

Appendix: Minimizing Disagreement as an Approximation to Minimizing Misclassifications

Note that the algorithm to minimize the disagreement corresponds to the LVQ2.1 algorithm except that the "label" for each modality's pattern is the hypothesized output of the other modality. To understand how making use of this label, through minimizing the disagreement between the two outputs, relates to the true goal of minimizing misclassifications in each modality, consider the conditionally independent (within a class) version of the two-modality example illustrated in Figure 7. In the supervised case (see Figure 7A), the availability of the actual labels allows sampling of the actual marginal distributions. For each modality, the number of misclassifications can be minimized by setting the boundaries for each modality at the crossing points of their marginal distributions.

However, in the self-supervised system, the labels are not available. Instead we are given the output of the other modality. Consider the system from the point of view of modality 2. Its patterns are labeled according to the outputs of modality 1. This labels the patterns in class A as shown in Figure 7B. Thus, from the actual class A patterns, the second modality sees the "labeled" distributions shown. Letting a be the fraction of class A patterns that are misclassified by modality 1, the resulting distributions of the real class A patterns seen by modality 2 are $(1 - a)P(C_A)p(x_2|C_A)$ and $(a)P(C_A)p(x_2|C_A)$.

Similarly, Figure 7C shows modality 2's view of the patterns from class B (given modality 1's current border). Letting b be the fraction of class B patterns misclassified by modality 1, the distributions are given by $(1 - b)P(C_B)p(x_2|C_B)$ and $(b)P(C_B)p(x_2|C_B)$. Combining the effects on both classes

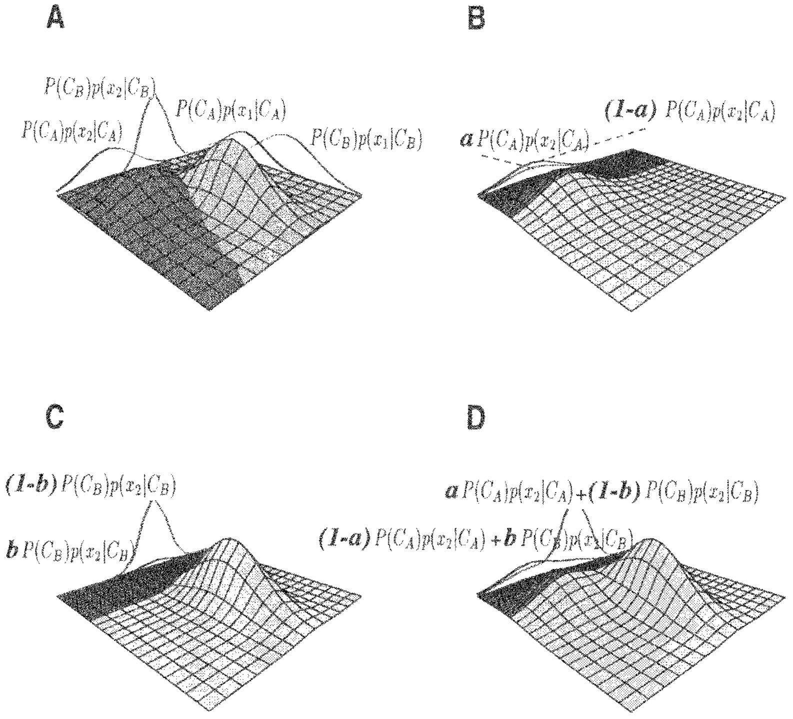

Figure 7: An example of the joint and marginal distributions for the conditionally independent version of the example problem introduced in Figure 2. The darker gray represents patterns labeled A; the lighter gray are labeled B. (A) The labeling for the supervised case. (B) The labeling of class A patterns as seen by modality 2 given the modality 1 border shown. ***a*** represents the fraction of the class A patterns that are misclassified by modality 1. (C) The labeling of class B patterns as seen by modality 2 given the same modality 1 border. ***b*** represents the fraction of the class B patterns that are misclassified by modality 1. (D) The total pattern distributions seen by modality 2 given the labels determined by modality 1. These distributions can be considered as the labeled distributions on which modality 2 is performing a form of supervised learning. (However, it is more complicated; modality 1's border is concurrently influenced by the current position of modality 2's border.) See the text for details.

results in the "labeled" distributions shown in Figure 7D. The "apparent class A" distribution is given by $(1-a)P(C_A)p(x_2|C_A) + (b)P(C_B)p(x_2|C_B)$ and the "apparent class B" distribution by $(a)P(C_A)p(x_2|C_A) + (1-b)P(C_B)p(x_2|C_B)$. The crossing point of these two distributions occurs at the value of x_2 for which $(1-2a)P(C_A)p(x_2|C_A) = (1-2b)P(C_B)p(x_2|C_B)$. Comparing this with the crossing point of the actual distributions that occurs at x_2 satisfying $P(C_A)p(x_2|C_A) = P(C_B)p(x_2|C_B)$ reveals that if the proportion of class A patterns misclassified by modality 1 is the same as the proportion of class B patterns misclassified by modality 1 (i.e., $a = b$), the crossing points of the distributions will be identical. This is true even though the approximated distributions will be discrepant for all cases where there are any misclassified patterns ($a > 0$ OR $b > 0$). If $a \approx b$, the crossing point will be close.

Simultaneously the second modality is labeling the patterns to the first modality. At each iteration of the algorithm, both borders move according to the samples from the "apparent" marginal distributions.

Acknowledgments

This research was supported by grants from the Human Frontier Science Program, the National Institutes of Health Biotechnology Resource Program, and a Canadian NSERC 1967 Science and Engineering Scholarship. We thank Sue Becker, Peter Dayan, Jeff Schneider, Zoubin Ghahramani, Mary Hayhoe, Chris Brown, David Williams, and the anonymous reviewers for helpful comments and Ramprasad Polana and Ramesh Sarukkai for help with the visual and auditory pattern processing.

References

Artola, A., & Singer, W. (1993). Long-term depression of excitatory synaptic transmission and its relationship to long-term potentiation. *Trends in Neurosciences*, *16*(11), 480–487.

Becker, S. (1996). Mutual information maximization: Models of cortical self-organization. *Network: Computation in Neural Systems*, *7*, 7–31.

Becker, S., & Hinton, G. E. (1992). A self-organizing neural network that discovers surfaces in random-dot stereograms. *Nature*, *355*, 161–163.

Buser, P., & Borenstein, P. (1959). Responses somesthesiques, visuel et auditives, recuellies, au niveau du cortex "associatif" infrasylvien chez le chat curarise non anesthesie. *Electroencephalog. Clin. Neurophysiol.*, *11*, 285–304.

Carpenter, G. A., Grossberg, S., & Reynolds, J. H. (1991). Artmap: Supervised real-time learning and classification of nonstationary data by a self-organizing neural network. *Neural Networks*, *4*, 565–588.

Coultrip, R., Granger, R., & Lynch, G. (1992). A cortical model of winner-take-all competition via lateral inhibition. *Neural Networks*, *5*, 47–54.

de Sa, V. R., & Ballard, D. H. (1993a). A note on learning vector quantization. In C. Giles, S. J. Hanson, & J. Cowan (Eds.), *Advances in neural information processing systems 5* (pp. 220–227). San Mateo, CA: Morgan Kaufmann.

de Sa, V. R., & Ballard, D. H. (1993b). Self-teaching through correlated input. In F. H. Eeckman & J. M. Bower (Eds.), *Computation and neural systems 1992* (pp. 437–441). Needham, MA: Kluwer Academic.

Diamantini, C., & Spalvieri, A. (1995). Pattern classification by the Bayes machine. *Electronics Letters*, *31*(24), 2086–2088.

Durgin, F. H., & Proffitt, D. R. (1996). Visual learning in the perception of texture: Simple and contingent aftereffects of texture density. *Spatial Vision*, *9*(4), 423–474.

Fishman, M. C., & Michael, C. R. (1973). Integration of auditory information in the cat's visual cortex. *Vision Research*, *13*, 1415–1419.

Grossberg, S. (1976a). Adaptive pattern classification and universal recoding: I. Parallel development and coding of neural feature detectors. *Biological Cybernetics*, *23*, 121–134.

Grossberg, S. (1976b). Adaptive pattern classification and universal recoding: II. Feedback, expectation, olfaction, illusions. *Biological Cybernetics*, *23*, 187–202.

Haenny, P., Maunsell, J., & Schiller, P. (1989). State dependent activity in monkey visual cortex II. Retinal and extraretinal factors in V4. *Experimental Brain Research*, *69*(2), 245–259.

Hebb, D. O. (1949). *The organization of behavior*. New York: Wiley.

Howells, T. (1944). The experimental development of color-tone synesthesia. *Journal of Experimental Psychology*, *34*(2), 87–103.

Kay, J., Floreano, D., & Phillips, W. (1998). Contextually guided unsupervised learning using local multivariate binary processors. *Neural Networks*, *11*(1), 117–140.

Kohonen, T. (1982). Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, *43*, 59–69.

Kohonen, T. (1990). Improved versions of learning vector quantization. In *IJCNN International Joint Conference on Neural Networks* (Vol. 1, pp. I545–I550).

MacDonald, J., & McGurk, H. (1978). Visual influences on speech perception processes. *Perception and Psychophysics*, *24*(3), 253–257.

Maunsell, J., Sclar, G., Nealey, T., & DePriest, D. (1991). Extraretinal representations in area V4 of Macaque monkey. *Visual Neuroscience*, *7*(6), 561–573.

McGurk, H., & MacDonald, J. (1976). Hearing lips and seeing voices. *Nature*, *264*, 746–748.

Miikkulainen, R. (1991). Self-organizing process based on lateral inhibition and synaptic resource redistribution. In T. Kohonen, K. Makisära, O. Simula, & J. Kangas (Eds.), *Artificial neural networks* (pp. 415–420). Amsterdam: Elsevier.

Morrell, F. (1972). Visual system's view of acoustic space. *Nature*, *238*, 44–46.

Munro, P. (1988). *Self-supervised learning of concepts by single units and "weakly local" representations* (Tech. Rep. No. LIS003/IS88003). Pittsburgh, PA: School of Library and Information Science, University of Pittsburgh.

Murata, K., Cramer, H., & Rita, P. B. (1965). Neuronal convergence of noxious, acoustic and visual stimuli in the visual cortex of the cat. *Journal of Neurophysiology*, *28*, 1233–1239.

Phillips, W., Kay, J., & Smyth, D. (1995). The discovery of structure by multistream networks of local processors with contextual guidance. *Network: Computation in Neural Systems*, *6*, 225–246.

Polana, R. (1994). *Temporal texture and activity recognition*. Unpublished doctoral dissertation, Department of Computer Science, University of Rochester.

Robbins, H., & Monro, S. (1951). A stochastic approximation method. *Annals of Math. Stat.*, *22*, 400–407.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning internal representations by error propagation. In D. E. Rumelhart, J. L. McClelland, & the PDP Research Group (Eds.), *Parallel distributed processing: Explorations in the microstructure of cognition* (Vol. 1, pp. 318–364). Cambridge, MA: MIT Press.

Rumelhart, D. E., & Zipser, D. (1986). Feature discovery by competitive learning. In D. E. Rumelhart, J. L. McClelland, & the PDP Research Group (Eds.), *Parallel distributed processing: Explorations in the microstructure of cognition* (Vol. 1, pp. 151–193). Cambridge, MA: MIT Press.

Sams, M., Aulanko, R., Hämääinen, M., Hari, R., Lounasmaa, O. V., Lu, S.-T., & Simola, J. (1991). Seeing speech: Visual information from lip movements modifies activity in the human auditory cortex. *Neuroscience Letters*, *127*, 141–145.

Schmidhuber, J., & Prelinger, D. (1993). Discovering predictable classifications. *Neural Computation*, *5*, 625–635.

Sklansky, J., & Wassel, G. N. (1981). *Pattern classifiers and trainable machines*. Berlin: Springer-Verlag.

Spinelli, D., Starr, A., & Barrett, T. W. (1968). Auditory specificity in unit recordings from cat's visual cortex. *Experimental Neurology*, *22*, 75–84.

Stork, D. G., Wolff, G., & Levine, E. (1992). Neural network lipreading system for improved speech recognition. In *IJCNN International Joint Conference on Neural Networks* (Vol. 2, pp. II286–II295).

Tan, A.-H. (1995). Adaptive resonance associative map. *Neural Networks*, *8*(3), 437–446.

Wassel, G. N., & Sklansky, J. (1972). Training a one-dimensional classifier to minimize the probability of error. *IEEE Transactions on Systems, Man, and Cybernetics*, *SMC-2*(4), 533–541.

Yuhas, B., Goldstein, M. W., Jr., & Sejnowski, T. J. (1988). Neural network models of sensory integration for improved vowel recognition. *Proceedings of the IEEE*, *78*(10), 1658–1668.

Zellner, D. A., & Kautz, M. A. (1990). Color affects perceived odor intensity. *Journal of Experimental Psychology: Human Perception and Performance*, *16*(2), 391–397.

Received January 8, 1997; accepted October 8, 1997.