Communicated by Garrison Cottrell

Dimension Reduction by Local Principal Component Analysis

Nandakishore Kambhatla
Todd K. Leen
Department of Computer Science and Engineering, Oregon Graduate Institute of Science and Technology, Portland, Oregon 97291-1000, U.S.A.

Reducing or eliminating statistical redundancy between the components of high-dimensional vector data enables a lower-dimensional representation without significant loss of information. Recognizing the limitations of principal component analysis (PCA), researchers in the statistics and neural network communities have developed nonlinear extensions of PCA. This article develops a local linear approach to dimension reduction that provides accurate representations and is fast to compute. We exercise the algorithms on speech and image data, and compare performance with PCA and with neural network implementations of nonlinear PCA. We find that both nonlinear techniques can provide more accurate representations than PCA and show that the local linear techniques outperform neural network implementations.

1 Introduction

The objective of dimension reduction algorithms is to obtain a parsimonious description of multivariate data. The goal is to obtain a compact, accurate, representation of the data that reduces or eliminates statistically redundant components.

Dimension reduction is central to an array of data processing goals. Input selection for classification and regression problems is a task-specific form of dimension reduction. Visualization of high-dimensional data requires mapping to a lower dimension—usually three or fewer. Transform coding (Gersho & Gray, 1992) typically involves dimension reduction. The initial high-dimensional signal (e.g., image blocks) is first transformed in order to reduce statistical dependence, and hence redundancy, between the components. The transformed components are then scalar quantized. Dimension reduction can be imposed explicitly, by eliminating a subset of the transformed components. Alternatively, the allocation of quantization bits among the transformed components (e.g., in increasing measure according to their variance) can result in eliminating the low-variance components by assigning them zero bits.

Recently several authors have used neural network implementations of dimension reduction to signal equipment failures by novelty, or outlier,

detection (Petsche et al., 1996; Japkowicz, Myers, & Gluck, 1995). In these schemes, high-dimensional sensor signals are projected onto a subspace that best describes the signals obtained during normal operation of the monitored system. New signals are categorized as normal or abnormal according to the distance between the signal and its projection[1].

The classic technique for linear dimension reduction is principal component analysis (PCA). In PCA, one performs an orthogonal transformation to the basis of correlation eigenvectors and projects onto the subspace spanned by those eigenvectors corresponding to the largest eigenvalues. This transformation decorrelates the signal components, and the projection along the high-variance directions maximizes variance and minimizes the average squared residual between the original signal and its dimension-reduced approximation.

A neural network implementation of one-dimensional PCA implemented by Hebb learning was introduced by Oja (1982) and expanded to hierarchical, multidimensional PCA by Sanger (1989), Kung and Diemantaras (1990), and Rubner and Tavan (1989). A fully parallel (nonhierarchical) design that extracts orthogonal vectors spanning an m-dimensional PCA subspace was given by Oja (1989). Concurrently, Baldi and Hornik (1989) showed that the error surface for linear, three-layer autoassociators with hidden layers of width m has global minima corresponding to input weights that span the m-dimensional PCA subspace.

Despite its widespread use, the PCA transformation is crippled by its reliance on second-order statistics. Though uncorrelated, the principal components can be highly statistically dependent. When this is the case, PCA fails to find the most compact description of the data. Geometrically, PCA models the data as a hyperplane embedded in the ambient space. If the data components have nonlinear dependencies, PCA will require a larger-dimensional representation than would be found by a nonlinear technique. This simple realization has prompted the development of nonlinear alternatives to PCA.

Hastie (1984) and Hastie and Stuetzle (1989) introduce their principal curves as a nonlinear alternative to one-dimensional PCA. Their parameterized curves $f(\lambda)$: $R \rightarrow R^n$ are constructed to satisfy a self-consistency requirement. Each data point x is projected to the closest point on the curve $\lambda_f(x) = \text{argmin}_\mu \ \|x - f(\mu)\|$, and the expectation of all data points that project to the same parameter value λ is required to be on the curve. Thus $f(\Lambda) = E_x[x \mid \lambda_f(x) = \Lambda]$. This mathematical statement reflects the desire that the principal curves pass through the middle of the data.

[1] These schemes also use a sigmoidal contraction map following the projection so that new signals that are close to the subspace, yet far from the training data used to construct the subspace, can be properly tagged as outliers.

Hastie and Stuetzle (1989) prove that the curve $f(\lambda)$ is a principal curve iff it is a critical point (with respect to variations in $f(\lambda)$) of the mean squared distance between the data and their projection onto the curve. They also show that if the principal curves are lines, they correspond to PCA. Finally, they generalize their definitions from principal curves to principal surfaces.

Neural network approximators for principal surfaces are realized by five-layer, autoassociative networks. Independent of Hastie and Stuetzle's work, several researchers (Kramer, 1991; Oja, 1991; DeMers & Cottrell, 1993; Usui, Nakauchi, & Nakano, 1991) have suggested such networks for nonlinear dimension reduction. These networks have linear first (input) and fifth (output) layers, and sigmoidal nonlinearities on the second- and fourth-layer nodes. The input and output layers have width n. The third layer, which carries the dimension-reduced representation, has width $m < n$. We will refer to this layer as the *representation layer*. Researchers have used both linear and sigmoidal response functions for the representation layer. Here we consider only linear response in the representation layer.

The networks are trained to minimize the mean squared distance between the input and output and, because of the middle-layer bottleneck, build a dimension-reduced representation of the data. In view of Hastie and Stuetzle's critical point theorem, and the mean square error (MSE) training criteria for five-layer nets, these networks can be viewed as approximators of principal surfaces.[2]

Recently Hecht-Nielsen (1995) extended the application of five-layer autoassociators from dimension reduction to encoding. The third layer of his replicator networks has a staircase nonlinearity. In the limit of infinitely steep steps, one obtains a quantization of the middle-layer activity, and hence a discrete encoding of the input signal.[3]

In this article, we propose a locally linear approach to nonlinear dimension reduction that is much faster to train than five-layer autoassociators and, in our experience, provides superior solutions. Like five-layer autoassociators, the algorithm attempts to minimize the MSE between the original

[2] There is, as several researchers have pointed out, a fundamental difference between the representation constructed by Hastie's principal surfaces and the representation constructed by five-layer autoassociators. Specifically, autoassociators provide a continuous parameterization of the embedded manifold, whereas the principal surfaces algorithm does not constrain the parameterization to be continuous.

[3] Perfectly sharp staircase hidden-layer activations are, of course, not trainable by gradient methods, and the plateaus of a rounded staircase will diminish the gradient signal available for training. However, with parameterized hidden unit activations $h(x; a)$ with $h(x; 0) = x$ and $h(x; 1)$ a sigmoid staircase, one can envision starting training with linear activations and gradually shift toward a sharp (but finite sloped) staircase, thereby obtaining an approximation to Hecht-Nielsen's replicator networks. The changing activation function will induce both smooth changes and bifurcations in the set of cost function minima. Practical and theoretical issues in such homotopy methods are discussed in Yang and Yu (1993) and Coetzee and Stonick (1995).

data and its reconstruction from a low-dimensional representation—what we refer to as the *reconstruction error*. However, while five-layer networks attempt to find a global, smooth manifold that lies close to the data, our algorithm finds a set of hyperplanes that lie close to the data. In a loose sense, these hyperplanes locally approximate the manifold found by five-layer networks.

Our algorithm first partitions the data space into disjoint regions by vector quantization (VQ) and then performs local PCA about each cluster center. For brevity, we refer to the hybrid algorithms as VQPCA. We introduce a novel VQ distortion function that optimizes the clustering for the reconstruction error. After training, dimension reduction proceeds by first assigning a datum to a cluster and then projecting onto the m-dimensional principal subspace belonging to that cluster. The encoding thus consists of the index of the cluster and the local, dimension-reduced coordinates within the cluster.

The resulting algorithm directly minimizes (up to local optima) the reconstruction error. In this sense, it is optimal for dimension reduction. The computation of the partition is a generalized Lloyd algorithm (Gray, 1984) for a vector quantizer that uses the reconstruction error as the VQ distortion function. The Lloyd algorithm iteratively computes the partition based on two criteria that ensure minimum average distortion: (1) VQ centers lie at the generalized centroids of the quantizer regions and (2) the VQ region boundaries are surfaces that are equidistant (in terms of the distortion function) from adjacent VQ centers. The application of these criteria is considered in detail in section 3.2. The primary point is that constructing the partition according to these criteria provides a (local) minimum for the reconstruction error.

Training the local linear algorithm is far faster than training five-layer autoassociators. The clustering operation is the computational bottleneck, though it can be accelerated using the usual tree-structured or multistage VQ. When encoding new data, the clustering operation is again the main consumer of computation, and the local linear algorithm is somewhat slower for encoding than five-layer autoassociators. However, decoding is much faster than for the autoassociator and is comparable to PCA.

Local PCA has been previously used for exploratory data analysis (Fukunaga & Olsen, 1971) and to identify the intrinsic dimension of chaotic signals (Broomhead, 1991; Hediger, Passamante, & Farrell, 1990). Bregler and Omohundro (1995) use local PCA to find image features and interpolate image sequences. Hinton, Revow, and Dayan (1995) use an algorithm based on local PCA for handwritten character recognition. Independent of our work, Dony and Haykin (1995) explored a local linear approach to transform coding, though the distortion function used in their clustering is not optimized for the projection, as it is here. Finally, local linear methods for regression have been explored quite thoroughly. See, for example, the LOESS algorithm in Cleveland and Devlin (1988).

In the remainder of the article, we review the structure and function of autoassociators, introduce the local linear algorithm, and present experimental results applying the algorithms to speech and image data.

2 Dimension Reduction and Autoassociators

As considered here, dimension reduction algorithms consist of a pair of maps g: $R^n \to R^m$ with $m < n$ and f: $R^m \to R^n$. The function $g(x)$ maps the original n-dimensional vector x to the dimension-reduced vector $y \in R^m$. The function $f(y)$ maps back to the high-dimensional space. The two maps, g and f, correspond to the encoding and decoding operation, respectively. If these maps are smooth on open sets, then they are diffeomorphic to a canonical projection and immersion, respectively.

In general, the projection loses some of the information in the original representation, and $f(g(x)) \neq x$. The quality of the algorithm, and adequacy of the chosen target dimension m, are measured by the algorithm's ability to reconstruct the original data. A convenient measure, and the one we employ here, is the average squared residual, or reconstruction error:

$$\mathcal{E} = E_x[\, \|x - f(g(x))\|^2 \,].$$

The variance in the original vectors provides a useful normalization scale for the squared residuals, so our experimental results are quoted as normalized reconstruction error:

$$\mathcal{E}_{\text{norm}} = \frac{E_x[\, \|x - f(g(x))\|^2 \,]}{E_x[\, \|x - E_x x\|^2 \,]}. \tag{2.1}$$

This may be regarded as the noise-to-signal ratio for the dimension reduction, with the signal strength defined as its variance.

Neural network implementations of these maps are provided by autoassociators, layered, feedforward networks with equal input and output dimension n. During training, the output targets are set to the inputs; thus, autoassociators are sometimes called self-supervised networks. When used to perform dimension reduction, the networks have a hidden layer of width $m < n$. This hidden, or representation, layer is where the low-dimensional representation is extracted. In terms of the maps defined above, processing from the input to the representation layer corresponds to the projection g, and processing from the representation to the output layer corresponds to the immersion f.

2.1 Three-Layer Autoassociators. Three-layer autoassociators with n input and output units and $m < n$ hidden units, trained to perform the identity transformation over the input data, were used for dimension reduction by several researchers (Cottrell, Munro, & Zipser, 1987; Cottrell &

Metcalfe, 1991; Golomb, Lawrence, & Sejnowski, 1991). In these networks, the first layer of weights performs the projection g and the second the immersion f. The output $f(g(x))$ is trained to match the input x in the mean square sense.

Since the transformation from the hidden to the output layer is linear, the network outputs lie in an m-dimensional linear subspace of R^n. The hidden unit activations define coordinates on this hyperplane. If the hidden unit activation functions are linear, it is obvious that the best one can do to minimize the reconstruction error is to have the hyperplane correspond to the m-dimensional principal subspace. Even if the hidden-layer activation functions are nonlinear, the output is still an embedded hyperplane. The nonlinearities introduce nonlinear scaling of the coordinate intervals on the hyperplane. Again, the solution that minimizes the reconstruction error is the m-dimensional principal subspace.

These observations were formalized in two early articles. Baldi and Hornik (1989) show that optimally trained three-layer linear autoassociators perform an orthogonal projection of the data onto the m-dimensional principal subspace. Bourlard and Kamp (1988) show that adding nonlinearities in the hidden layer cannot reduce the reconstruction error. The optimal solution remains the PCA projection.

2.2 Five-Layer Autoassociators. To overcome the PCA limitation of three-layer autoassociators and provide the capability for genuine nonlinear dimension reduction, several authors have proposed five-layer autoassociators (e.g., Oja, 1991; Usui et al., 1991; Kramer, 1991; DeMers & Cottrell, 1993; Kambhatla & Leen, 1994; Hecht-Nielsen, 1995). These networks have sigmoidal second and fourth layers and linear first and fifth layers. The third (representation) layer may have linear or sigmoidal response. Here we use linear response functions. The first two layers of weights carry out a nonlinear projection g: $R^n \rightarrow R^m$, and the last two layers of weights carry out a nonlinear immersion f: $R^m \rightarrow R^n$ (see Figure 1). By the universal approximation theorem for single hidden-layer sigmoidal nets (Funahashi, 1989; Hornik, Stinchcombe, & White, 1989), any continuous composition of immersion and projection (on compact domain) can be approximated arbitrarily closely by the structure.

The activities of the nodes in the third, or representation layer form global curvilinear coordinates on a submanifold of the input space (see Figure 1b). We thus refer to five-layer autoassociative networks as a global, nonlinear dimension reduction technique.

Several authors report successful implementation of nonlinear PCA using these networks for image (DeMers & Cottrell, 1993; Hecht-Nielsen, 1995; Kambhatla & Leen, 1994) and speech dimension reduction, for characterizing chemical dynamics (Kramer, 1991), and for obtaining concise representations of color (Usui et al., 1991).

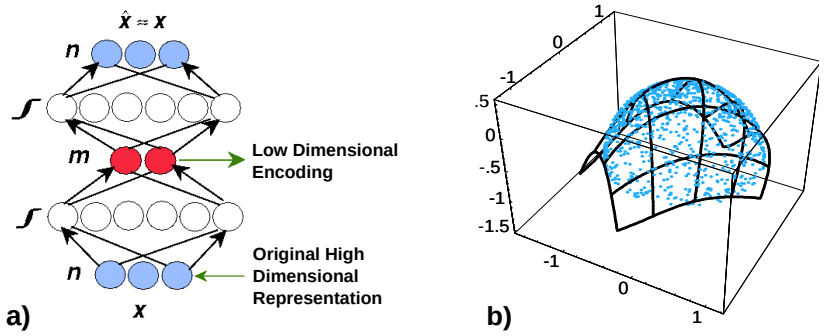

Figure 1: (a) Five-layer feedforward autoassociative network with inputs $x \in R^n$ and representation layer of dimension m. Outputs $x' \in R^n$ are trained to match the inputs, that is, to minimize $E[\,\|x - x'\|^2\,]$. (b) Global curvilinear coordinates built by a five-layer network for data distributed on the surface of a hemisphere. When the activations of the representation layer are swept, the outputs trace out the curvilinear coordinates shown by the solid lines.

3 Local Linear Transforms

Although five-layer autoassociators are convenient and elegant approximators for principal surfaces, they suffer from practical drawbacks. Networks with multiple sigmoidal hidden layers tend to have poorly conditioned Hessians (see Rognvaldsson, 1994, for a nice exposition) and are therefore difficult to train. In our experience, the variance of the solution with respect to weight initialization can be quite large (see section 4), indicating that the networks are prone to trapping in poor local optimal. We propose an alternative that does not suffer from these problems.

Our proposal is to construct local models, each pertaining to a different disjoint region of the data space. Within each region, the model complexity is limited; we construct linear models by PCA. If the local regions are small enough, the data manifold will not curve much over the extent of the region, and the linear model will be a good fit (low bias).

Schematically, the training algorithm is as follows:

1. Partition the input space R^n into Q disjoint regions $\{R^{(1)}, \ldots, R^{(Q)}\}$.
2. Compute the local covariance matrices

$$\Sigma^{(i)} = E[\,(x - Ex)(x - Ex)^T \mid x \in R^{(i)}\,]\,; \quad i = 1, \ldots, Q$$

and their eigenvectors $e_j^{(i)}$, $j = 1, \ldots, n$. Relabel the eigenvectors so

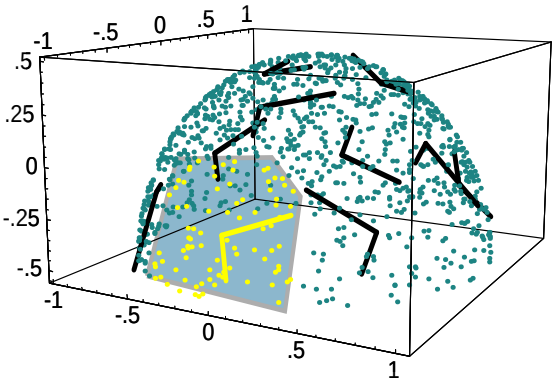

Figure 2: Coordinates built by local PCA for data distributed on the surface of a hemiphere. The solid lines are the two principal eigendirections for the data in each region. One of the regions is shown shaded for emphasis.

that the corresponding eigenvalues are in descending order $\lambda_1^{(i)} > \lambda_2^{(i)} > \cdots > \lambda_n^{(i)}$.

3. Choose a target dimension m and retain the leading m eigendirections for the encoding.

The partition is formed by VQ on the training data. The distortion measure for the VQ strongly affects the partition, and therefore the reconstruction error for the algorithm. We discuss two alternative distortion measures.

The local PCA defines local coordinate patches for the data, with the orientation of the local patches determined by the PCA within each region R_i. Figure 2 shows a set of local two-dimensional coordinate frames induced on the hemisphere data from figure 1, using the standard Euclidean distance as the VQ distortion measure.

3.1 Euclidean Partition. The simplest way to construct the partition is to build a VQ based on Euclidean distance. This can be accomplished by either an online competitive learning algorithm or by the generalized Lloyd algorithm (Gersho & Gray, 1992), which is the batch counterpart of competitive learning. In either case, the trained quantizer consists of a set of Q reference vectors $r^{(i)}$, $i = 1, \ldots, Q$ and corresponding regions $R^{(i)}$. The placement of the reference vectors and the definition of the regions satisfy Lloyd's optimality conditions:

1. Each region, $R^{(i)}$ corresponds to all $x \in R^n$ that lie closer to $r^{(i)}$ than to any other reference vector. Mathematically $R^{(i)} = \{x \mid d_E(x, r^{(i)}) <$

$d_E(x, r^{(j)}),\ \forall j \neq i\}$, where $d_E(a, b)$ is the Euclidean distance between a and b. Thus, a given x is assigned to its nearest neighbor r.

2. Each reference vector $r^{(i)}$ is placed at the centroid of the corresponding region $R^{(i)}$. For Euclidean distance, the centroid is the mean $r^{(i)} = E[x \mid x \in R^{(i)}]$.

For Euclidean distance, the regions are connected, convex sets called Voronoi cells.

As described in the introduction to section 3, one next computes the covariance matrices for the data in each region $R^{(i)}$ and performs a local PCA projection. The m-dimensional encoding of the original vector x is thus given in two parts: the index of the Voronoi region that the vector lies in and the local coordinates of the point with respect to the centroid, in the basis of the m leading eigenvectors of the corresponding covariance.[4] For example, if $x \in R^{(i)}$, the local coordinates are

$$z = \left(e_1^{(i)} \cdot (x - r^{(i)}), \ldots, e_m^{(i)} \cdot (x - r^{(i)})\right). \tag{3.1}$$

The decoded vector is given by

$$\hat{x} = r^{(i)} + \sum_{j=1}^{m} z_j\, e_j^{(i)}. \tag{3.2}$$

The mean squared reconstruction error incurred is

$$\mathcal{E}_{\text{recon}} = E[\|x - \hat{x}\|^2]. \tag{3.3}$$

3.2 Projection Partition. The algorithm described above is not optimal because the partition is constructed independent of the projection that follows. To understand the proper distortion function from which to construct the partition, consider the reconstruction error for a vector x that lies in $R^{(i)}$,

$$\begin{aligned} d(x, r^{(i)}) &\equiv \left\| x - r^{(i)} - \sum_{j=1}^{m} z_j\, e_j^{(i)} \right\|^2 = (x - r^{(i)})^T\, P^{(i)\,T}\, P^{(i)}\, (x - r^{(i)}) \\ &\equiv (x - r^{(i)})^T\, \Pi^{(i)}\, (x - r^{(i)}) \quad , \end{aligned} \tag{3.4}$$

where $P^{(i)}$ is the $m \times n$ matrix whose rows are the trailing eigenvectors of the covariance matrix $\Sigma^{(i)}$. The matrix $\Pi^{(i)}$ is the projection orthogonal to the local m-dimensional PCA subspace.

[4] The number of bits required to specify the region is small (between four and seven bits in all the experiments presented here) with respect to the number of bits used to express the double-precision coordinates within each region. In this respect, the specification of the region is nearly free.

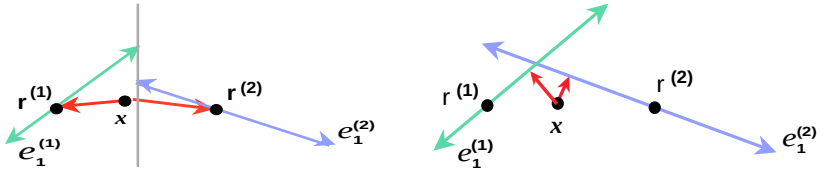

Figure 3: Assignment of the data point x to one of two regions based on (left) Euclidean distance and (right) the reconstruction distance. The reference vectors $r^{(i)}$ and leading eigenvector $e_1^{(i)}$ are shown for each of two regions ($i = 1, 2$). See text for explanation.

The reconstruction distance $d(x, r^{(i)})$ is the squared projection of the difference vector $x - r^{(i)}$ on the trailing eigenvectors of the covariance matrix for region $R^{(i)}$[5]. Equivalently, it is the squared Euclidean distance to the linear manifold that is defined by the local m-dimensional PCA in the ith local region. Clustering with respect to the reconstruction distance directly minimizes the expected reconstruction error $\mathcal{E}_{\text{recon}}$.

Figure 3 illustrates the difference between Euclidean distance and the reconstruction distance, with the latter intended for a one-dimensional local PCA. Suppose we want to determine to which of two regions the data point x belongs. For Euclidean clustering, the distance between the point x and the two centroids $r^{(1)}$ and $r^{(2)}$ is compared, and the data point is assigned to the cluster whose centroid is closest—in this case, region 1. For clustering by the reconstruction distance, the distance from the point to the two one-dimensional subspaces (corresponding to the principal subspace for the two regions) is compared, and the data point is assigned to the region whose principal subspace is closest—in this case, region 2. Data points that lie on the intersection of hyperplanes are assigned to the region with lower index.

Thus the membership in regions defined by the reconstruction distance can be different from that defined by Euclidean distance. This is because the reconstruction distance does not count the distance along the leading eigendirections. Neglecting the distance along the leading eigenvectors is exactly what is required, since we retain all the information in the leading directions during the PCA projection. Notice too that, unlike the Euclidean Voronoi regions, the regions arising from the reconstruction distance may not be connected sets.

Since the reconstruction distance (see equation 3.4) depends on the eigenvectors of $\Sigma^{(i)}$, an online algorithm for clustering would be prohibitively

[5] Note that when the target dimension m equals 0, the representation is reduced to the reference vector $r^{(i)}$ with no local coordinates. The distortion measure then reduces to the Euclidean distance.

expensive. Instead, we use the generalized Lloyd algorithm to compute the partition iteratively. The algorithm is:

1. Initialize the $r^{(i)}$ to randomly chosen inputs from the training data set. Initialize the $\Sigma^{(i)}$ to the identity matrix.
2. *Partition*. Partition the training data into Q regions $R^{(1)}, \ldots, R^{(Q)}$ where

$$R^{(i)} = \{x \mid d(x, r^{(i)}) \leq d(x, r^{(j)}); \text{ all } j \neq i\} \tag{3.5}$$

   with $d(x, r^{(i)})$ the reconstruction distance defined in (equation 3.4).
3. *Generalized centroid.* According to the Lloyd algorithm, the reference vectors $r^{(i)}$ are to be placed at the generalized centroid of the region $R^{(i)}$. The generalized centroid is defined by

$$r^{(i)} = \text{argmin}_r \frac{1}{N_i} \sum_{x \in R^{(i)}} (x - r)^T \Pi^{(i)} (x - r), \tag{3.6}$$

   where N_i is the number of data points in $R^{(i)}$. Expanding the projection operator Π in terms of the eigenvectors $e_j^{(i)}$, $j = m + 1, \ldots, n$ and setting to zero the derivative of the argument of the right-hand side of equation 3.6 with respect to r, one finds a set of equations for the generalized centroid[6] (Kambhatla, 1995),

$$\Pi^{(i)} r = \Pi^{(i)} \bar{x} \tag{3.7}$$

   where $\bar{x}$ is the mean of the data in $R^{(i)}$. Thus any vector r whose projection along the trailing eigenvectors equals the projection of $\bar{x}$ along the trailing eigenvectors is a generalized centroid of $R^{(i)}$. For convenience, we take $r = \bar{x}$. Next compute the covariance matrices

$$\Sigma^{(i)} = \frac{1}{N_i} \sum_{x \in R^{(i)}} (x - r^{(i)})(x - r^{(i)})^T$$

   and their eigenvectors $e_j^{(i)}$.
4. Iterate steps 2 and 3 until the fractional change in the average reconstruction error is below some specified threshold.

Following training, vectors are encoded and decoded as follows. To encode a vector x, find the reference vector $r^{(i)}$ that minimizes the reconstruction distance $d(x, r)$, and project $x - r^{(i)}$ onto the leading m eigenvectors

[6] In deriving the centroid equations, care must be exercised to take into account the dependence of $e_j^{(i)}$ (and hence $\Pi^{(i)}$) on $r^{(i)}$.

of the corresponding covariance matrix $\Sigma^{(i)}$ to obtain the local principal components

$$z = \left(e_1^{(i)} \cdot (x - r^{(i)}), \ldots, e_m^{(i)} \cdot (x - r^{(i)}) \right). \tag{3.8}$$

The encoding of x consists of the index i and the m local principal components z. The decoding, or reconstruction, of the vector x is

$$\hat{x} = r^{(i)} + \sum_{j=1}^{m} z_j \, e_j^{(i)}. \tag{3.9}$$

The clustering in the algorithm directly minimizes the expected reconstruction distance since it is a generalized Lloyd algorithm with the reconstruction distance as the distortion measure. Training in batch mode avoids recomputing the eigenvectors after each input vector is presented.

3.3 Accelerated Clustering. Vector quantization partitions data by calculating the distance between each data point x and all of the reference vectors $r^{(i)}$. The search and storage requirements for computing the partition can be streamlined by constraining the structure of the VQ. These constraints can compromise performance relative to what could be achieved with a standard, or unconstrained, VQ with the same number of regions. However, the constrained architectures allow a given hardware configuration to support a quantizer with many more regions than practicable for the unconstrained architecture, and they can thus improve speed and accuracy relative to what one can achieve in practice using an unconstrained quantizer. The two most common structures are the tree-search VQ and the multistage VQ (Gersho & Gray, 1992).

Tree-search VQ was designed to alleviate the search bottleneck for encoding. At the root of the tree, a partition into b_0 regions is constructed. At the next level of the tree, each of these regions is further partitioned into b_1 regions (often $b_0 = b_1 = \ldots$), and so forth. After k levels, each with branching ratio b, there are b^k regions in the partition. Encoding new data requires at most kb distortion calculations. The unconstrained quantizer with the same number of regions requires up to b^k distortion calculations. Thus the search complexity grows only logarithmically with the number of regions for the tree-search VQ, whereas the search complexity grows linearly with the number of regions for the unconstrained quantizer. However, the number of reference vectors in the tree is

$$\frac{b}{b-1}(b^k - 1),$$

whereas the unconstrained quantizer requires only b^k reference vectors. Thus the tree typically requires more storage.

Multistage VQ is a form of product coding and as such is economical in both search and storage requirements. At the first stage, a partition into b_0 regions is constructed. Then all of the data are encoded using this partition, and the residuals from *all* b_0 regions $\epsilon = x - r^{(i)}$ are pooled. At the next stage, these residuals are quantized by a b_1 region quantizer, and so forth. The final k-stage quantizer (again assuming b regions at each level) has b^k effective regions. Encoding new data requires at most kb distortion calculations.

Although the search complexity is the same as the tree quantizer, the storage requirements are more favorable than for either the tree or the unconstrained quantizer. There are a total b^k regions generated by the multistage architecture, requiring only bk reference vectors. The unconstrained quantizer would require b^k reference vectors to generate the same number of regions, and the tree requires more. The drawback is that the shapes of the regions in a multistage quantizer are severely restricted. By sketching regions for a two-stage quantizer with two or three regions per stage, the reader can easily show that the final regions consist of two or three shapes that are copied and translated to tile the data space. In constrast, an unconstrained quantizer will construct as many different shapes as required to encode the data with minimal distortion.

4 Experimental Results

In this section we present the results of experiments comparing global PCA, five-layer autoassociators, and several variants of VQPCA applied to dimension reduction of speech and image data. We compare the algorithms' training time and distortion in the reconstructed signal. The distortion measure we use is the reconstruction error normalized by the data variance,

$$\mathcal{E}_{\text{norm}} \equiv \frac{E[\|x - \hat{x}\|^2]}{E[\|x - E[x]\|^2]}, \tag{4.1}$$

where the expectations are with respect to empirical data distribution.

We trained the autoassociators using three optimization techniques: conjugate gradient descent (CGD), stochastic gradient descent (SGD), and the Broyden-Fletcher-Goldfarb-Shanno (BFGS) quasi-Newton method (Press, Flannery, Teukolsky, & Vetterling, 1987). The gradient descent code has a momentum term, and learning rate annealing as in Darken and Moody (1991).

We give results for VQPCA using both Euclidean and reconstruction distance, with unconstrained, multistage, and tree-search VQ. For the Euclidean distortion measure with an unconstrained quantizer, we used online competitive learning with the annealed learning rate schedule in Darken and Moody (1991).

We computed the global PCA for the speech data by diagonalizing the covariance matrices using Householder reduction followed by QL.[7] The image training set has fewer data points than the input dimension. Thus the covariance matrix is singular, and we cannot use Householder reduction. Instead we computed the PCA using singular-value decomposition applied to the data matrix.

All of the architecture selection was carried out by monitoring performance on a validation (or holdout) set. To limit the space of architectures, the autoassociators have an equal number of nodes in the second and fourth (sigmoidal) layers. These were varied from 5 to 50 in increments of 5. The networks were regularized by early stopping on a validation data set.

For VQPCA, we varied the number of local regions for the unconstrained VQ from 5 to 50 in increments of 5. The multistage VQ examples used two levels, with the number of regions at each level varied from 5 to 50 in increments of 5. The branching factor of the tree-structured VQ was varied from 2 to 9.

For all experiments and all architectures, we report only the results on those architectures that obtained the lowest reconstruction error on the validation data set. In this sense, we report the best results we obtained.

4.1 Dimension Reduction of Speech. We drew examples of the 12 monothongal vowels extracted from continuous speech in the TIMIT database (Fisher & Doddington, 1986). Each input vector consists of the lowest 32 discrete Fourier transform coefficients (spanning the frequency range 0–4 kHz), time-averaged over the central third of the vowel segment. The training set contains 1200 vectors, the validation set 408 vectors, and the test set 408 vectors. The test set utterances are from speakers not represented in the training or validation sets. Motivated by the desire to capture formant structure in the vowel encodings, we reduced the data from 32 to 2 dimensions.

The results of our experiments are shown in Table 1. Table 1 shows the mean and 2-σ error bars (computed over four random initializations) of the test set reconstruction error, and the Sparc 2 training times (in seconds). The numbers in parentheses are the values of architectural parameters. For example Autoassoc. (35) is a five-layer autoassociator with 35 nodes in each of the second and fourth layers. VQPCA-E-MS (40×40) is a Euclidean distortion, multistage VQPCA with 40 cells in each of two stages.

The VQPCA encodings have about half the reconstruction error of the global PCA or the five-layer networks. The latter failed to obtain significantly better results than the global PCA. The autoassociators show high variance in the reconstruction errors with respect to different random weight initializations. Several nets had higher error than PCA, indicating trapping

[7] The Householder algorithm reduces the covariance matrix to tridiagonal form, which is then diagonalized by the QL procedure (Press et al., 1987).

Table 1: Speech Dimension Reduction.

Algorithm	$\mathcal{E}_{\mathbf{norm}}$	**Training Time** (seconds)
PCA	0.443	11
Autoassoc.-CGD (35)	$0.496 \pm .103$	7784 ± 7442
Autoassoc.-BFGS (20)	$0.439 \pm .059$	3284 ± 1206
Autoassoc.-SGD (35)	$0.440 \pm .016$	$35{,}502 \pm 182$
VQPCA-Eucl (50)	$0.272 \pm .010$	1915 ± 780
VQPCA-E-MS (40×40)	$0.244 \pm .008$	144 ± 41
VQPCA-E-T (15×15)	$0.259 \pm .002$	195 ± 10
VQPCA-Recon (45)	$0.230 \pm .004$	864 ± 102
VQPCA-R-MS (45×45)	$0.208 \pm .005$	924 ± 50
VQPCA-R-T (9×9)	$0.242 \pm .005$	484 ± 128

in particularly poor local optima. As expected, stochastic gradient descent showed less variance due to initialization. In contrast, the local linear algorithms are relatively insensitive to initialization.

Clustering with reconstruction distance produced somewhat lower error than clustering with Euclidean distance. For both distortion measures, the multistage architecture produced the lowest error.

The five-layer autoassociators are very slow to train. The Euclidean multistage and tree-structured local linear algorithms trained more than an order of magnitude faster than the autoassociators. For the unconstrained, Euclidean distortion VQPCA, the partition was determined by online competitive learning and could probably be speeded up a bit by using a batch algorithm.

In addition to training time, practicality depends on the time required to encode and decode new data. Table 2 shows the number of floating-point operations required to encode and decode the entire database for the different algorithms. The VQPCA algorithms, particularly those using the reconstruction distance, require many more floating-point operations to encode the data than the autoassociators. However, the decoding is much faster than that for autoassociators and is comparable to PCA. [8] The results indicate that VQPCA may not be suitable for real-time applications like videoconferencing where very fast encoding is desired. However, when only the decoding speed is of concern (e.g., for data retrieval), VQPCA algorithms are a good choice because of their accuracy and fast decoding.

[8] However, parallel implementation of the autoassociators could outperform the VQPCA decode in terms of clock time.

Table 2: Encoding and Decoding Times for the Speech Data.

Algorithm	**Encode Time** (FLOPs)	**Decode Time** (FLOPs)
PCA	158	128
Autoassoc.-CGD (35)	2380	2380
Autoassoc.-BFGS (20)	1360	1360
Autoassoc.-SGD (35)	2380	2380
VQPCA-Eucl (50)	4957	128
VQPCA-E-MS (40×40)	7836	192
VQPCA-E-T (15×15)	3036	128
VQPCA-Recon (45)	87,939	128
VQPCA-R-MS (45×45)	96,578	192
VQPCA-R-T (9×9)	35,320	128

In order to test variability of these results across different training and test sets, we reshuffled and repartitioned the data into new training, validation, and tests sets of the same size as those above. The new data sets gave results very close to those reported here (Kambhatla, 1995).

4.2 Dimension Reduction of Images. Our image database consists of 160 images of the faces of 20 people. Each image is a 64×64, 8-bit/pixel grayscale image. The database was originally generated by Cottrell and Metcalfe and used in their study of identity, gender, and emotion recognition (Cottrell & Metcalfe, 1991). We adopted the database, and the preparation used here, in order to compare our dimension reduction algorithms with the nonlinear autoassociators used by DeMers and Cottrell (1993).

As in DeMers and Cottrell (1993), each image is used complete, as a 4096-dimensional vector, and is preprocessed by extracting the leading 50 principal components computed from the ensemble of 160 images. Thus the base dimension is 50. As in DeMers and Cottrell (1993), we examined reduction to five dimensions.

We divided the data into a training set containing 120 images, a validation set of 20 images, and a test set of 20 images. We used PCA, five-layer autoassociators, and VQPCA for reduction to five dimensions. Due to memory constraints, the autoassociators were limited to 5 to 40 nodes in each of the second and fourth layers. The autoassociators and the VQPCA were trained with four different random initializations of the free parameters.

The experimental results are shown in Table 3. The five-layer networks attain about 30 percent lower error than global PCA. VQPCA with either Euclidean or reconstruction distance distortion measures attain about 40 per-

Table 3: Image Dimension Reduction.

Algorithm	$\mathcal{E}_{\mathbf{norm}}$	**Training Time** (seconds)
PCA	0.463	5
Autoassoc.-CGD (35)	$0.441 \pm .090$	698 ± 533
Autoassoc.-BFGS (35)	$0.377 \pm .127$	$18{,}905 \pm 15{,}081$
Autoassoc.-SGD (25)	$0.327 \pm .027$	4171 ± 41
VQPCA-Eucl (20)	$0.179 \pm .048$	202 ± 57
VQPCA-E-MS (5×5)	$0.307 \pm .031$	14 ± 2
VQPCA-E-Tree (4×4)	$0.211 \pm .064$	31 ± 9
VQPCA-Recon (20)	$0.173 \pm .050$	62 ± 5
VQPCA-R-MS (20×20)	$0.240 \pm .042$	78 ± 32
VQPCA-R-Tree (5×5)	$0.218 \pm .029$	79 ± 15

Table 4: Encoding and Decoding Times (FLOPs) for the Image Data.

Algorithm	**Encode Time** (FLOPs)	**Decode Time** (FLOPs)
PCA	545	500
Autoassoc.-CGD (35)	3850	3850
Autoassoc.-BFGS (35)	3850	3850
Autoassoc.-SGD (25)	2750	2750
VQPCA-Eucl (20)	3544	500
VQPCA-E-MS (5×5)	2043	600
VQPCA-E-T (4×4)	1743	500
VQPCA-Recon (20)	91,494	500
VQPCA-R-MS (20×20)	97,493	600
VQPCA-R-T (5×5)	46,093	500

cent lower error than the best autoassociator. There is little distinction between the Euclidean and reconstruction distance clustering for these data.

The VQPCA trains significantly faster than the autoassociators. Although the conjugate gradient algorithm is relatively quick, it generates encodings inferior to those obtained with the stochastic gradient descent and BFGS simulators.

Table 4 shows the encode and decode times for the different algorithms. We again note that VQPCA algorithms using reconstruction distance clustering require many more floating-point operations (FLOPs) to encode an

Table 5: Image Dimension Reduction: Training on All Available Data.

Algorithm	$\mathcal{E}_{\mathbf{norm}}$	**Training Time** (seconds)
PCA	0.405	7
Autoassoc.-SGD (30)	0.103	25,306
Autoassoc.-SGD (40)	0.073	31,980
VQPCA-Eucl (25)	0.026	251
VQPCA-Recon (30)	0.022	116

input vector than does the Euclidean distance algorithm or the five-layer networks. However, as before, the decode times are much less for VQPCA. As before, shuffling and repartitioning the data into training, validation, and test data sets and repeating the experiments returned results very close to those given here.

Finally, in order to compare directly with DeMers and Cottrell's (1993) results, we also conducted experiments training with all the data (no separation into validation and test sets). This is essentially a model fitting problem, with no influence from statistical sampling. We show results only for the autoassociators trained with SGD, since these returned lower error than the conjugate gradient simulators, and the memory requirements for BFGS were prohibitive. We report the results from those architectures that provided the lowest reconstruction error on the training data.

The results are shown in Table 5. Both nonlinear techniques produce encodings with lower error than PCA, indicating significant nonlinear structure in the data. For the same data and using a five-layer autoassociator with 30 nodes in each of the second and fourth layers, DeMers and Cottrell (1993) obtain a reconstruction error $\mathcal{E}_{norm} = 0.1317$.[9] This is comparable to our results. We note that the VQPCA algorithms train two orders of magnitude faster than the networks while obtaining encodings with about one-third the reconstruction error.

It is useful to examine the images obtained from the encodings for the various algorithms. Figure 4 shows two sample images from the data set along with their reconstructions from five-dimensional encodings. The algorithms correspond to those reported in Table 5. These two images were selected because their reconstruction error closely matched the average. The left-most column shows the images as reconstructed from the 50 principal components. The second column shows the reconstruction from 5 princi-

[9] DeMers and Cottrell report half the MSE per output node, $E = (1/2) * (1/50) * MSE = 0.001$. This corresponds to $\mathcal{E}_{norm} = 0.1317$.

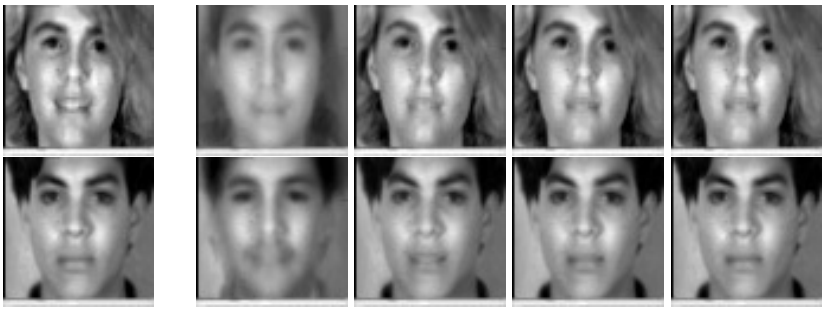

Figure 4: Two representative images: Left to right—Original 50 principal components reconstructed image, reconstruction from 5-D encodings: PCA, Autoassoc-SGD(40), VQPCAEucl(25), and VQPCA-Recon(30). The normalized reconstruction errors and training times for the whole data set (all the images) are given in Table 5.

pal components. The third column is the reconstruction from the five-layer autoassociator, and the last two columns are the reconstruction from the Euclidean and reconstruction distance VQPCA.

The five-dimensional PCA has grossly reduced resolution, and gray-scale distortion (e.g., the hair in the top image). All of the nonlinear algorithms produce superior results, as indicated by the reconstruction error. The lower image shows a subtle difference between the autoassociator and the two VQPCA reconstructions; the posture of the mouth is correctly recovered in the latter.

5 Discussion

We have applied locally linear models to the task of dimension reduction, finding superior results to both PCA and the global nonlinear model built by five-layer autoassociators. The local linear models train significantly faster than autoassociators, with their training time dominated by the partitioning step. Once the model is trained, encoding new data requires computing which region of the partition the new data fall in, and thus VQPCA requires more computation for encoding than does the autoassociator. However, decoding data with VQPCA is faster than decoding with the autoassociator and comparable to decoding with PCA. With these considerations, VQPCA is perhaps not optimal for real-time encoding, but its accuracy and computational speed for decoding make it superior for applications like image retrieval from databases.

In order to optimize the partition with respect to the accuracy of the projected data, we introduced a new distortion function for the VQ, the reconstruction distance. Clustering with the reconstruction distance does indeed provide lower reconstruction error, though the difference is data dependent.

We cannot offer a definitive reason for the superiority of the VQPCA representations over five-layer autoassociators. In particular, we are uncertain how much of the failure of autoassociators is due to optimization problems and how much to the representation capability. We have observed that changes in initialization result in large variability in the reconstruction error of solutions arrived at by autoassociators. We also see strong dependence on the optimization technique. This lends support to the notion that some of the failure is a training problem. It may be that the space of architectures we have explored does have better solutions but that the available optimizers fail to find them.

The optimization problem is a very real block to the effective use of five-layer autoassociators. Although the architecture is an elegant construction for nonlinear dimension reduction, realizing consistently good maps with standard optimization techniques has proved elusive on the examples we considered here.

Optimization may not be the only issue. Autoassociators constructed from neurons with smooth activation functions are constrained to generate smooth projections and immersions. The VQPCA algorithms have no inherent smoothness constraint. This will be an advantage when the data are not well described by a manifold.[10] Moreover, Malthouse (1996) gives a simple example of data for which continuous projections are *necessarily* suboptimal.

In fact, if the data are not well described by a manifold, it may be advantageous to choose the representation dimension locally, allowing a different dimension for each region of the partition. The target dimension could be chosen using a hold out data set, or more economically by a cost function that includes a penalty for model complexity. Minimum description length (MDL) criteria have been used for PCA (Hediger et al., 1990; Wax & Kailath, 1985) and presumably could be applied to VQPCA.

We have explored no such methods for estimating the appropriate target dimension. In contrast, the autoassociator algorithm given by DeMers and Cottrell (1993) includes a pruning strategy to reduce progressively the dimensionality of the representation layer, under the constraint that the reconstruction error not grow above a desired threshold.

Two applications deserve attention. The first is transform coding for which the algorithms discussed here are naturally suited. In transform coding, one transforms the data, such as image blocks, to a lower-redundancy

[10] A visualization of just such a case is given in Kambhatla (1995).

representation and then scalar quantizes the new representation. This produces a product code for the data. Standard approaches include preprocessing by PCA or discrete cosine transform, followed by scalar quantization (Wallace, 1991). As discussed in the introduction, the nonlinear transforms considered here provide more accurate representations than PCA and should provide for better transform coding.

This work suggests a full implementation of transform coding, with comparisons betweeen PCA, autoassociators, and VQPCA in terms of rate distortion curves. Transform coding using VQPCA with the reconstruction distance clustering requires additional algorithm development. The reconstruction distance distortion function depends explicitly on the target dimension, while the latter depends on the allocation of transform bits between the new coordinates. Consequently a proper transform coding scheme needs to couple the bit allocation to the clustering, an enhancement that we are developing.

The second potential application is in novelty detection. Recently several authors have used three-layer autoassociators to build models of normal equipment function (Petsche et al., 1996; Japkowicz et al., 1995). Equipment faults are then signaled by the failure of the model to reconstruct the new signal accurately. The nonlinear models provided by VQPCA should provide more accurate models of the normal data, and hence improve the sensitivity and specificity for fault detection.

Acknowledgments

This work was supported in part by grants from the Air Force Office of Scientific Research (F49620-93-1-0253) and the Electric Power Research Institute (RP8015-2). We thank Gary Cottrell and David DeMers for supplying image data and the Center for Spoken Language Understanding at the Oregon Graduate Institute of Science and Technology for speech data. We are grateful for the reviewer's careful reading and helpful comments.

References

Baldi, P., & Hornik, K. (1989). Neural networks and principal component analysis: Learning from examples without local minima. *Neural Networks, 2*, 53–58.

Bourlard, H., & Kamp, Y. (1988). Auto-association by multilayer perceptrons and singular value decomposition. *Biol. Cyb., 59*, 291–294.

Bregler, C., & Omohundro, S. M. (1995). Nonlinear image interpolation using manifold learning. In G. Tesauro, D. Touretzky, & T. Leen (Eds.), *Advances in neural information processing systems 7*. Cambridge, MA: MIT Press.

Broomhead, D. S. (1991, July). Signal processing for nonlinear systems. In S. Haykin (Ed.), *Adaptive Signal Processing, SPIE Proceedings* (pp. 228–243). Bellingham, WA: SPIE.

Cleveland, W. S., & Devlin, S. J. (1988). Locally weighted regression: An approach to regression analysis by local fitting. *J. Amer. Stat. Assoc., 83*, 596–610.

Coetzee, F. M., & Stonick, V. L. (1995). Topology and geometry of single hidden layer network, least squares weight solutions. *Neural Computation, 7*, 672–705.

Cottrell, G. W., & Metcalfe, J. (1991). EMPATH: Face, emotion, and gender recognition using holons. In R. Lippmann, J. Moody, & D. Touretzky (Eds.), *Advances in neural information processing systems 3* (pp. 564–571). San Mateo, CA: Morgan Kaufmann.

Cottrell, G. W., Munro, P., & Zipser, D. (1987). Learning internal representations from gray-scale images: An example of extensional programming. In *Proceedings of the Ninth Annual Cognitive Science Society Conference* (pp. 461–473). Seattle, WA.

Darken, C., & Moody, J. (1991). Note on learning rate schedules for stochastic optimization. In R. Lippman, J. Moody, D. Touretzky, (Eds.), *Advances in neural information processing systems 3*. San Mateo, CA: Morgan Kaufmann.

DeMers, D., & Cottrell, G. (1993). Non-linear dimensionality reduction. In C. Giles, S. Hanson, & J. Cowan (Eds.), *Advances in neural information processing systems 5*. San Mateo, CA: Morgan Kaufmann.

Dony, R. D., & Haykin, S. (1995). Optimally adaptive transform coding. *IEEE Transactions on Image Processing* (pp. 1358–1370).

Fisher, W. M., & Doddington, G. R. (1986). The DARPA speech recognition research database: Specification and status. In *Proceedings of the DARPA Speech Recognition Workshop* (pp. 93–99). Palo Alto, CA.

Fukunaga, K., & Olsen, D. R. (1971). An algorithm for finding intrinsic dimensionality of data. *IEEE Transactions on Computers, C-20*, 176–183.

Funahashi, K. (1989). On the approximate realization of continuous mappings by neural networks. *Neural Networks, 2*, 183–192.

Gersho, A., & Gray, R. M. (1992). *Vector quantization and signal compression*. Boston: Kluwer.

Golomb, B. A., Lawrence, D. T., & Sejnowski, T. J. (1991). Sexnet: A neural network identifies sex from human faces. In R. Lippmann, J. Moody, & D. Touretzky (Eds.), *Advances in neural information processing systems 3* (pp. 572–577). San Mateo, CA: Morgan Kauffmann.

Gray, R. M.(1984, April). Vector quantization. *IEEE ASSP Magazine*, pp. 4–29.

Hastie, T. (1984). *Principal curves and surfaces*. Unpublished dissertation, Stanford University.

Hastie, T., & Stuetzle, W. (1989). Principal curves. *Journal of the American Statistical Association, 84*, 502–516.

Hecht-Nielsen, R. (1995). Replicator neural networks for universal optimal source coding. *Science, 269*, 1860–1863.

Hediger, T., Passamante, A., & Farrell, M. E. (1990). Characterizing attractors using local intrinsic dimensions calculated by singular-value decomposition and information-theoretic criteria. *Physical Review, A41*, 5325–5332.

Hinton, G. E., Revow, M., & Dayan, P. (1995). Recognizing handwritten digits using mixtures of linear models. In G. Tesauro, D. Touretzky, & T. Leen (Eds.), *Advances in neural information processing systems 7*. Cambridge, MA: MIT Press.

Hornik, M., Stinchcombe, M., & White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural Networks, 2*, 359–368.

Japkowicz, N., Myers, C., & Gluck, M. (1995). A novelty detection approach to classification. In *Proceedings of IJCAI*.

Kambhatla, N. (1995) *Local models and gaussian mixture models for statistical data processing*. Unpublished doctoral dissertation, Oregon Graduate Institute.

Kambhatla, N., & Leen, T. K. (1994). Fast non-linear dimension reduction. In J. D. Cowan, G. Tesauro, & J. Alspector (Eds.), *Advances in neural information processing systems 6*. San Mateo, CA: Morgan Kaufmann.

Kramer, M. A. (1991). Nonlinear principal component analysis using autoassociative neural networks. *AIChE Journal, 37*, 233–243.

Kung, S. Y., & Diamantaras, K. I. (1990). A neural network learning algorithm for adaptive principal component extraction (APEX). In *Proceedings of the IEEE International Conference on Acoustics Speech and Signal Processing* (pp. 861–864).

Malthouse, E. C. (1996). Some theoretical results on non-linear principal components analysis (Unpublished research report). Evanston, IL: Kellogg School of Management, Northwestern University.

Oja, E. (1982). A simplified neuron model as a principal component analyzer. *J. Math. Biology, 15*, 267–273.

Oja, E. (1989). Neural networks, principal components, and subspaces. *International Journal of Neural Systems, 1*, 61–68.

Oja, E. (1991). Data compression, feature extraction, and autoassociation in feedforward neural networks. In *Artificial neural networks* (pp. 737–745). Amsterdam: Elsevier Science Publishers.

Petsche, T., Marcantonio, A., Darken, C., Hanson, S. J., Kuhn, G. M., & Santoso, I. (1996) A neural network autoassociator for induction motor failure prediction. In D. S. Touretzky, M. C. Mozer, & M. E. Hasselmo (Eds.), *Advances in neural information processing systems 8*. Cambridge, MA: MIT Press.

Press, W. H., Flannery, B. P., Teukolsky, S. A., & Vetterling, W. T. (1987). *Numerical recipes—the art of scientific computing*. Cambridge: Cambridge University Press.

Rognvaldsson, T. (1994). On Langevin updating in multilayer perceptrons. *Neural Computation, 6*, 916–926.

Rubner, J., & Tavan, P. (1989). A self-organizing network for principal component analysis. *Europhysics Lett., 20*, 693–698.

Sanger, T. (1989). An optimality principle for unsupervised learning. In D. S. Touretzky (ed.), *Advances in neural information processing systems 1*. San Mateo, CA: Morgan Kaufmann.

Usui, S., Nakauchi, S., & Nakano, M. (1991). Internal color representation acquired by a five-layer neural network. In O. Simula, T. Kohonen, K. Makisara, & J. Kangas (Eds.), *Artificial neural networks*. Amsterdam: Elsevier Science Publishers, North-Holland.

Wallace, G. K. (1991). The JPEG still picture compression standard. *Communications off the ACM, 34*, 31–44.

Wax, M., & Kailath, T. (1985). Detection of signals by information theoretic criteria. *IEEE Transactions on Acoustics, Speech and Signal Processing, ASSP-33*(2), 387–392.

Yang, L., & Yu, W. (1993). Backpropagation with homotopy. *Neural Computation*, *5*, 363–366.

Received September 6, 1996; accepted February 28, 1997.