PERGAMON

Pattern Recognition 34 (2001) 203–214

PATTERN RECOGNITION
THE JOURNAL OF THE PATTERN RECOGNITION SOCIETY

www.elsevier.com/locate/patcog

A nonlinear neural network model of mixture of local principal component analysis: application to handwritten digits recognition

Bailing Zhang[a,]*, Minyue Fu[b], Hong Yan[c]

[a]*System Engineering and Design Automation Laboratory (SEDAL), School of Electrical and Information Engineering, University of Sydney, NSW 2006, Australia*

[b]*Department of Electrical and Computer Engineering, The University of Newcastle, NSW 2308, Australia*

[c]*Imaging Science and Engineering Laboratory, School of Electrical and Information Engineering, University of Sydney, NSW 2006, Australia*

Received 10 December 1998; received in revised form 30 August 1999; accepted 30 August 1999

Abstract

Principal component analysis (PCA) is a popular tool in multivariate statistics and pattern recognition. Recently, some mixture models of local principal component analysis have attracted attention due to a number of benefits over global PCA. In this paper, we propose a mixture model by concurrently performing global data partition and local linear PCA. The partition is optimal or near optimal, which is realized by a soft competition algorithm called 'neural gas'. The local PCA type representation is approximated by a neural learning algorithm in a nonlinear autoencoder network, which is set up on the generalization of the least-squares reconstruction problem leading to the standard PCA. Such a local PCA type representation has a number of numerical advantages, for example, faster convergence and insensitive to local minima. Based on this mixture model, we describe a modular classification scheme to solve the problem of handwritten digits recognition. We use 10 networks (modules) to capture different features in the 10 classes of handwritten digits, with each network being a mixture model of local PCA type representations. When a test digit is presented to all the modules, each module provides a reconstructed pattern by a prescribed principle and the system outputs the class label by comparing the reconstruction errors from the 10 networks. Compared with some traditional neural network-based classifiers, our scheme converges faster and recognizes with higher accuracy. For a relatively small size of each module, the classification accuracy reaches 98.6% on the training set and 97.8% on the testing set. © 2000 Pattern Recognition Society. Published by Elsevier Science Ltd. All rights reserved.

Keywords: Principal component analysis; Subspace pattern recognition; Mixture model; Handwritten digit recognition

1. Introduction

Principal component analysis (PCA) is a general purpose tool in pattern recognition for extracting ensemble features that reflect global statistical properties of pattern space. By PCA, an original pattern space which is likely of high dimension is approximated by a subspace of lower dimension, which is spanned by the first several principal eigenvectors of the data covariance matrix Σ. The well-known subspace pattern recognition method [1,3] is set up on PCA, in which each pattern class is represented by a subspace. Subspace approach has a number of advantages, for example, it is scale-invariant, and the two phases of most classification systems, i.e., feature extraction and class representation, are actually combined. Originating from Oja's [2] work on extracting the first principal component by a linear neuron model, the issue of neural learning PCA has risen great interests in recent years [5,6].

*Corresponding author. Tel.: + 61-2-9351-7207; fax: + 61-2-9351-7209.

E-mail address: bailing@ee.usyd.edu.au (B. Zhang).

PII: S0031-3203(00)00009-1

A limitation of PCA is its reliance on second-order statistics and linear projection. This has motivated various developments of nonlinear generalizations of PCA. One method is to model nonlinear structure in the data by some mixtures of local principal component projections [8,9,11,12]. A mixture model of local PCA usually involves two procedures: partition a data set into a number of nonoverlapping regions and each region is represented not by its central point as in clustering but by a localized linear subspace. It is easy to understand that such a mixture model has no unique definition and its performances depend on global partition and local linear fitting.

Kohonen proposed a modular neural network architecture called adaptive-subspace self-organizing map (ASSOM) [11], which creats a set of subspace representations by competitive selection and learning. Independently, Dony and Haykin [8] and Kambhatla and Leen [12] all proposed a kind of VQ/PCA mixture models which first partition the data into disjoint regions by vector quantization (VQ) and then perform a local PCA about each cluster center. In these works, the reconstruction error is employed as the relevant distortion measure for determining the partitions. Hinton et al. [9] considered the partition assignments of N examples among M different PCA models (called submodels) in both k-means clustering procedure and the expectation maximization (EM) framework. In the k-means clustering procedure, a given data point is assigned exclusively to one submodel for which the reconstruction error is smallest. The k-means clustering has some disadvantages, especially the under-utilization of submodels. The soft EM algorithm is to let the submodels share the responsibilities for reconstructing or generating a data point. Though EM algorithm has some advantages, it is prone to encounter local maxima problem which makes the performance dependent on the initial values.

Based on the above progresses, in this article, we propose an improved mixture model which has the following two features. Firstly, the data manifold is partitioned by generalizing a near optimal clustering algorithm called ‘neural gas’ [13]. Compared with k-means clustering [14], maximum-entropy clustering [15], ‘neural gas’ algorithm converges very quickly to a much lower distortion error. Secondly, the local principal component analysis is also realized by a neural learning algorithm which is quick, stable and robust to outliers, mainly due to introducing sigmoidal nonlinearities to the projections [16–18]. A local principal component analysis can be generally implemented in an autoencoder network which performs identity mapping. In this paper, we consider a simple feedforward network with a single hidden layer, for which an input pattern is duplicated at the output layer as the desired output. Such an autoencoder can be treated as a nonparametric statistical model for fitting data space, by which an input vector is represented as the hidden units' activations and the combination of these activations using the weights between the hidden units and output units provides a reconstruction of the input pattern. By generalizing the optimization problem of the least square reconstruction from the standard PCA to nonlinear autoencoder networks, a numerically advantageous neural learning algorithm for PCA type representation has been proposed [16–18], which is utilized for local representation in our mixture model.

Our mixture model can be applied to classification as a generalization of traditional PCA-based subspace classifier. An important problem of measuring goodness-of-fit of a mixture model is tackled by two approaches. In the first approach, we simply compare all the reconstruction errors provided by the local principal subspaces and choose the smallest one as a measure of fitting the data by the mixture model. In the second approach, we define a “distance” between a data point and the averaged subspaces corresponding to different submodels. For practical multiclass problems, we train a separate mixture model to describe each class of data and then classify unknown data points according to whichever model yields the best fit.

As an application, we apply our mixture model to handwritten digit recognition. In the past several decades, a large number of approaches have been proposed to design a recognition system for handwritten digits [19]. The approaches can be roughly categorized into two types, i.e., statistical method and syntactic method. Statistical methods represent a pattern by a set of feature vectors and classification is based on some similarity measures such as a distance metric or a discriminant function. Important examples include K-nearest-neighbor classifier, template matching, etc. Syntactic methods represent a pattern as a string, a tree, or a graph of pattern primitives and their relations. Pattern primitives are some important shape features of the digits, generally taken from their skeletons or contours such as loops, junctions, arcs, etc. Classification is usually built upon a syntax analysis. Although much progress has been achieved, handwritten digit recognition remains a difficult problem, mainly because it is often hard to characterize the wide diversity inherent in handwritten digits.

In recent years, neural network techniques have often been applied in handwritten digits recognition [9,19–24]. Compared with classical statistical techniques, neural-networks-based classifiers often bear some advantages such as being more tolerant and robust when dealing with complex real data. In some traditional neural network recognition systems, a set of features is extracted and then a neural network serves as a classifier. For example, Le Cun et al. achieved a very good result with a backpropagation network using size-normalized images as input [20]. The network is highly constrained as specifically designed. In many previous works, a neural network classifier is trained to output a unique class label indicating that the input pattern belongs to this class. Such a paradigm has some disadvantages, as have been discussed in [9]. As handwritten digits recognition is a multiclass classification problem, a better alternative is to apply a modularity [23]. In a modular classification system, modules learn to specialize in different subtasks. For example, a reasonable approach is to train an individual autoencoder network on examples of each digit class and then to recognize digits by deciding which autoencoder offers the best reconstruction of the data [23]. Each network learns a low-dimensional hidden layer representation and gives reconstructions which are similar to the subspace projections of the examples of one class. The learned networks can be also considered as discriminant functions from the viewpoint that a reconstruction error is a matching score for measuring the degree that a test pattern belongs to the specific class.

In this paper, we present a modular classification scheme for handwritten digit recognition, in which each module is a mixture model of local PCA for modeling the manifolds of handwritten digits bitmaps. Each module is composed of a number of submodels which corresponds to a subspace in the respective class. An individual module is trained only by the images belonging to each digit class. During classification, upon presenting a test pattern, each module provides its own reconstruction according to a prescribed principle and the overall decision is determined by comparing all of the reconstruction errors. Our scheme generalizes the traditional subspace pattern recognition method in the sense that we find a set of subspaces for each digit class and the classification is performed by first finding the closest (or averaged) reconstructions from all the mixture modules which can be compared to subspace projections, and then assigning the associated class. Our results show that the mixture model is very effective in the recognition accuracy. With each module having 40 units, the recognition rate arrives at 98.6% on the training set and 97.8% on the testing set, with no rejection.

This paper is organized as follows. In the next section, we first review some neural learning algorithms for PCA and subspace pattern recognition, which are based on the least-squares reconstruction principle. The learning performance for a single neural unit (or a symmetrical autoencoder with one hidden node) is discussed in more detail, as it is mainly employed in our method. In Section 3, after first introducing the 'neural gas' algorithm for clustering, we propose a mixture model of local principal component analysis by performing competition based on a generalized 'neural gas' algorithm, in which the reconstruction errors provided by the submodules serve as distortion measures for making partition. In Section 4, we define a generalized reconstruction distance between an input pattern and a trained mixture model of local PCA, which can be considered as a 'goodness-of-fit' measure for describing the data. A modular classification system is then set up for the handwritten digits recognition, with detailed experiment results reported. Finally, discussion and concluding remarks are given in Section 5.

2. Neural learning principal component analysis and subspace pattern recognition

2.1. Principal component analysis

Assume that $\mathbf{x}$ is an L-dimensional input data vector which is assumed to be zero mean. The purpose of PCA is to find those M ($M \leqslant L$) linear combinations $\mathbf{w}_1^{\mathrm{T}}\mathbf{x}$, $\mathbf{w}_2^{\mathrm{T}}\mathbf{x}, \dots, \mathbf{w}_M^{\mathrm{T}}\mathbf{x}$ of the elements of $\mathbf{x}$ that satisfy [5]

1. $E\{(\mathbf{w}_i^{\mathrm{T}}\mathbf{x})^2\}$, $i = 1, \dots, M$, are maximized, under the constraints,
2. $\mathbf{w}_i^{\mathrm{T}}\mathbf{w}_j = \delta_{ij}$, for $j < i$,

where E stands for an expectation operator.

The solution for the vectors $\mathbf{w}_1, \dots, \mathbf{w}_M$ are the M dominant eigenvectors of the data covariance matrix

$$\Sigma = E\{\mathbf{x}\mathbf{x}^{\mathrm{T}}\}. \tag{1}$$

These are the M orthogonal unit vectors $\mathbf{c}_1, \dots, \mathbf{c}_M$ given by

$$\Sigma\mathbf{c}_i = \lambda_i\mathbf{c}_i, \tag{2}$$

where $\lambda_1, \lambda_2, \dots, \lambda_M$ are the M largest eigenvalues of Σ in descending order of magnitude. The first linear combination $\mathbf{c}_1^{\mathrm{T}}\mathbf{x}$ is called the first principal component [5].

The r principal components of $\mathbf{x}$'s distribution capture the greatest variation in the distribution. In other words, among all the subspaces onto which data samples can be projected, principal component subspace is such that the variance of the projected examples is maximal. If the data samples have a multivariate Gaussian distribution, then the information is maximally conveyed by the magnitude of the projections onto these r principal component directions.

2.2. Learning first principal component by a linear neuron

A linear neuron model with weight vector $\mathbf{w}$, input sample $\mathbf{x}$ and output $y = \mathbf{w}^{\mathrm{T}}\mathbf{x}$ can learn the largest principal component [2], which can be achieved by optimizing the reconstruction mean-square error, i.e., the following objective:

$$\begin{aligned}\text{minimize } J &= E\{\|\mathbf{x} - \hat{\mathbf{x}}\|^2\} \\ &= E\{\|\mathbf{x} - y\mathbf{w}\|^2\}\end{aligned} \tag{3}$$

will result in the largest component direction of the distribution of $\mathbf{x}$. In other words, $\mathbf{w}$ converges to a unit eigenvector corresponding to the largest eigenvalue of Σ. In the above equation, $\hat{\mathbf{x}} = y\mathbf{w}$ is the reconstruction of input. The corresponding stochastic gradient descent learning rule is [15,16]

$$\mathbf{w}_{t+1} = \mathbf{w}_t + \mu_t(\mathbf{x}_t\mathbf{e}_t^{\mathrm{T}}\mathbf{w}_t + \mathbf{e}_t\mathbf{x}_t^{\mathrm{T}}\mathbf{w}_t), \tag{4}$$

where $\mathbf{e}_t = \mathbf{x} - y\mathbf{w}_t$, t is a time scale and μ_t is a learning rate.

A number of unsupervised learning algorithms for extracting multiple principal components or their subspace have been proposed, usually developed from the objective (1) (or maximum variance objective) and the following consideration: the second largest principal component also satisfies the minimal reconstruction property with restriction that the second principal component direction must be othogonal to the first component direction, and so on for the remaining principal component directions.

2.3. Extensions of PCA learning to nonlinear neural networks

The extension of (3) to a nonlinear unit has been proposed in several papers [16–18]. With a nonlinear neuron $z = f(\mathbf{w}^{\mathrm{T}}\mathbf{x})$ and a simple reconstruction of input vector $\mathbf{x}$ represented as $\hat{\mathbf{x}} = z\mathbf{w}$, the learning objective is then

$$\begin{aligned}\text{minimize } J &= E\{\|\mathbf{x} - \hat{\mathbf{x}}\|^2\} \\ &= E\{\|\mathbf{x} - z\mathbf{w}\|^2\}.\end{aligned} \tag{5}$$

In this paper, we take f as a sigmoidal function bounded between 0 and 1.

A stochastic approximization approach will lead to the following learning rule:

$$\mathbf{w}_{t+1} = \mathbf{w}_t + \mu_t(\mathbf{e}_t^{\mathrm{T}}\mathbf{w}_t\mathbf{x}g + z\mathbf{e}_t), \tag{6}$$

where $\mathbf{e}_t = \mathbf{x} - z\mathbf{w}_t$ is the reconstruction error, g is the derivative of z, and $g = f'(\mathbf{w}_t^{\mathrm{T}}\mathbf{x})$. The proof of Eq. (6) can be found in Refs. [16,17].

Learning objective (5) can be further extended to a network with M neurons. In this case, the network can be treated as autoencoder, which has equal number L of input and output nodes and M hidden nodes, $M < L$. During learning, the inputs are duplicated at the output nodes to perform identity mapping. $L \times M$ matrix $\mathbf{W}$, $M \times L$ matrix $\hat{\mathbf{W}}$ denote the connection weight from input to hidden nodes and from hidden nodes to output, respectively. In our work, we only consider the symmetrical case, i.e., $\hat{\mathbf{W}} = \mathbf{W}^{\mathrm{T}}$. Let $\mathbf{w}(m)$ be the weight vector associated with mth neuron, i.e., $\mathbf{w}(m)$ is mth column vector of $\mathbf{W}$, and $z_m = f(\mathbf{w}^{\mathrm{T}}(m)\mathbf{x})$ the nonlinear activation, then reconstruction of input $\mathbf{x}$ from $\mathbf{z}$ can be written as $\sum_{m=1}^{M} z_m\mathbf{w}(m) = \mathbf{W}^{\mathrm{T}}\mathbf{z}$. Similar to the derivation of Eq. (6), a best reconstruction objective (5) will yield the following algorithm:

$$\mathbf{w}_{t+1}(m) = \mathbf{w}_t(m) + \mu_t(\mathbf{e}_t^{\mathrm{T}}\mathbf{w}_t(m)\mathbf{x}g_m + z_m\mathbf{e}_t),\ m = 1,\dots,M, \tag{7}$$

where $\mathbf{e}_t = \mathbf{x} - \mathbf{W}_t^{\mathrm{T}}\mathbf{z}$, g_m is the derivative of z_m, and $g_m = f'(\mathbf{w}_t^{\mathrm{T}}(m)\mathbf{x})$.

The nonlinear approximative subspace algorithm in Eq. (7) can be regarded as a straightforward nonlinear generalization of Oja's PCA subspace rule [4]. In this algorithm, the reconstruction $\hat{\mathbf{x}}$ is linear with respect to the weight vectors $\mathbf{w}(m)$, but the combination coefficients (hidden activations) are nonlinear. Its advantages can be intuitively understood that nonlinear activations implicitly take higher-order statistical information into account and the neurons become more independent than in standard linear PCA networks after convergence [17,18]. In general, the nonlinear PCA-type algorithm in Eq. (7) yields something else than the standard PCA solution. However, especially for mild nonlinearities, the results still approach the respective PCA solutions. On the other hand, the converged weight vectors of different neurons are typically not exactly orthogonal, but not far from orthogonality. In practice, the nonlinear approximative subspace algorithm has some numerical advantages. For example, it is not sensitive to local minima and has better stability properties comparing the corresponding standard neural PCA algorithms as the odd nonlinearity function grows less than linearly [17,18].

2.4. Subspace pattern recognition method with autoencoders

In pattern recognition, the subspace pattern recognition method (SPRM) [1,4] can be directly set up on PCA. In SPRM, a pattern class is represented by a subspace spanned by a group of basis vectors, i.e., the orthogonal components obtained by PCA. Denote $\mathscr{L} = \mathscr{L}(\mathbf{u}_1,\dots,\mathbf{u}_m)$ a subspace spanned by a set of independent basis vectors $\mathbf{u}_1,\dots,\mathbf{u}_m$ in pattern space R^L. The basic operation to determine whether a vector $\mathbf{x}$ belongs to $\mathscr{L}$ is the projection of $\mathbf{x}$ on $\mathscr{L}$, which is given by $\hat{\mathbf{x}} = P\mathbf{x}$, where P is the projection matrix of $\mathscr{L}$. For an arbitrary $\mathbf{x}$, its distance to $\mathscr{L}$ can be defined as $\|\mathbf{x} - \hat{\mathbf{x}}\|$, which measures the efficiency of representing the data by the subspace.

Loosely speaking, a linear autoencoder with an $L \times M$ weight matrix $\mathbf{W}$ can be considered as a neural network counterpart of principal subspace for matching a data set. The network represents an input data $\mathbf{x}$ as $\hat{\mathbf{x}} = \mathbf{W}\mathbf{h}$ via the hidden units' activations $\mathbf{h} = \mathbf{W}^{\mathrm{T}}\mathbf{x}$, in the exactly same way as the principal subspace. In other words,

$\mathbf{W}\mathbf{W}^{\mathrm{T}}$ spans the same subspace as spanned by the principal eigenvectors of Σ under the best reconstruction objective discussed in this section. The autoencoder's output $\mathbf{W}\mathbf{W}^{\mathrm{T}}\mathbf{x}$ reconstructs input with corresponding squared reconstruction error $E = \|\mathbf{x} - \hat{\mathbf{x}}\|^2$, which measures how well the model describes the data. Based on the same intuition, in a nonlinear autoencoder, the network still represents an input $\mathbf{x}$ as $\hat{\mathbf{x}} = \mathbf{W}\mathbf{z}$ via the hidden units' activation $\mathbf{z}$, but now the activation $\mathbf{z} = f(\mathbf{W}^{\mathrm{T}}\mathbf{x})$ is a nonlinear function of $\mathbf{x}$, which makes a theoretical analysis difficult. Conceptually, we can consider the reconstruction error $\|\mathbf{x} - \hat{\mathbf{x}}\|^2 = \|\mathbf{x} - \mathbf{W}\mathbf{z}\|^2$ as a matching score for measuring the representation efficiency, which is a direct extension from linear autoencoder. Among a number of benefits of replacing linear autoencoder with nonlinear one, the numerical advantages as mentioned above is a main reason we apply it in our mixture model as local PCA type representations.

3. A mixture model of local principal component representation

3.1. Motivation of mixture of local PCA

PCA or subspace method provides a continuous distributed representation. A limitation of PCA is its reliance on second-order statistics and global linear transform. This has motivated various developments of nonlinear generalizations of PCA, which can be roughly divided into two categories. The first is global approach, typical examples of this type include principal curve analysis [7] and five layers autoencoders. The second is to follow the "divide-and-conquer" principle [10] to model nonlinear structure in the data by some mixtures of local principal component projections. Divide-and-conquer is a popular principle in statistical and machine learning areas for handling complex tasks, which usually realizes a global classification or regression by explicitly dividing the input space into a nested sequence of regions and by fitting simple surfaces within these regions. Following this philosophy, a mixture model of local PCA usually involves two procedures: partition a data set into a number of nonoverlapping regions and each region is represented not by its central point as in clustering but by a localized linear subspace. It is easy to understand that such a mixture model has no unique definition and its performances depend on global partition and local linear modeling.

3.2. 'Neural gas' algorithm for clustering

In data analysis, clustering techniques provide a nonlinear discrete representation, which use a number of local centers to represent input vectors. For a set of M reference vectors, $\{\mathbf{w}(1), \ldots, \mathbf{w}(M)\}$, an input vector $\mathbf{x}$ is considered being best matched by one of its reference vector $\mathbf{w}(k)$ in the sense that an appropriately defined distortion measure such as the squared Euclidean distance $\|\mathbf{x} - \mathbf{w}(k)\|^2$ is minimal. In other words, $\mathbf{x}$ is represented by the kth center such that the reconstructed vector $\hat{\mathbf{x}}$ is

$$\hat{\mathbf{x}} = \mathbf{w}(k) \quad \text{if } \|\mathbf{x} - \mathbf{w}(k)\| = \min_{l=1} \|\mathbf{x} - \mathbf{w}(l)\|. \tag{8}$$

The reference vectors partition the input space R^L into the so-called Voronoi polygons defined as

$$V_k = \{\mathbf{x} \in R^L | \ \|\mathbf{x} - \mathbf{w}(k)\| \leqslant \|\mathbf{x} - \mathbf{w}(l)\|, \forall l\}. \tag{9}$$

The problem can be described as a statistical optimization issue with the cost defined as

$$J = \int P(\mathbf{x})\,\mathrm{d}\mathbf{x} \sum_{k=1}^{M} p_k \|\mathbf{x} - \mathbf{w}(k)\|^2, \tag{10}$$

where $P(\mathbf{x})$ is the distribution of input data and p_k is a membership indicator variable. The traditional k-means clustering with p_k defined as a δ-function is the most straightforward way to minimize (8) via gradient descent on J, which bears some disadvantages such as local minima and underutilization of reference vectors, etc.

The 'neural gas' algorithm proposed in Ref. [13] is an efficient method for solving Eq. (10). Its idea is to partition the input space by competition and at the same time order the neurons based on a distance metric defined in the input space. In a neural gas model, reference vectors $\mathbf{w}(m)$, $m = 1, \ldots, M$, are associated with connection weights of neural units and adapted by the relative distances between the neural units within the input space. Each time an input $\mathbf{x}$ is presented, we first make an ordering of the elements of a set of distortions $E_{\mathbf{x}} = \{\|\mathbf{x} - \mathbf{w}(m)\|, m = 1, \ldots, M\}$ and then determine the adjustment of reference vector $\mathbf{w}(m)$. Specifically, for a given data vector $\mathbf{x}$, we determine the "neighborhood-ranking" $(E_{\mathbf{x}}(m_0), E_{\mathbf{x}}(m_1), \ldots, E_{\mathbf{x}}(m_{M-1}))$ of the distortion set, which means $\mathbf{w}(m_0)$ is closest to $\mathbf{x}$, $\mathbf{w}(m_1)$ second closest to $\mathbf{x}$, $\mathbf{w}(m_k)$, $k = 0, \ldots, M-1$ the reference vector for which there are k vectors $\mathbf{w}(j)$ with $\|\mathbf{x} - \mathbf{w}(j)\| < \|\mathbf{x} - \mathbf{w}(m_k)\|$. Then, each neuron adjusts its own weight via a dynamical learning rate which depends on the ranking of its representation capability. Denote the number k associated with each neural unit m by k_m. The following learning rule is the simple 'neural gas' algorithm in Ref. [13]:

$$\mathbf{w}_{t+1}(m) = \mathbf{w}_t(m) + \mu_t h_\lambda(k_m)(\mathbf{x} - \mathbf{w}_t(m)), \ m = 1, \ldots, M, \tag{11}$$

where t is the time scale and μ_t is the learning rate, $h_\lambda(k_m)$ is 1 for $k_m = 0$ and decays to zero for increasing k_m with characteristic decay constant.

The 'neural gas' algorithm has a number of advantages over other clustering or VQ algorithms, particularly, it has fast convergence and very small distortion errors [13].

3.3. Quality criterion of local PCA approximation

While PCA provides a global, linear transform of the data, clustering or VQ offers a local, nonlinear mapping between the data and the representation. In practice, these two basic forms of data representation can be combined in an appropriate way to establish some kind of nonlinear distributed representations. A mixture model of local principal component analysis is such as a combination, which partitions the data set into a number of K regions and each region V_k is represented by a respective M_k-dimensional linear subspace $\mathscr{L}^{(k)}$. In other words, each input vector is assigned to the most appropriate partition and then represented by the M_k basis vectors of the region. In linear case, this representation can be expressed as

$$\mathbf{y}^{(k)} = \mathbf{W}^{(k)^{\mathrm{T}}}\mathbf{x} \quad \text{if } \mathbf{x} \in V_k,\ k = 1,\dots,K, \tag{12}$$

where $\mathbf{W}^{(k)}$ is an $L \times M_k$ matrix whose columns are the M_k principal components of the partition V_k. The reconstructed vector $\hat{\mathbf{x}}^{(k)}$ is calculated as

$$\hat{\mathbf{x}}^{(k)} = \mathbf{W}^{(k)}\mathbf{y}^{(k)} \quad \text{if } \mathbf{x} \in V_k,\ k = 1,\dots,K. \tag{13}$$

The reconstruction error

$$\begin{aligned} E^{(k)} &= \|\mathbf{x} - \hat{\mathbf{x}}^{(k)}\|^2 \\ &= \|\mathbf{x} - \mathbf{W}^{(k)}\mathbf{W}^{(k)^{\mathrm{T}}}\mathbf{x}\|^2 \\ &= \|\mathbf{x} - P^{(k)}\mathbf{x}\|^2 \quad k = 1,\dots,K \end{aligned} \tag{14}$$

measures the distance between $\mathbf{x}$ and the subspace $\mathscr{L}^{(k)}$, where $P^{(k)}$ is the projection matrix of $\mathscr{L}^{(k)}$. The squared Euclidean distance $E^{(k)}$ to the linear manifold defined by local M_k-dimensional PCA in the kth local region can be termed as reconstruction distance. Consider a nonlinear model composed of K autoencoders as shown in Fig. 1, a similar representation can be expressed as

$$\mathbf{z}^{(k)} = f(\mathbf{W}^{(k)^{\mathrm{T}}}\mathbf{x}) \quad \text{if } \mathbf{x} \in V_k \tag{15}$$

where f is a sigmoidal nonlinearity, $\mathbf{W}^{(k)}$ is an $L \times M_k$ connection weight matrix belonging to the kth network whose column vectors span the same subspace as spanned by the M_k principal components from the partition V_k. Such a model can be considered as a "mixture of experts"paradigm [10] with 'gating' mechanism being realized by competition among experts' performance.

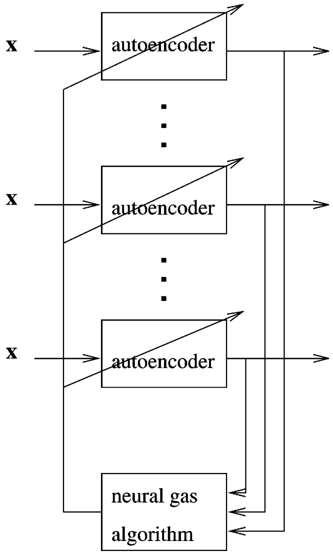

Fig. 1. Illustration of the training scheme for a mixture model of local PCA. In the figure, each autoencoder realizes a local principal subspace projection.

We denote mth column vector of $\mathbf{W}^{(k)}$ as $\mathbf{w}^{(k)}(m)$. The reconstruction vector $\hat{\mathbf{x}}^{(k)}$ is calculated as

$$\begin{aligned} \hat{\mathbf{x}}^{(k)} &= \mathbf{W}^{(k)}\mathbf{z}^{(k)} \\ &= \sum_{m=1}^{M_k} z_m^{(k)}\mathbf{w}^{(k)}(m) \quad \text{if } \mathbf{x} \in V_k. \end{aligned} \tag{16}$$

The corresponding reconstruction error is then

$$\begin{aligned} E^{(k)} &= \|\mathbf{x} - \hat{\mathbf{x}}^{(k)}\|^2 \\ &= \|\mathbf{x} - \mathbf{W}^{(k)}f(\mathbf{W}^{(k)^{\mathrm{T}}}\mathbf{x})\|^2 \quad k = 1,\dots,K. \end{aligned} \tag{17}$$

3.4. Learning algorithm

By the local PCA approximation quality criterion, Eq. (17), input space can then be partitioned by a competition among these PCA type representations on the basis of the reconstruction distances $E_{\mathbf{x}} = \{\|\mathbf{x} - \hat{\mathbf{x}}^{(k)}\|^2, k = 1,\dots,K\}$; K is the number of subspaces. Each time an input $\mathbf{x}$ is presented, we first make an ordering of the elements of $E_{\mathbf{x}}$ and then determine the adjustment of each subspace $\mathscr{L}^{(k)}$, $k = 1,\dots,K$. In other words, we make a ranking $(E_{\mathbf{x}}^{(k_0)}, E_{\mathbf{x}}^{(k_1)},\dots,E_{\mathbf{x}}^{(k_{K-1})})$ of the reconstruction error set, with $\hat{\mathbf{x}}^{(k_0)}$ being closest to $\mathbf{x}$, $\hat{\mathbf{x}}^{(k_1)}$ being second closest to $\mathbf{x}$, $\hat{\mathbf{x}}^{(k_l)}$, $l = 0,\dots,K-1$ being the reconstructed vector for which there are k_l vectors $\hat{\mathbf{x}}^{(j)}$ with $\|\mathbf{x} - \hat{\mathbf{x}}^{(j)}\| < \|\mathbf{x} - \hat{\mathbf{x}}^{(k_l)}\|$. Specifically, each network adjusts its weight matrix via a dynamical learning rate which depends on the ranking of its reconstruction error. Denote the number d associated with each subnetwork

k by d_k. The following learning rule generalizes the 'neural gas' algorithm in

$$\mathbf{w}_{t+1}^{(k)}(m) = \mathbf{w}_t^{(k)}(m) + \mu_t h_\lambda(d_k)(\mathbf{e}_t^{(k)^{\mathrm{T}}}\mathbf{w}_t^{(k)}(m)\mathbf{x}g_m + z_m^{(k)}\mathbf{e}_t^{(k)}),$$

$$m = 1,\dots,M_k, \quad k = 1,\dots,K \qquad (18)$$

where $\mathbf{e}^{(k)} = \mathbf{x} - \hat{\mathbf{x}}^{(k)}$ is the reconstruction error obtained from kth network, $h_\lambda(d_k)$ is 1 for $d_k = 0$ and decays to zero for increasing d_k. In the simulations we choose the dynamical adaptation step $h_\lambda(d_k) = \exp(-d_k/\lambda)$, with λ being a decay constant, which is same as in the original 'neural gas' algorithm [13]. By $h_\lambda(d_k)$, a data point $\mathbf{x}$ is assigned to a subspace $\mathscr{L}^{(k)}$ with a degree p_k:

$$p_k = \frac{h_\lambda(d_k)}{\sum_{l=0}^{M_k} h_\lambda(d_l)} \qquad (19)$$

which can be regarded as a membership of $\mathbf{x}$ to $\mathscr{L}^{(k)}$.

In summary, the learning process can be outlined as concurrently performing the following two steps:

1. For an input pattern, determine a winner autoencoder c in a mixture model, the subspace $\mathscr{L}^{(c)}$ of which is closest to input $\mathbf{x}$ based on the reconstruction distance (17) or other local PCA metric. Then perform local PCA learning algorithm, Eq. (18), separately for each submodel k with the adaptations being proportional to responsibilities p_k in Eq. (19). In other words, adjust the projection matrix of subspace $\mathscr{L}^{(k)}$ or its autoencoder counterpart, of submodel k, $k = 1,\dots,K$, according to their closeness to a given pattern $\mathbf{x}$.
2. Stop if the adaptations have converged, otherwise pick a new example and return to Step 1.

4. A classification scheme for handwritten digit recognition

4.1. Experimentation

In this section, we propose a modular classification system based on our mixture model to solve the handwritten digit recognition problem. Instead of training a single neural network classifier to output 10 class labels, we exploit 10 mixture models (called modules) to describe the 10 digit classes. During training, each model only accepts training examples of its own class. In this way, the computation can be considerably saved. For each class, a mixture model composed of a number of autoencoders is built. We apply the learning rule (18) in our experiments.

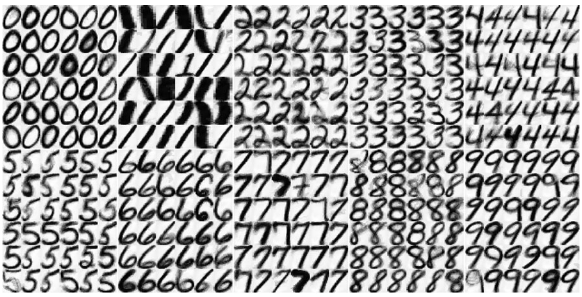

Fig. 2. The converged weight vectors in 10 mixture models, each with 36 submodels. Each submodel is an autoencoder with one hidden node. The weight vectors are visulized in mask forms after being equalized to 256 grey levels.

We exploited 20 000 digits from the segmented handwritten digit database produced by the US National Institute of Standards and Technology (NIST). 10,000 samples were used for training and another 10,000 samples from different forms for testing. The binary images have been scaled to a 25 × 20 pixel grid. In this paper, we directly use digit bitmaps without extracting any features. In a mixture model composed of K autoencoders, each autoencoder has $L = 500$ input units and M output units. In our experiments, we have compared different values of K and M. Learning is proceeded in three cycles with the training samples. The parameter μ in Eq. (18) is initially set to 1 and then dynamically decreases to 0.1. The decay constant λ in the dynamical adaptation step $h_\lambda(d_k)$ changes from 20 to 0.1. The time dependence for μ and λ is taken a same form as $g(t) = g_i(g_f/g_i)^{t/t_{max}}$ [11], in which t is the current adaptation step, t_{max} is a predefined maximum adaptation step, i.e., $t_{max} = 30{,}000$ in our experiments. The subscripts i and f stand for initial value and final value, respectively, i.e., $\mu_i = 1$, $\mu_f = 0.1$, $\lambda_i = 20$, $\lambda_f = 0.1$.

In a mixture model, the converged weight vectors in different autoencoders will be specialized for different styles. In Fig. 2, we illustrate the converged weight vectors resulted from an experiment, in which each mixture model or module has 36 submodels and each autoencoder has one hidden node with nonlinearity parameter $\beta = 0.1$. In other words, an autoencoder in a mixture model gives an one-dimensional subspace for describing a specialized style in that digit class. The weight vectors are visualized in mask forms after being equalized to 256 grey-level images, which are quite close to binary bitmaps.

After the training is completed, the modular classification system is as shown in Fig. 3, which recognizes an input digit by simply comparing all the reconstructed patterns calculated from the 10 modules. In Fig. 4, we schematically demonstrate the classification process, with more detailed technical issues expounded as follows.

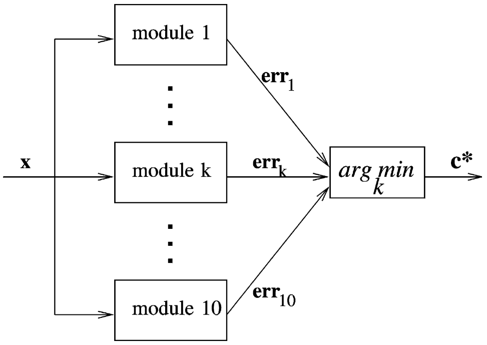

Fig. 3. Our proposed modular classification system based on the mixture models of local principal component analysis.

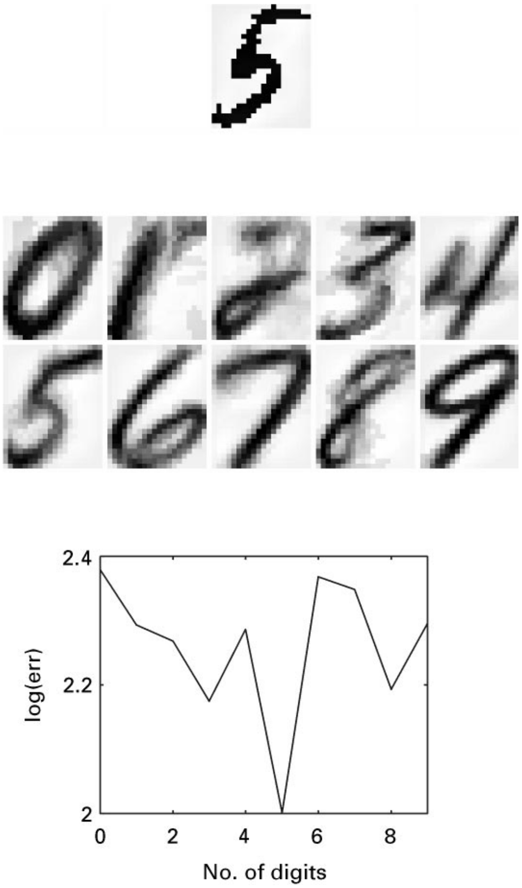

Fig. 4. Illustrations of a recognition process. Top figure is an input bitmap of digit '5'. Ten reconstructed patterns are then provided by the 10 modules via the Eq. (20), as shown in the middle. The bottom figure demonstrates the corresponding reconstruction errors according to Eq. (24), from which the class label of '5' can be identified.

4.2. Quality criteria

When a test image $\mathbf{x}$ is presented to all the 10 modules, an important problem is to define a matching score of a module. We consider two approaches. The first approach can be illustrated in Fig. 5. In a mixture model or module, we evaluate all the reconstruction errors from all the submodels (autoencoders) and use the smallest error as the measure of how well this mixture model matches the data, i.e.,

$$\hat{\mathbf{x}} = \hat{\mathbf{x}}_{k^*}, \quad k^* = \arg\min_k \|\mathbf{x} - \hat{\mathbf{x}}_k\| \tag{20}$$

In the second approach, we determine a reconstructed pattern from a mixture model by a population of the reconstructions $\hat{\mathbf{x}}^{(m)}$ by first specifying a response function a_k of the kth autoencoder. Our choice is a gaussian function and is as follows:

$$a_k = \exp\left(-\frac{\|\mathbf{x} - \hat{\mathbf{x}}_k\|^2}{2\sigma_k^2}\right), \quad k = 1, \dots, K, \tag{21}$$

where σ_k is the width of kth autoencoder's receptive field. Decoding of the semantics of the reconstruction response vectors $\mathbf{a} = [a_1, \dots, a_K]^{\mathrm{T}}$ can follow the center-of-gravity principle, i.e., an overall reconstruction $\hat{\mathbf{x}}$ associated to $\mathbf{a}$ is given as the activity-weighted average over all $\hat{\mathbf{x}}^{(k)}$, i.e.,

$$\hat{\mathbf{x}} = \frac{\sum_{k=1}^{K} a_k \hat{\mathbf{x}}_k}{\sum_{k=1}^{K} a_k}. \tag{22}$$

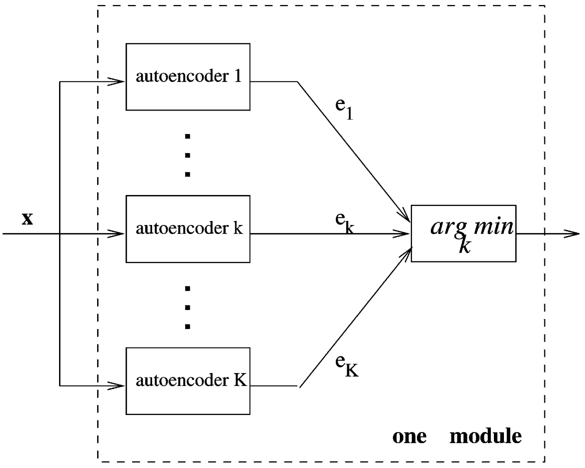

Fig. 5. The diagram of our first approach for obtaining a reconstructed pattern $\hat{\mathbf{x}}$ from a mixture model. In the figure, each autoencoder is used as a submodel for realizing local PCA type representation. $e_k = \|\mathbf{x} - \hat{\mathbf{x}}_k\|$ is the reconstruction error from kth submodel.

Table 1
Recognition accuracy based on mixture models with different sigmoidal β and different module size K, the number of submodels in each mixture. A submodel in this experiment is an autoencoder with one hidden node ($M = 1$). The classification is based on the second decoding scheme, Eq. (22), with $\sigma = 1$. Both training and testing data set has 10,000 samples

K	$\beta = 0.1$		$\beta = 0.5$		$\beta = 1$	
	Training set	Testing set	Training set	Testing set	Training set	Testing set
20	96.87%	95.50%	97.05%	95.40%	97.08%	95.42%
25	97.75%	96.04%	97.56%	95.93%	97.53%	95.91%
30	98.15%	96.55%	98.05%	96.13%	98.02%	96.42%
35	98.33%	97.15%	98.32%	97.23%	98.33%	96.85%
40	98.61%	97.38%	98.72%	97.8%	98.40%	97.31%

Based on the above discussion, for a test sample $\mathbf{x}$, the following reconstruction error

$$err = ||\mathbf{x} - \hat{\mathbf{x}}||^2 \tag{23}$$

can be used as a matching score of a mixture model.

Recognition process can then be simply proceeded by first presenting a test sample $\mathbf{x}$ to all the 10 mixture models and then comparing the reconstruction errors err_l, $l = 1, \ldots, 10$,

$$err_l = ||\mathbf{x} - \hat{\mathbf{x}}^{(l)}||^2, \; l = 1, \ldots, 10, \tag{24}$$

where l indicates the number of modules, and $\hat{\mathbf{x}}^{(l)}$ is calculated by Eq. (20) or Eq. (22). Classification is made by using a decision module which compares the reconstruction errors in Eq. (24) between the reconstructed pattern and presented pattern. In the following, we simply associate the class of a module with the smallest error, i.e., $\mathbf{x}$ is assigned to the class c^* if

$$c^* = \arg\min_c \; err_c. \tag{25}$$

4.3. Results

In the following, we give some quantitative experiment results on the performance of our modular classification scheme. In a first set of experiments, we compared the classification accuracies with different values of the sigmoidal nonlinearity β and different sizes of each mixture model, with main results shown in Table 1. In these experiments, the reconstruction from each mixture model is based on the second decoding scheme, Eq. (22), with the receptive field parameter σ taken as 1. From the results we can see that there is no significant differences among the performances of different β values. And as can be expected, the larger each mixture module, the more accurate the recognition result. However, as the number of autoencoders increase in each module, the learning will slow down and the improvement over the recognition will be marginal.

Table 2
Recognition accuracy for the mixture model by the first reconstruction scheme. In these experiments, $\beta = 0.5$. K is the size of a mixture model. A submodel in this experiment is an autoencoder with one hidden node ($M = 1$). Both training and testing data set have 10,000 samples

K	20	25	30	35	40
Training set	96.87%	97.56%	98.13%	98.29%	98.65%
Testing set	95.13%	95.64%	95.83%	96.12%	96.20%

In the second set of experiments, we compared the recognition differences between the two methods for reconstructing a test pattern from a mixture model. The experiment is proceeded in a similar way as in the last set of experiments, with $\beta = 0.5$, $M = 1$ and Eq. (20). Table 2 gives the classification results based on this simpler reconstruction scheme. Comparing the results with those in Table 1 for $\beta = 0.5$, we can see that the second reconstruction method Eq. (22), offers better classification accuracy. On the other hand, the first method, Eq. (20), is simpler, thus providing faster classification.

In a third set of experiments, we take a different receptive field parameter σ in the second reconstruction scheme, Eq. (22), for comparison purpose. From the classification results in Table 3 which compares $\sigma = 1$ and 3, we can find that there is no obvious differences. In practice, we take $1 \leqslant \sigma \leqslant 5$ as a relatively small value.

We also conducted some other comparisons. In the above experiments, the autoencoders in each mixture module have only one hidden node ($M = 1$). The classification accuracy could be improved by increasing the number of hidden nodes in each autoencoder, i.e., increasing the dimension of local PCA from 1 dimension to 2 or higher.

Above experiments give us the classification accuracy or raw recognition rate without any rejection. In both training and testing set, there are some digit samples with great variances in shapes, thickness, etc., which are

Table 3
Recognition accuracy for the mixture model with different β and σ. Each mixture module has 36 units. Both training and testing data set has 10,000 samples

β	0.1		0.5		1	
ρ	1	3	1	3	1	3
Training set	98.57%	98.37%	98.68%	98.64%	98.69%	98.68%
Testing set	97.21%	97.18%	97.27%	97.29%	97.42%	97.61%

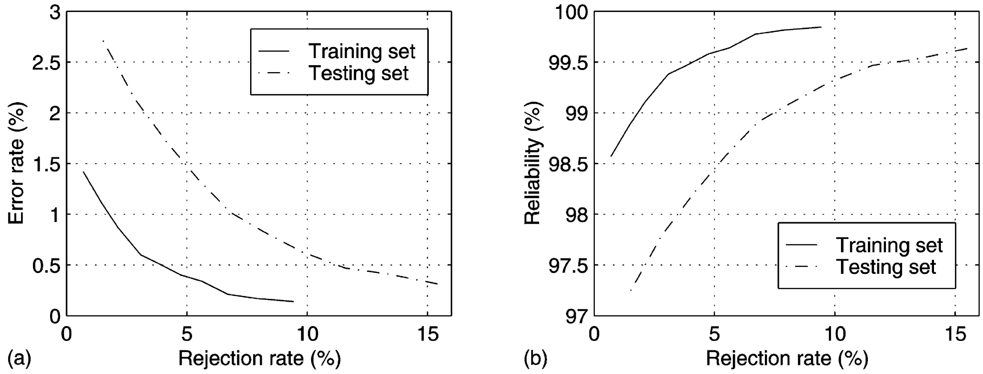

Fig. 6. (a) Relationship between the error rate and the rejection rate, and (b) relationship between the recognition rate and the rejection rate.

harder to be correctly classified. In practice, a very small error rate is often required. When a recognition system is established, error rate can be lowered by rejecting some test patterns. In our modular classification system, a test pattern can be rejected if the smallest reconstruction error and the second smallest error differs by less than a threshold. Specifically, we define an indicator variable η as

$$\eta = 1 - \frac{err_j}{err_i} \tag{26}$$

where err_j and err_i are the smallest and second smallest reconstruction errors, respectively. A decision is made by the following rule:

$$\begin{aligned}&\mathbf{x} \text{ is rejected from classification if } \eta > \eta_T\\ &\mathbf{x} \text{ is accepted for classification if } \eta \leqslant \eta_T\end{aligned} \tag{27}$$

where η_T is a threshold which can be experimentally decided. Usually, error rate will be lowered by increasing the threshold η_T and a larger η_T means a higher rejection rate. We undertook an experiment by varing the threshold η_T from 0.01 to 0.1. Fig. 6(a) shows the error rate versus the rejection rate which result from changing η_T. In this experiment, the error rate and rejection rate are defined as follows:

$$\text{Error rate} = \frac{\text{number of misrecognized test patterns}}{\text{total number of test patterns}}, \tag{28}$$

$$\text{Rejection rate} = \frac{\text{number of rejected patterns}}{\text{total number of test patterns}}. \tag{29}$$

From Fig. 6(a) we see that the lowest error rate is less than 1% with rejection rate of 7% on the testing data set. Reducing the rejection rate will cause the increasing on error rate. In addition to error rate and rejection rate, another index for evaluating a handwritten digit recognition system is reliability, which refers to the proportion of correctly recognized patterns in all the test patterns. The relationship between reliability and error rate can be written as

$$\text{Reliability} = \frac{\text{recognition rate}}{\text{recognition rate} + \text{error rate}} \times 100. \tag{30}$$

The reliability from the same experiment as in Fig. 6(a) is shown in Fig. 6(b), which shows again the satisfactory result.

5. Discussion and conclusion

An appropriate mixture model of local PCA has been proven to constitute a significant alternative to the standard global PCA. The problem of constructing a mixture model can be decomposed into two distinct procedures: first partitioning the data space and then estimating the principal subspace within each partition. In this paper we proposed an efficient mixture model which creates a set of statistical representations pertinent to different aspects of input. By 'neural-gas'-based soft competitive learning, each representation selectively focuses on a different subclass of the data. By utilizing a nonlinear autoencoder learning algorithm which is based on the least-squares reconstruction principle, the local PCA type representation is robust to noise or outliers.

A major motivation of studying mixture of local PCA is solving some difficult pattern recognition problems. Regarding handwritten digit images, a high-quality classifier should discover the data-generative mechanism or model the images manifolds [9]. From this viewpoint, Hinton et al. were the first to use some mixture models of PCA or factor analysis (FA) to recognize handwritten digits [9]. Our recognition scheme is closely related to their work. However, some important differences exist. Instead of applying the EM algorithm to calculate the responsibility of a module for reconstructing a test pattern, which requires introducing a variance parameter whose value is often arbitrarily chosen, we use the 'neural gas' algorithm to directly perform clustering. 'Neural gas' algorithm dynamically adjusts the adaptation step in the clustering process, which has been proven outperforming a number of other clustering or VQ algorithms, especially in its fast convergence and low distortion error. In Ref. [9], each digit's manifold is modelled by a number of linear autoencoders which approximately perform linear subspace projections. In our method, linear PCA is replaced by a symmetrical nonlinear autoencoder for describing the local statistical structure. A last difference lies in the classification criterion. In our method, we define an averaged "distance" between a test digit and a number of reconstructed patterns from different submodels for describing the same class, whereas in Ref. [9], classification is directly set up on the reconstruction errors given by the autoencoders.

We have shown that an appropriately constructed mixture model of handwritten digits bitmaps can classify digits quite well. In a mixture model, an autoencoder can capture the local structure of digits bitmaps. Different autoencoders in a mixture complement each other in capturing different styles of digits and sharing their responsibilities for reconstructing or explaining a test digit. Comparing some other recently proposed methods such as in Ref. [24] which offer about 3% error rate on the original data (bitmaps with mean size of 45×60 pixels), our modular classification system based on the mixture models demonstrates a much improved recognition accuracy, with much smaller bitmaps size. With moderate module size, the classification accuracy reaches 98.6% on the training set and closes to 97.8% on the testing set, without any rejection. The lowest error rate is less than 1% with rejection rate of 7% on testing data set.

References

[1] S. Watanable, N. Pakvasa, Subspace method in pattern recognition, Proceedings of the International Joint Conference on Pattern Recognition, 1973, pp. 25–32.

[2] E. Oja, Simplified neuron model as a principal component analyzer, J. Math. Biol. 15 (1982) 267–273.

[3] E. Oja, Subspace Methods of Pattern Recognition, Research Studies Press, Letchworth, UK, 1983.

[4] E. Oja, Neural networks, principal components and subspace, Int. J. Neural System 1 (1989) 61–68.

[5] E. Oja, Principal components, minor components, and linear neural networks, Neural Networks 5 (1992) 927–935.

[6] T.D. Sanger, Optimal unsupervised learning in a single-layer linear feedforward neural network, Neural Networks 2 (1989) 459–473.

[7] T. Hastie, W. Stuetzle, Principal curves, J. Am. Stat. Assoc. 84 (1989) 502–516.

[8] R.D. Dony, S. Haykin, Image segmentation using a mixture of principal components representation, IEE Proc.-Visual Image Signal Process 144 (1997) 73–80.

[9] G.E. Hinton, P. Dayan, M. Revow, Modelling the manifolds of images of handwritten digits, IEEE Trans. Neural Networks 8 (1997) 65–74.

[10] R.A. Jacobs, M.I. Jordan, S.J. Nowlan, G.E. Hinton, Adaptive mixtures of local experts, Neural Comput. 3 (1991) 79–87.

[11] T. Kohonen, Self-organized formation of various invariant-feature filters in the adaptive subspace SOM, Neural Comput. 9 (1997) 1321–1344.

[12] N. Kambhatla, T.K. Lee, Dimension reduction by local principal component analysis, Neural Comput. 9 (1997) 1493–1516.

[13] T.M. Martinetz, S.G. Berkovich, K.J. Schulten, Neural-gas network for vector quantization and its application to time-series prediction, IEEE Trans. Neural Networks 4 (1993) 558–568.

[14] J.L. Marroquin, F. Girosi, Some extensions of the K-means algorithm for image segmentation and pattern classification, MIT Technical Report, AI Memo No. 11390, 1994.

[15] K. Rose, E. Gurewitz, G.C. Fox, A deterministic annealing approach to clustering, Phys. Rev. Lett. 11 (1990) 589–594.

[16] L. Xu, Least mean square error reconstruction principle for self-organization, Neural Networks 6 (1993) 627–648.

[17] J. Karhunen, J. Joutsensalo, Representation and separation of signals using nonlinear PCA type learning, Neural Networks 7 (1994) 113–127.

[18] J. Karhunen, J. Joutsensalo, Generalizations of principal component analysis, optimization problems, and neural networks, Neural Networks 8 (1995) 549–562.

[19] C.Y. Suen, C. Nadal, R. Legault, T.A. Mai, L. Lam, Computer recognition of unconstrained handwritten numerals, Proc. IEEE 80 (1992) 1162–1180.

[20] Y.L. Cun et al., Constrained neural network for unconstrained handwritten digit recognition, Proceedings of the First International Workshop on Frontiers in Handwritting Recognition, Montreal, Canada, 1990, pp. 145–154.

[21] Sung-Bae Cho, Neural network classifiers for recognizing totally unconstrained handwritten numerals, IEEE Trans. Neural Networks 8 (1997) 43–53.

[22] H. Schwenk, M. Milgram, Transformation invariant autoassociation with application to handwritten character recognition, in: G. Tesauro, D. Touretzky, T. Lee (Eds.), NIPS 7, Morgan Kaufmann, Los Altos, CA, 1995, pp. 991–998.

[23] R. Anand et al., Efficient classification for multiclass problems using modular neural networks, IEEE Trans. Neural Networks 6 (1995) 117–124.

[24] M. Revow, C.K.I. Williams, G.E. Hinton, Using generative models for handwritten digit recognition, IEEE Trans. Pattern Anal. Machine Intell. 18 (1996) 592–606.

About the Author—BAI-LING ZHANG was born in China. He received the Bachelor of Engineering degree in Electrical Engineering from Wuhan Institute of Geodesy, Photogrammetry and Chartography, in 1982, and Master of Engineering degree in Electronic System from the South China University of Technology, Ph.D. degree in Electrical and Computer Engineering from the University of Newcastle, Australia, in 1987 and 1999, respectively. Currently he works as a postdoctoral fellow in the University of Sydney, Australia. His research interest includes artificial neural networks, image processing, pattern recognition, and time-series analysis and prediction.

About the Author—MINYUE FU received his Bachelors degree in electrical engineering from the China University of Science and Technology, Hefei, China, in 1982, and M.S. and Ph.D. degrees in electrical engineering from the University of Wisconsin-Madison in 1983 and 1987, respectively.

From 1983 to 1987, he held a teaching assistantship and a research assistantship at the University of Wisconsin-Madison. He worked as a Computer Engineering Consultant at Nicolet Instruments, Inc., Madison, Wisconsin, during 1987. He joined the Department of Electrical and Computer Engineering, the University of Newcastle, Australia, in 1989. Currently, he is an Associate Professor and Head of the Department.

His main research interests include control systems and signal processing. He has been an Associate Editor of the IEEE Transactions on Automatic Control, and an Associate Editor for the Conference Editor Board of the IEEE Control Systems Society. He is currently an Associate Editor for the journal Optimization and Engineering.

About the Author—HONG YAN received his B.E. degree from Nanking Institute of Posts and Telecommunications in 1982, M.S.E. degree from the University of Michigan in 1984, and Ph.D. degree from Yale University in 1989, all in electrical engineering. From 1986 to 1989 he was a research scientist at General Network Corporation, New Haven, CT, USA, where he worked on developing a CAD system for optimizing telecommunication systems. Since 1989 he has been with the University of Sydney where he is currently a Professor in Electrical and Information Engineering. His research interests include computer animation, signal and image processing, pattern recognition, neural and fuzzy algorithms and quantum computers. He is an author or co-author of one book and has over 200 technical papers in these areas. Dr. Yan is a fellow of the Institution of Engineers, Australia (IEAust), a senior member of the IEEE, and a member of the SPIE, the International Neural Network Society, the Pattern Recognition Society, and the International Society for Magnetic Resonance in Medicine.