Neural Processing Letters **8:** 241–251, 1998.

Handwritten Digit Recognition by a Mixture of Local Principal Component Analysis

BAILING ZHANG[1], MINYUE FU[1] and HONG YAN[2]
[1] *Department of Electrical and Computer Engineering, The University of Newcastle, NSW 2308, Australia;* [2] *Department of Electrical Engineering, University of Sydney, NSW 2006, Australia, E-mail: bailing@ee.usyd.edu.au*

Abstract. Mixture of local principal component analysis (PCA) has attracted attention due to a number of benefits over global PCA. The performance of a mixture model usually depends on the data partition and local linear fitting. In this paper, we propose a mixture model which has the properties of optimal data partition and robust local fitting. Data partition is realized by a soft competition algorithm called neural 'gas' and robust local linear fitting is approached by a nonlinear extension of PCA learning algorithm. Based on this mixture model, we describe a modular classification scheme for handwritten digit recognition, in which each module or network models the manifold of one of ten digit classes. Experiments demonstrate a very high recognition rate.

Key words: neural networks, mixture of principal component analysis, handwritten digit recognition

1. Introduction

Principal component analysis (PCA) is a general purpose tool in pattern recognition for extracting average features that reflect global statistical properties of the pattern space. PCA essentially performs singular value decomposition (SVD) of data covariance matrix Σ and projects data onto the subspace spanned by those eigenvectors corresponding to the largest eigenvalues of Σ. This transformation decorrelates the signal components and the projection minimizes an average squared residual between the original signal and its dimension-reduced approximation. The well-known subspace pattern recognition method [1–2] is set up on PCA, in which each pattern class is represented by a subspace spanned by a group of basis vectors. 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.

Despite some useful properties of PCA, its applications are limited by its reliance on second-order statistics and linear projection. Geometrically, PCA models the data as a hyperplane embedded in the data space. This has motivated various developments of nonlinear PCA, for example, principal curve analysis, independent component analysis, etc. Among these techniques, appropriate mixture models of local PCA have some attractivenesses [3–6, 8–10], which 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.

Intuitively, mixture of local PCA has no unique definition and the performances of such a model depend on global partition and local linear fitting. In this article, we propose a mixture model which has the following two features. First, the data manifold is described by an optimal partition algorithm called neural 'gas' [13]; Second, the local principal component analysis is robust to outliers due to introducing sigmoidal nonlinearities to the projections [12].

We illustrate the performance of our proposed mixture model by applying it to handwritten digit recognition problem. Handwritten digits recognition is a typical multiclass classification problem, where each pattern belongs to exactly one of K ($K = 10$) classes. Many previous works including neural network methods utilize a single classifier or network. A better alternative is to apply an appropriate modularity method. In a modular classification system, modules learn to specialize in different subtasks, for example, a modular architecture proposed in [16] reduces a K-class problem to a set of K two-class problems. In this paper, we present a modular classification scheme from a different perspective, in which each module is a single layer network for modeling the manifolds of images of one of the ten classes. Each unit in a network corresponds to a subspace in the respective class and the network is trained only by the digits belonging to the class. During classification, upon presenting a test pattern, each module provides an individual reconstruction according to a population decoding principle and the overall decision is determined by comparing all of the ten reconstruction errors. Our modular classification system shares certain similarities with some previously proposed decision-based networks [9]. The most famous one is Kohonen's Learning Vector Quantization (LVQ) algorithm [7], which finds a set of cluster centers for each class and the classification is performed by first finding the closest center and then assigning the associated class. In a similar way, our approach finds a set of subspaces for each class and the classification is performed by first finding the closest (averaged) reconstruction from subspace projections and then assigning the associated class. Our results show that the mixture model is very effective in the recognition accuracy.

2. Learning Algorithm for Principal Components Analysis

Given an input data sample $\mathbf{x}$, weight vector $\mathbf{w}$, a linear neuron's output can be written as $y = \mathbf{w}^{\mathrm{T}}\mathbf{x}$. The largest principal component optimizes the reconstruction mean-square-error by choosing the best one dimensional subspace to project the input vector onto, i.e., the following objective:

$$\begin{aligned} \text{minimize } J &= E\left\{||\mathbf{x} - \hat{\mathbf{x}}||^2\right\} \\ &= E\left\{||\mathbf{x} - y\mathbf{w}||^2\right\} \end{aligned} \tag{1}$$

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

$$\mathbf{w}_{t+1} = \mathbf{w}_t + \mu_t(y\mathbf{e}_t + \mathbf{e}_t^T\mathbf{w}_t\mathbf{x}), \tag{2}$$

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, which can be developed from the objective (1) and the following consideration: the second largest principal component also statisfies the minimal reconstruction property with restriction that the second principal component direction must be orthogonal to the first component direction, and so on for the remaining principal component directions.

The extension of (1) to a nonlinear unit has been proposed in many papers [11–12]. With a nonlinear neuron $z = f(\mathbf{w}^T\mathbf{x})$, the learning objective is

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

In this paper, we take f as a sigmoidal function bounded between 0 and 1. Accordingly, the learning rule is

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

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

Algorithm (4) has the following properties [12]. First, the weight vector converges towards the true nonnormalized eigenvectors of Σ, even though a sigmoidal nonlinearity is used; Second, learning is robust to noise or outliers due to nonlinear neuron's selectiveness; Third, though input patterns $\mathbf{x}$ are still represented in a linear basis, the coefficients of the expansion are generally nonlinear.

3. Mixture Model of Principal Component Representation

A global M principal components representation of an L-dimensional data $\mathbf{x}$ can be simply written as

$$\mathbf{y} = \mathbf{W}^T\mathbf{x}, \tag{5}$$

where $\mathbf{W}$ is an $L \times M$ matrix whose ith column is the ith principal eigenvector of Σ. The corresponding reconstruction

$$\hat{\mathbf{x}} = \mathbf{W}\mathbf{y} \tag{6}$$

will result in minimal reconstruction error in least mean square sense. Such a global representation of data constitutes the basis of subspace methods of pattern recognition, in which a class is represented by a linear subspace within the original pattern space R^L. A set of m linearly independent basis vectors $\mathbf{u}_1, \ldots, \mathbf{u}_m$ in R^L spans a subspace $\mathcal{L}$, $\mathcal{L} = \mathcal{L}(\mathbf{u}_1, \ldots, \mathbf{u}_m)$. The basic operation to determine whether a vector $\mathbf{x}$ belongs to $\mathcal{L}$ is the projection of $\mathbf{x}$ on $\mathcal{L}$, which is given by $\hat{\mathbf{x}} = P\mathbf{x}$, where P is the projection matrix of $\mathcal{L}$. For an arbitrary $\mathbf{x}$, its distance to $\mathcal{L}$ can be defined as $||\mathbf{x} - \hat{\mathbf{x}}||$, which measures the efficiency of representing the data by the subspace.

Being different with PCA or subspace methods, in data analysis, clustering or VQ techniques provide discrete representations, which use one of a number of Voroni centers for each input vector. For a set of M centers, $\{\mathbf{w}_1, \ldots, \mathbf{w}_M\}$, an input vector $\mathbf{x}$ is represented by the k-th center such that the reconstructed vector $\hat{\mathbf{x}}$ is

$$\hat{\mathbf{x}} = \mathbf{w}_k, \text{ if } ||\mathbf{x} - \mathbf{w}_k|| = \min_{l=1} ||\mathbf{x} - \mathbf{w}_l||. \tag{7}$$

While the global PCA representation is a linear transform of the data, the representation under vector quantization is highly nonlinear function of the input vector. Between these two representations, some kind of nonlinear distributed representations are more desirable [3–6], which combine the advantages of the both PCA and VQ. Intuitively, we can introduce a mixture of local principal component transform, which partitions the data set into a number of regions and each region is represented by a respective M_k-dimensional linear subspace. 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. This representation can be expressed as

$$\mathbf{z} = f\left(\mathbf{W}_k^T \mathbf{x}\right), \text{ if } \mathbf{x} \in C_k, \tag{8}$$

where f is a sigmoidal nonlinearity, $\mathbf{W}_k$ is an $L \times M_k$ matrix whose columns are the M_k principal components of the partition C_k. In the following, we denote mth column vector of $\mathbf{W}_k$ as $\mathbf{w}^{(k)}$ (m). The reconstruction vector $\hat{\mathbf{x}}_k$ is calculated as

$$\begin{aligned} \hat{\mathbf{x}}_k &= \mathbf{W}_k \mathbf{z} \\ &= \sum_{m=1}^{M_k} \mathbf{w}^{(k)}(m) z_m, \quad \text{if } \mathbf{x} \in C_k. \end{aligned} \tag{9}$$

As there are many algorithms for realizing PCA, clustering of VQ can be also approached by numerious ways. Here we prefer to an efficient learning algorithm called neural 'gas' [13], which can be briefly outlined as follows.

Our startpoint is to partition the input space by competition and at the same time record the relationships among the local subspaces based on a distance metric defined in the input space. In a simple way, the metric can be the reconstruction

error provided by a subspace. The information about the relationship among the subspaces is provided by a set of reconstruction errors $E_x = \{||\mathbf{x} - \hat{\mathbf{x}}_m||^2,\ m = 1, \cdots, M_k\}$, M_k is the number of subspaces in kth partition. Each time an input $\mathbf{x}$ is presented, we first make an ordering of the elements of E_x and then determine the adjustment of mth subspace $\mathbf{w}^{(k)}(m)$ within kth region, $m = 1, \cdots, M_k,\ k = 1, \cdots, K$. In other words, we make a ranking $\left(E_x(m_0), E_x(m_1), \cdots, E_x(m_{M_k-1})\right)$ of the reconstruction error set, with $\hat{\mathbf{x}}_{m_0}$ being closest to $\mathbf{x}$, $\hat{\mathbf{x}}_{m_1}$ being second closest to $\mathbf{x}$, $\hat{\mathbf{x}}_{m_k},\ k = 0, \cdots, M_k - 1$, being the reconstructed vector for which there are m_k vectors $\hat{\mathbf{x}}_j$ with $||\mathbf{x} - \hat{\mathbf{x}}_j|| < ||\mathbf{x} - \hat{\mathbf{x}}_{m_k}||$. Specifically, each neuron adjusts its weight via a dynamical learning rate which depends on the ranking of its reconstruction error. Denote the number d associated with each neural unit m by d_m. The following learning rule generalizes the neural 'gas' algorithm in [13].

$$\begin{aligned} \Delta \mathbf{w}_t^{(k)}(m) &= \mu_t h_\lambda(d_m) \left(\mathbf{e}_t^T \mathbf{w}_t^{(k)}(m) \mathbf{x} g_m + z_m \mathbf{e}_t\right) \\ m &= 1, \cdots, M_k \\ k &= 1, \cdots, K \end{aligned} \tag{10}$$

where $h_\lambda(d_m)$ is 1 for $d_m = 0$ and decays to zero for increasing d_m. In the simulation we choose the same one as in [13], $h_\lambda(d_m) = \exp(-d_m/\lambda)$, with λ being a decay constant. In Equation (10), all the quantities are specific to kth partition.

In this paper, we do not discuss the topological relationships among the subspaces, which can be described by a dynamically adapted graph called Delaunary triangular (DT) [13]. Using a connectivity matrix to represent the graph structure with nonnegative elements, the adaptation of the connections can be executed in the exactly same way as in the 'gas' algorithm.

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

1. Specify a network among a number of candidates according to the given class label, which is the same in spirit as the strategy in LVQ [7]. This is the basis of our modular classification scheme which will be further expounded in detail in next section.

2. Determine a winner unit c in the specified network, the subspace $\mathcal{L}^{(c)}$ of which is closest to the pattern space of input $\mathbf{x}$. Note a unit in each network can be replaced by a subnetwork which represent a subspace with more than one dimension.

3. Adjust the projection matrix of subspace $\mathcal{L}^{(m)}$ of unit m, $m = 1, \cdots, M_k$, according to their closeness to a given pattern $\mathbf{x}$. In one-dimensional subspace situation, the algorithm is equation (10). Extension to more than one-dimensional subspace is straight-forward.

After the learning completes, a critical problem is to define a 'distance' between an input $\mathbf{x}$ and a number of subspaces which describe the same class, as there is no overall projection matrix as in the original subspace method. Based on a biologically inspired concept of population decoding [14], we proposed in section 4 a generalized distance and the corresponding classification scheme.

4. Handwritten Digit Recognition Based on Generalized Subspace Pattern Classification

In recent years, neural networks have been often applied to handwritten digit recognition. In most of previous works, a neural network model is mainly used as a classifier which is trained to output one of the ten classes. The shortcomings of such a paradigm have been pointed out in [5]. Another 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. Any network can be defined as an autoencoder provided that it has a meaningful internal representation which can be used to appropriately reconstruct input. In this sense, our proposed mixture of local principal component analysis is an efficient autoencoder model which can be used to construct a modular classification system for recognizing handwritten digits. Specifically, an individual network is trained on examples of each digit class and then a digit bitmap is classified by deciding which net offers the best reconstruction of the data.

We exploited 20000 digits from the segmented handwritten digit database produced by the U.S. National Institute of Standards and Technology (NIST). Some digit bitmaps are illustrated in Figure 1. 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. Each network has $L = 500$ input units and M output units. Learning is proceeded in three cycles with the training samples. The parameter μ in Equation (1) is initially set to 1 and then dynamically decreases to 0.1. The deacy constant λ in Equation (10) changes from 20 to 0.1.

The trained networks can then be directly used as classifiers, with scheme shown in Figure 2. In this architecture, each module is a bidirectional single layer network, as shown in Figure 3. In each network, the local principal component projections provide a population of representations and an overall reconstruction can be estimated based on the principle of population decoding [14]. In each network, we regard mth reconstruction $\hat{\mathbf{x}}_m$ from z_m as a partial description of input $\mathbf{x}$ and a complete reconstruction $\hat{\mathbf{x}}$ is the center of gravity of $\hat{\mathbf{x}}_k, k = 1, \cdots, M$. Specifically, the reconstruction vector $\hat{\mathbf{x}}$ for each network can be expressed as

$$\hat{\mathbf{x}} = \frac{\Sigma_k a_k \hat{\mathbf{x}}_k}{\Sigma_k a_k}, \tag{11}$$

where a_k is an interpolating function. A simple form is the following Gaussian kernel function, i.e.,

$$a_k = \exp\left(-\frac{||\mathbf{x} - \hat{\mathbf{x}}_k||^2}{2\rho_k^2}\right), \tag{12}$$

where ρ_k is the width of kth unit's receptive field, which should be chosen relatively sharp [14]. When a test sample $\mathbf{x}$ is presented to all the ten networks, reconstruction errors $err_l, l = 1, \cdots, 10$, are obtained,

$$err_l = ||\mathbf{x} - \hat{\mathbf{x}}^{(l)}||^2, \ l = 1, \cdots, 10 \tag{13}$$

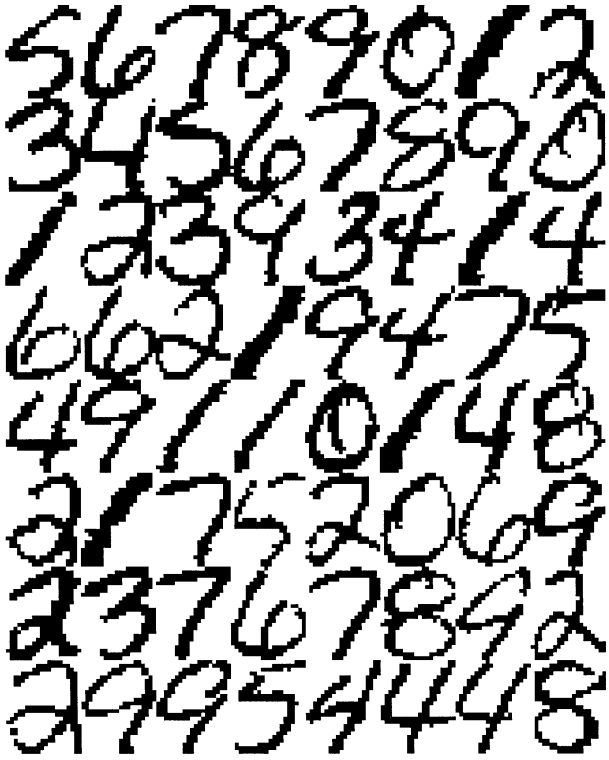

Figure 1. Some typical examples of handwritten digits.

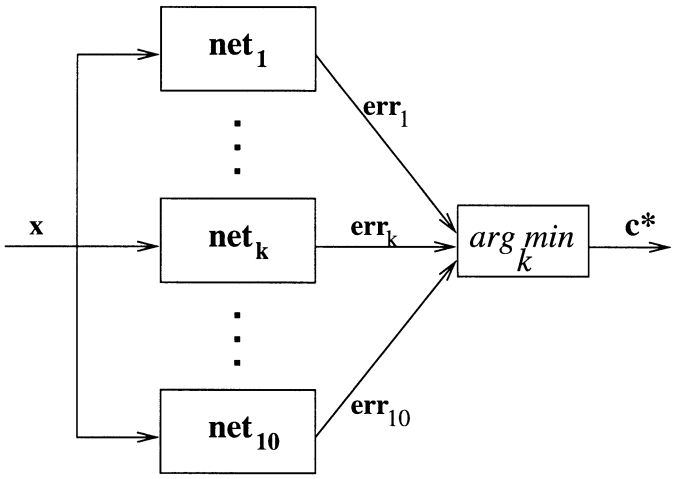

Figure 2. Our proposed modular classification system based on the mixture models of local principal component analysis.

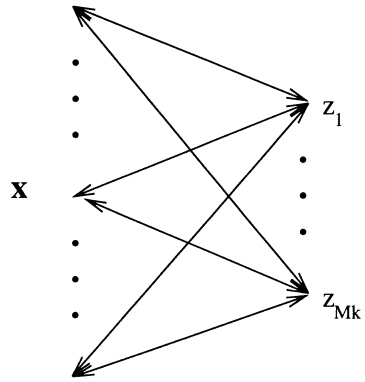

Figure 3. One module net_k in the Figure 2. Each unit represents a 1D subspace for describing the corresponding subclass.

where l indicates the number of network. Clearly, the squared reconstruction error is a measure of how well a module fits a digit's bitmap, thus can be considered as an extension of the 'distance' in traditional subspace method. We build a classifier by using a decision module which compares the distances or reconstruction errors in Equation (13) between the reconstructed vectors and presented pattern. We can simply associate the class of a model with the smallest error, i.e., **x** is assigned to the class c^* if

$$c^* = \arg\min_c err_c \tag{14}$$

For comparison purpose, we experimented with different nonlinearities β and different receptive field parameter ρ. In Figure 4, we demonstrate the converged weight vectors in ten networks, each with 25 units. The main recognition results are shown in Table 1 and 2. A rough guideline for choosing width ρ_k is $\rho_k < \sqrt{M}/3$, i.e., the sharping tuning kernel functions. The effect of nonlinearity β on the recognition rate is not obvious. We also compared different size of each network. As can be expected, the larger the network, the more accurate the recognition result. However, as the number of nodes increase in each network, the learning will slow down and the improvement over the recognition will be small. In Figure 5 we illustrate some typical digits that have not been recognized.

The minimum operator in Equation (14) is the simplest hard decision rule. A more reasonable replacement is using a fuzzy decision. Another further improvement of our recognition scheme is to incorporate classification into learning phase, as discussed in [15]. Pragmatically, we can also combine different kinds of efficient classifiers to vote for a final decision.

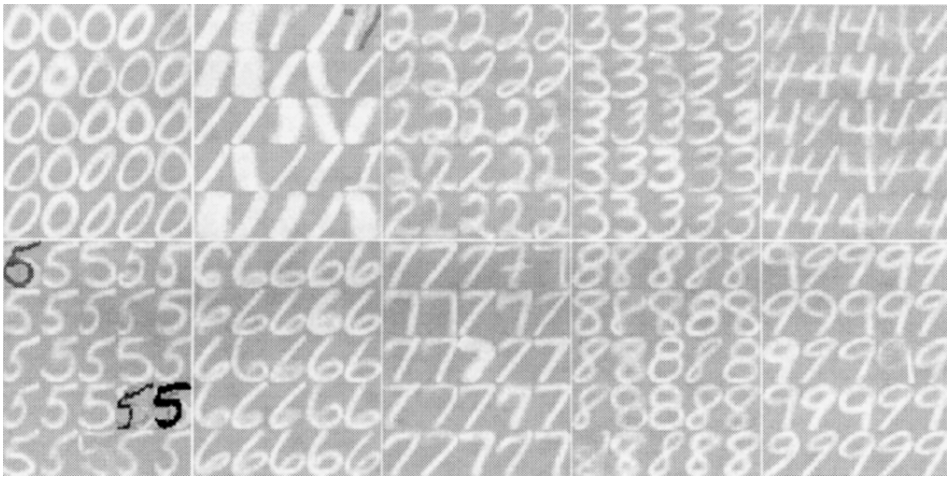

Figure 4. The converged weight vectors in ten networks, each with 25 units.

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

β	0.5		1	
ρ	1	2	1	2
training set	98.68%	98.64%	98.59%	98.68%
testing set	96.07%	96.99%	96.46%	97.1%

5. Discussion and Conclusion

An appropriate mixture of local PCA constitutes a significant alternative to the standard PCA. In this paper we proposed an efficient mixture model which creates a set of statistical representations pertinent to different aspects of input. By neural

Table II. Recognition accuracy for the mixture model with different β and ρ. Each network has 25 units. Both training and testing data has 10,000 samples.

β	0.5		1	
ρ	1	2	1	2
training set	97.13%	97.66%	98.03%	97.83%
testing set	96.83%	96.41%	96.30%	96.57%

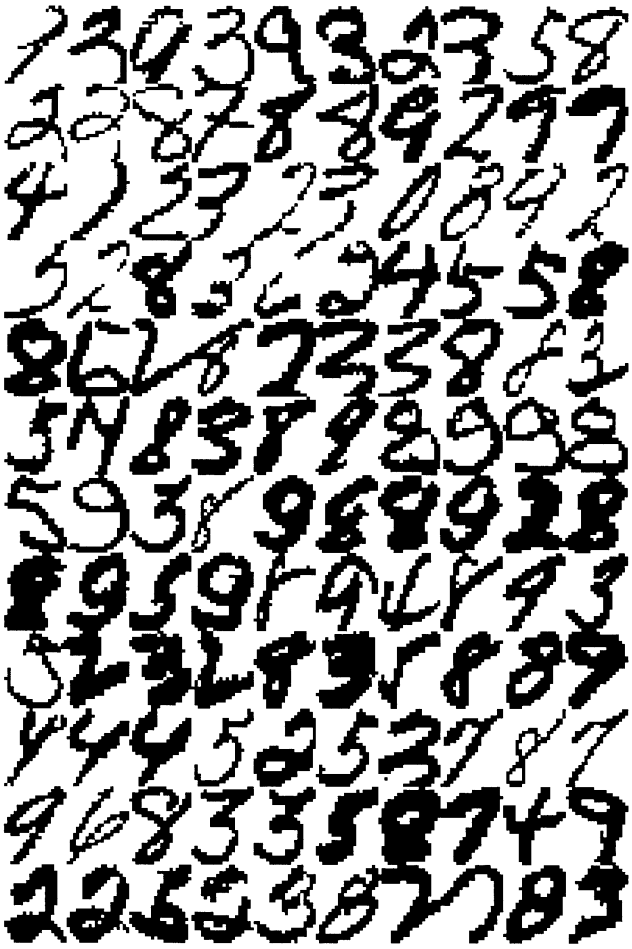

Figure 5. Some digits in the training set that have not been recognized. Some of them appear many times.

'gas' based soft competitive learning, each representation selectively focus on a different subclass. By a nonlinear extension of PCA algorithm, the low dimensional subspace projection is robust.

A major motivation of studying mixture of local PCA is solving some difficult pattern recognition problem. Regarding handwritten digit images, a high quality classifier should discover the data generative mechanism or model the images manifolds. From this viewpoint, Hinton et al. were the first to use some mixture models of PCA or factor analysis (FA) to recognize handwritten digits [5]. Our recognition scheme is closely related to their work. However, some important differences exist. First, instead of applying the EM algorithm, we use the neural 'gas' algorithm to directly perform clustering, which has been proved optimal. Second, in Hinton's

method, each digit's manifold is modelled by a number of linear autoencoders which perform linear subspace projections. In our method, local fitting is provided by a nonlinear extension of PCA algorithm which is robust to noise and outliers. A last and most important difference lies in the classification criterion. In our method, we apply the population decoding concept to define an averaged 'distance' between a test digit and the subspaces describing the same class, whereas in Hinton's method, classification is directly set up on the reconstruction errors given by the autoencoders.

Our modular classification system demonstrates a very high performance on the handwritten digit recognition task. With moderate module size, 98.68% and 97% recognition rates have been achieved for training set and testing set, respectively, without any rejection. This performance is a combined result from (1) a modular approach (or one-class-one-net); (2) optimal partitioning input space by neural 'gas' algorithm and robust local fitting; and (3) population decoding for better measuring the fidelity of a module's fitting of digit image, thus offering a better decision.

References

1. E. Oja, "Subspace methods of pattern recognition," Research Studies Press, Letchworth, U.K. 1983.
2. E. Oja, "Neural networks, principal components and subspace," Int. J. Neural Syst., Vol. 1, pp. 61–68, 1989.
3. M.E. Tipping and C.M. Bishop, "Mixtures of probabilistic principal component analysers," Technical reports, Aston University, 1997.
4. R.D. Dony and S. Haykin, "Image segmentation using a mixture of principal components representation," IEE Proc.-Vis. Image Signal Process., Vol. 144, pp. 73–80, 1997.
5. G.E. Hinton, P. Dayan and M. Revow, "Modelling the manifolds of images of handwritten digits," IEEE Trans. Neural Networks, Vol. 8, pp. 65–74, 1997.
6. G.E. Hinton and Z. Ghahramni, "Generative models for discovering sparse distributed representations," Technical reports, University of Toronto, 1997.
7. T. Kohonen, "Self-organization and associative memory," Springer-Verlag, Berlin, 1989.
8. T. Kohonen, "Self-organized formation of various invariant-feature filters in the adaptive subspace SOM," Neural Computation, Vol. 9, pp. 1321–1344, 1997.
9. S.Y. Kung and J.S. Taur, "Decision-based neural networks with signal/image classification applications," IEEE Trans. Neural Networks, Vol. 6, pp. 170–181, 1995.
10. N. Kambhatla and T.K. Lee, "Dimension reduction by local principal component analysis," Neural Computation, Vol. 9, pp. 1493–1516, 1997.
11. L. Xu, "Least mean square error reconstruction principle for self-organization," Neural Networks, Vol. 6, pp. 627–648, 1993.
12. J. Karhunen and J. Joutsensalo, "Representation and separation of signals using nonlinear PCA type learning," Neural Networks, Vol. 7, pp. 113–127, 1994.
13. T.M. Martinetz, S.G. Berkovich and K.J. Schulten, " 'Neural-gas' network for vector quantization and its application to time-series prediction," IEEE Trans. Neural Networks, Vol. 4, pp. 558–568, 1993.
14. H.P. Snippe, "Parameter extraction from population codes: a critical assessment," Neural Computation, Vol. 8, pp. 511–529, 1996.
15. H. Yan, "Prototype optimization of a nearest neighbor classifier using a multi-layer neural network," Pattern Recognition, Vol. 26, pp. 317–324, 1992.
16. R. Anand, et al., "Efficient classification for multiclass problems using modular neural networks," IEEE Trans. Neural Networks, Vol. 6, pp. 117–124, 1995.