Handwritten Digit Recognition by Adaptive-Subspace Self-Organizing Map (ASSOM)

Bailing Zhang, Minyue Fu, Hong Yan, and Marwan A. Jabri

***Abstract*— The adaptive-subspace self-organizing map (ASSOM) proposed by Kohonen is a recent development in self-organizing map (SOM) computation. In this paper, we propose a method to realize ASSOM using a neural learning algorithm in nonlinear autoencoder networks. Our method has the advantage of numerical stability. We have applied our ASSOM model to build a modular classification system for handwritten digit recognition. Ten ASSOM modules are used to capture different features in the ten classes of digits. When a test digit is presented to all the modules, each module provides a reconstructed pattern and the system outputs a class label by comparing the ten reconstruction errors. Our experiments show promising results. For relatively small size modules, the classification accuracy reaches 99.3% on the training set and over 97% on the testing set.**

***Index Terms*— Adaptive-subspace self-organizing map, handwritten digit recognition, principal component analysis.**

I. Introduction

MANY neural networks learning methods have been proposed to realize principal component analysis (PCA). For example, an earlier approach for extracting the first principal component [1] has been extended to a network with multiple output units, with weight vectors spanning the subspace of the first M principal components [2]. Another popular learning rule which allows the weight vectors to converge to exactly the first M eigenvectors was proposed in [4]. The significance of PCA has been much discussed. For example, in pattern recognition, the subspace pattern recognition method (SPRM) [3] can be directly set up on PCA. However, as a linear method, PCA is inadequate in many real-world nonlinear problems. In recent years, many developments on nonlinear extension of PCA have been proposed. Examples include principal curves [5], [6] and multilayer autoencoders [8], which establish a global parametric or nonparametric model to describe the nonlinear data structure. Another paradigm is to use a mixture of local PCA to collectively model the data space.

Constructing a mixture of local PCA usually involves the partitioning of the data space followed by the estimation of the principal subspace within each partition. A common practice is to utilize the reconstruction errors from local principal subspace projections as the relevant distortion measures to guide the data space partitioning. For example, Dony and Haykin [8] and Kambhatla and Leen [9] independently proposed a kind of vector quantization (VQ)/PCA mixture model which first partitions the data into disjoint regions by VQ and then performs a local PCA about each cluster center. Hinton *et al.* [10] considered the partition assignments of examples among different PCA models in both k-means clustering procedure and the expectation maximization (EM) framework. Recently, Kohonen proposed a modular neural-network architecture called adaptive-subspace self-organizing map (ASSOM) [11], [12], which creates a set of local subspace representations by competitive selection and cooperative learning. In traditional SOM [13], [15], a set of reference vectors is spatially organized to partition the input space. In ASSOM, a number of submodels is topologically ordered, with each submodel responsible for describing a specific region of the input space by its local principal subspace. The ASSOM model is attractive not only because it inherits the topographic representation property in the original SOM, but also because the learning results of ASSOM can faithfully describe the kernels of various transformation groups. The simulation results in [11] and [12] have illustrated that different feature filters can be self-organized to different low-dimensional manifolds and a wavelet type representation does emerge in the learning.

In an ASSOM model, local subspaces can be adapted by linear PCA learning algorithms, which often converge slowly. More importantly, when applying a linear PCA algorithm to an ASSOM model, it is prone to instability problems. It is known that there are a number of advantages in introducing nonlinearities into a PCA type network [16]–[18]. For example, by extending the minimization problem of the mean-square representation error from a linear network to its nonlinear counterpart, the stability properties of the resulting learning algorithm can be much improved over the corresponding linear PCA learning algorithm. From this consideration, we are proposing to realize local principal component representation by an approximative principal subspace algorithm [16]–[18] and apply the ASSOM model to classification as a generalization of the traditional PCA-based subspace pattern recognition method (SPRM) [3].

For practical multiclasses classification problems, we can train a separate ASSOM model to describe each class of data and then classify an unknown data point according to whichever model gives the best match. As an application, we apply the ASSOM model to handwritten digit recognition. Although much progress has been made [19], handwritten digit recognition remains a difficult problem. A major reason is that it is often hard to successfully characterize the wide diversity inherent in handwritten digits. Due to the importance of invariance with respect to some basic transformation groups such as translation, rotation, scaling, some efforts have been made toward designing a recognizer which is tolerant to

Manuscript received June 30, 1998; revised March 30, 1999.

B. Zhang and M. Fu are with the Department of Electrical and Computer Engineering, University of Newcastle, NSW 2308, Australia.

H. Yan and M. A. Jabri are with the Department of Electrical Engineering, University of Sydney, NSW 2006, Australia.

Publisher Item Identifier S 1045-9227(99)05975-5.

some small transformations. For example, Simard *et al.* [20] have established a computationally expensive nearest neighbor method that allows for typical digit transformation. The elastic deformable matching [21] also attempts to capture all the variations with a single model, but using a very complex matching scheme. As ASSOM has an important feature that some basic transformation groups invariant filters (detectors) will emerge directly, we can use these models to combine knowledge from many examples about the diversity of characters.

This paper is organized as follows. In the next section, we first review the method of subspace pattern recognition and neural-network implementations. The least square reconstruction principle for a simple nonlinear autoencoder is introduced. In Section III, we propose an approximative implementation of ASSOM. Section IV presents ASSOM-based modular classification scheme for handwritten digit recognition and Section V gives experimental results. Finally, concluding remarks are discussed in Section VI.

II. Pattern Recognition Using Autoencoders

The goal of PCA is to find the M orthogonal directions in the L-dimensional data space that account for the greatest possible percentage of the data's variance. Projecting the data onto the M-dimensional subspace spanned by these M basis vectors produces the optimal dimensionality reduced description of the data in the sense that it achieves the minimum possible information loss. In pattern recognition, these properties have been straightforwardly used for classification, which is the subspace pattern recognition method (SPRM) [3]. In SPRM, certain linear subspaces within a pattern space are used to represent classes and the basis vectors that span the subspace define the features of the pattern. The most important significance of establishing a correspondence between classes and linear subspaces is that many important transformation groups can be automatically taken into account. Classification of an unknown pattern can then be set up on an efficiency metric by which the subspace can represent the data.

A pattern subspace is defined by its basis vectors. A set of M linearly independent vectors $\mathbf{u}_1, \cdots, \mathbf{u}_M$ in $R^L (M < L)$ spans a subspace $\mathcal{L}$

$$\begin{aligned}\mathcal{L} &= \mathcal{L}(\mathbf{u}_1, \cdots, \mathbf{u}_M) \\ &= \left\{ \mathbf{x} | \mathbf{x} = \sum_{m=1}^{M} \alpha_m \mathbf{u}_m, \quad \text{for some scalars } \alpha_1, \cdots, \alpha_M \right\}.\end{aligned} \tag{1}$$

The basic operation on a subspace is a projection of a vector $\mathbf{x}$ via $\hat{\mathbf{x}} = P\mathbf{x}$, where the projection matrix P of $\mathcal{L}$ is given by $P = U(U^TU)^{-1}U^T$, $U = [\mathbf{u}_1, \cdots, \mathbf{u}_M]$. If U is an orthonomal matrix, then P can be simplified to $P = UU^T$ and the projection of $\mathbf{x}$ by P is $\hat{\mathbf{x}} = UU^T\mathbf{x}$. In this case, the subspace $\mathcal{L}$ can be written as

$$\mathcal{L} = \{ \hat{\mathbf{x}} | \hat{\mathbf{x}} = UU^T\mathbf{x} \} \tag{2}$$

which is spanned by the M L-dimensional column vectors of U. The length of the corresponding orthogonal residual $\tilde{\mathbf{x}} = \mathbf{x} - \hat{\mathbf{x}}$ can be used as a measure for efficiency the subspace $\mathcal{L}$ describing data $\mathbf{x}$, thereby forms the classification criterion.

For a problem with K pattern classes, each class can be represented by its own subspace $\mathcal{L}^{(k)} = \mathcal{L}(\mathbf{u}_1^{(k)}, \cdots, \mathbf{u}_{M^{(k)}}^{(k)})$, where $M^{(k)}$ is the number of basis vectors in $\mathcal{L}^{(k)}$. An input pattern $\mathbf{x}$ is classified according to its distances from K classes, with the distance being evaluated by decomposing $\mathbf{x}$ in terms of each subspace's projection

$$\mathbf{x} = \hat{\mathbf{x}}^{(k)} + \tilde{\mathbf{x}}^{(k)}. \tag{3}$$

The class c to which $\mathbf{x}$ is assigned can be defined as

$$c = \arg \min_k \left\{ \| \tilde{\mathbf{x}}^{(k)} \| \right\}. \tag{4}$$

That is, we choose the closest subspace for each data point. This is in contrast to other classification approach such as nearest neighbor, where a single point represents a class. In SPRM, the decision boundaries between the classes (i.e., the boundaries where two subspaces are equally far in terms of orthogonal residual) are quadratic surfaces.

In the above SPRM, it is expedient to select principal components for the basis vectors of a subspace and use neural networks in the implementation [11], [12]. Amongst a number of neural learning paradigms, a three-layer feedforward network can be used to extract principal components, which has L nodes in the input and output layers and M nodes in the hidden layer. The network is usually called an autoassociative network or L-M-L autoencoder because it is trained to reproduce its input.

In this paper, we consider a symmetrical network structure in which L input elements x_i are transferred to M hidden units' activations y_m via the feedforward weights w_{ml}, $m = 1, \cdots, M$, $l = 1, \cdots, L$, $M < L$, $y_m = f(\sum_{l=1}^{L} w_{ml} x_l)$, with f being an activation function. The hidden layer activations are then carried to linear output units via connections $\hat{w}_{lm}$, $l = 1, \cdots, L$, $m = 1, \cdots, M$, yielding the outputs $\hat{x}_l = \sum_{m=1}^{M} y_m \hat{w}_{lm}$, $l = 1, \cdots, L$. We can rewrite the connection weights in matrix form $\mathbf{W}_{L \times M}$, $\hat{\mathbf{W}}_{M \times L}$. The symmetry assumption implies $\hat{\mathbf{W}} = \mathbf{W}^T$. We denote $\mathbf{w}(m)$ as the mth column vector of $\mathbf{W}$, then the reconstruction vector $\hat{\mathbf{x}}$ can be written as

$$\hat{\mathbf{x}} = \sum_{m=1}^{M} y_m \mathbf{w}(m). \tag{5}$$

Consider the optimization criterion

$$\begin{aligned}\text{minimize } J &= E\{ \| \mathbf{x} - \hat{\mathbf{x}} \|^2 \} \\ &= E\{ \| \mathbf{x} - \mathbf{W}y \|^2 \}\end{aligned} \tag{6}$$

where E stands for the expectation operator. Obviously, (6) is a generalization of the least square reconstruction problem leading to the standard PCA. Detailed studies of the objective (6) have been given in [16]–[18], with the gradient descent based learning rule as follows:

$$\mathbf{W}_{t+1} = \mathbf{W}_t + \mu_t \left[\mathbf{x} \left(\mathbf{e}_t^T \mathbf{W}_t \text{diag}(\mathbf{g}_t) \right) + \mathbf{e}_t \mathbf{y}_t^T \right] \tag{7}$$

where t is the time scale, $\text{diag}(\mathbf{g}_t)$ is the diagonal matrix whose mth element is the derivative of $\mathbf{y}_m$, i.e., $g_m = f'(\mathbf{x}^T \mathbf{w}(m))$, $m = 1, \cdots, M$. $\mathbf{e}_t = \mathbf{x} - \mathbf{W}_t^T \mathbf{y}_t$.

The close relationship between autoencoders and principal component analysis has been discussed in many papers, for example, [16]–[18] and [22]. For a linear autoencoder, i.e., $\mathbf{y} = \mathbf{W}^T\mathbf{x}$, it has been proven that the autoencoder acts like a linear PCA filter. In this case, the output $\mathbf{W}\mathbf{W}^T\mathbf{x}$ reconstructs the input with a corresponding squared reconstruction error $E = ||\mathbf{x} - \hat{\mathbf{x}}||^2$, which measures how well the model fits the data. Similarly, in a nonlinear autoencoder, the activation $\mathbf{y} = f(\mathbf{W}^T\mathbf{x})$ provides the coefficients for linear combination of the basis vectors $\mathbf{w}(1), \cdots, \mathbf{w}(M)$ and the reconstruction error $||\mathbf{x} - \hat{\mathbf{x}}||^2 = ||\mathbf{x} - \mathbf{W}\mathbf{y}||^2$ can be considered as a distance between $\mathbf{x}$ and the subspace spanned by $\mathbf{w}(m)$, $m = 1, \cdots, M$. Algorithm (7) can be termed as an approximative subspace algorithm because its results are different from but quite close to strict PCA solutions, especially for mild nonlinearities, and the learned weight vectors of different units are typically not exactly orthogonal, but not far from being orthogonal [17], [18]. In this paper, we choose f as a typical sigmoidal function, i.e., $f(t) = 1/(1 + e^{-\beta t})$, with parameter β controlling the nonlinearity.

As pointed out in [17], using nonlinear activation in a PCA type network implicitly introduces higher-order statistics into the computations and increases the independence of the components. In addition, an attractive merit of the approximative subspace algorithm (7) lies in its numerical benefits. It has been observed by several researchers that algorithm (7) is not sensitive to local minima and has much better stability property compared with some standard neural PCA learning algorithms. Therefore, an autoencoder with sigmoidal activation in the "bottleneck" layer can be applied together with the principle of SPRM. The squared error between the autoencoder input and output is a measure of the distance between the actual data point and its projection onto the lower-dimensional subspace employed by the bottleneck layer. In practice, we can train separate autoencoder for each class of data and classify an unknown data point according to whichever autoencoder produces the smallest reconstruction error.

III. Self-Organization of Adaptive Subspaces

Basically, a set of K autoencoders can be trained competitively so that each one contributes its own representation to a specific data class. Given a data point from a class, the reconstruction errors are first compared for all networks. In either way, we can allow the network with the smallest reconstruction error to learn and keep all other networks unchanged, or we can let each network adapt with a step size depending on the portion of its reconstruction error in all the errors. In the following, we consider the competitive learning SOM [15] proposed by Kohonen which plays an important role as a component in a variety of natural and artificial neural information processing systems.

A. A Brief Review of SOM

The underlying principle of SOM (and its variants) is the preservation of the probability distribution and the topology. The SOM model usually uses a simple single layer network, where output units affiliated with a predefined topology compete for each input. Units that are neighbors of the winner c update their weights $\mathbf{v}(m)$, $m = 1, \cdots, M$, together with the winner unit, according to

$$\mathbf{v}_{t+1}(m) = \mathbf{v}_t(m) + \mu_t h_{mc}(\mathbf{x} - \mathbf{v}_t(m)), \quad m = 1, \cdots, M \tag{8}$$

where t is the time scale and h_{mc} is a unimodal function that decreases monotonically for increasing distance between c and m. In this original form of the SOM algorithm, arbitrary sample vectors are compared to weight vectors using a metric for measuring their distances in the input space. The algorithm can be generalized by associating each unit with a dynamic operator, which is the idea of operator map proposed in [14]. ASSOM is a more recent development of SOM [11]–[13]. An essential architectural difference between ASSOM and traditional SOM is that the simple formal neuron in SOM is replaced by a basic operational unit, which could be a module consisting of a linear input layer and a quadratic neuron. The input pattern is compared with the signal subspace represented by the module. The learning results of ASSOM are most descriptive of the kernels of the transformation groups [11], [12]. In other words, the various feature filters emerge in learning and become tuned to different low-dimensional manifolds.

In [11] and [12], the local subspaces are adapted using a linear learning subspace method, which is computational costly for real problems. In [11] and [12], and to overcome the algorithm's stability problem, measures were proposed in order to achieve a sufficient stability in the self-organizing process. As we discussed in the last section, approximative subspace algorithm (7) has a number of advantages. In our work, we propose to implement ASSOM by adapting the local subspaces with algorithm (7), as detailed below.

B. Implementation of an Approximative ASSOM

Consider K autoencoders with each one affiliated with a predefined topology. The kth autoencoder associates $L \times M^{(k)}$ weight matrix $\mathbf{W}^{(k)}$, where $M^{(k)}$ is the number of units in its bottleneck layer. When an input pattern is presented to all the networks, the cth network with the smallest reconstruction error is called the winner and satisfies the following condition:

$$||\mathbf{x} - \hat{\mathbf{x}}^{(k^*)}|| < ||\mathbf{x} - \hat{\mathbf{x}}^{(k)}||, \qquad \text{for } k \neq k^* \tag{9}$$

where $\hat{\mathbf{x}}^{(k)}$ is the reconstruction vector from kth network, $\hat{\mathbf{x}}^{(k)} = \mathbf{W}^{(k)} f(\mathbf{W}^{(k)T}\mathbf{x})$. The autoencoders can be defined to be topologically ordered, if for each autoencoder k, the principal subspace representations provided by its immediate neighbors are closer to principal subspace representation provided by kth autoencoder. An ASSOM can be realized by first selecting a winning autoencoder c and then choosing a neighborhood set N_c around c, which determines those autoencoders within c's neighborhood. All the autoencoders within the N_c adapt their weights according to the following

learning rule:

$$\mathbf{W}_{t+1}^{(k)} = \begin{cases} \mathbf{W}_t^{(k)} + \mu_t \Big[\mathbf{x}\Big(\mathbf{e}_t^{(k)^T} \mathbf{W}_t^{(k)} \operatorname{diag}\Big(\mathbf{g}_t^{(k)}\Big)\Big) \\ \quad + \mathbf{e}_t^{(k)} \mathbf{y}_t^T \Big], & \text{for } k \in N_c \\ \mathbf{W}_t^{(k)} & \text{otherwise.} \end{cases} \tag{10}$$

As in the Kohonen's SOM, the size of N_c starts large and slowly decreases over time. For a two dimensional topology, a square array or hexagonal array topological neighborhood shape can be selected. However, such a neighborhood selection procedure usually results in slow convergence. A better alternative is to replace the neighborhood set N_c by a continuous neighborhood interaction function h_{ij} of the distance between units i and j in the lattices, as in (8). The corresponding learning algorithm then becomes

$$\mathbf{W}_{t+1}^{(k)} = \mathbf{W}_t^{(k)} + \mu_t h_{kc} \Big[\mathbf{x}\Big(\mathbf{e}_t^{(k)^T} \mathbf{W}_t^{(k)} \cdot \operatorname{diag}\Big(\mathbf{g}_t^{(k)}\Big)\Big) + \mathbf{e}_t^{(k)} \mathbf{y}_t^T \Big], \quad k = 1, \cdots, K \tag{11}$$

where the neighborhood interaction function h_{kc} is a monotonically decreasing function of the distance d_{kc} between autoencoders k and c in the ASSOM, typically selected as a Gaussian function

$$h_{kc} = \exp\left(-\frac{d_{kc}^2}{2\sigma^2}\right). \tag{12}$$

In practice, it has been found useful for convergence to start off with a wide range σ and then gradually reduce it during learning. This allows the networks initially to form a crude ordering, and then refine them with respect to the inputs. The learning rate μ_t is typically reduced during learning.

If different transformations exist in the input patterns, different networks of ASSOM's autoencoders become tuned to these transformation classes. In other words, each autoencoder can be made to become invariant to one transformation type and decode a certain range of features invariantly of this transformation. Our implementation of the ASSOM algorithm (11) is always stable with regard to the initial weight selection, learning step size, and input range.

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

1) For an input pattern, determine a winner network c in K autoencoders, the subspace $\mathcal{L}^{(c)}$ of which is closest to input $\mathbf{x}$ based on the reconstruction distance (9). Then adapt the local subspaces via the learning rule of (11) for each autoencoder with the step size being proportional to h_{kc} in (12).
2) Stop if the adaptations have converged, otherwise pick a new example and return to Step 1).

IV. An ASSOM-Based Modular Classification Scheme for Handwritten Digit Recognition

Neural networks have been often exploited in handwritten digit recognition and a common practice is to train a multilayer perceptron (MLP) classifier to output one of the ten class labels. In general, an MPL classifier can yield quite different classification boundaries with respect to different initial conditions or different training sets from the same data space. A better alternative is to train a separate model on examples of each class and to classify unknown data points by checking which model offers the best reconstruction of the data. This idea was proposed in [10] for handwritten digit recognition, where linear PCA is employed as local model. Motivated by their works, we focus our attention on classification performance of the ASSOM. 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 directly use ASSOM paradigm to introduce competition among the modules. In [10], each digit's manifold is modeled by a number of linear autoencoders which performs linear subspace projections. In our method, local modeling is implemented by a nonlinear autoencoder.

In this paper, we have used a handwritten digit database of the U.S. National Institute of Standards and Technology (NIST), which consists of 20 000 numerals. The numerals of this database have been digitized in bilevel on a 25×20 grid. Among the data, 10 000 numerals were used for training and another 10 000 for testing. In our experiment, we directly use the digit bitmaps without a prestage to extract features. Our modular recognition system shown in Fig. 1 has ten modules to describe the ten digit classes. Each module consists of a set of K autoencoders and each autoencoder has M hidden nodes and $L = 500$ input nodes corresponding to the pixel values in a numeral bitmap. Learning proceeds in four cycles with samples taken from the training set. The learning parameter μ in (11) is initially set to 1 and then dynamically decreases to 0.1. The decay constant σ in the interaction function h_{kc} changes from c to $c/100$, where c is the size of a predefined square grid. The time dependence for these parameters takes a similar form, i.e., $g(t) = g_i(g_f/g_i)^{t/t_{\max}}$, in which t is the current adaptation step, $t_{\max}$ is a predefined maximum adaptation step, $t_{\max} = 40\,000$ in all our experiments. The subscripts i and f stand for initial value and final value, respectively. We show the converged weight vectors in Fig. 2(a) and (b), corresponding to the first and second components in each class, respectively. There are 49 classes in each ASSOM module and $M = 2$ principal components in each class. Each weight vector is visualized in mask form after being equalized to 256 greylevels.

After each module is trained by the examples of its class, the classification of an unlabeled input digit is performed by finding which module best reconstruct the input pattern. Obviously, the problem is how to yield an overall reconstruction from an ASSOM module. Consider the reconstruction vectors $\hat{\mathbf{x}}^{(kl)}$, $k = 1, \cdots, K$, from K autoencoders in lth ASSOM module, $l = 1, \cdots, 10$. In order to determine an overall reconstruction, we first specify a virtual response function a_{kl} of the kth autoencoder and then an overall reconstruction vector $\hat{\mathbf{x}}^{(l)}$ associated with it $a_{1l}, \cdots, a_{Kl}$ is simply given as

TABLE I
CLASSIFICATION ACCURACIES OF THE ASSOM. IN THESE EXPERIMENTS, $\beta = 0.1$. c IS THE SIZE OF THE PREDEFINED GRID ($K = c^2$ IS THE NUMBER OF CLASSES) AND M IS THE NUMBER OF PRINCIPAL COMPONENTS IN EACH CLASS. BOTH TRAINING AND TESTING DATA-SET HAVE 10 000 SAMPLES

c	M=1		M=2		M=3	
	training set	**testing set**	**training set**	**testing set**	**training set**	**testing set**
3	95.58%	94.91%	96.71%	95.56%	97.61%	95.95%
4	96.63%	95.61%	97.96%	96.08%	98.32%	96.83%
5	98.03%	96.13%	98.47%	96.43%	99.07%	96.82%
6	98.37%	96.52%	98.95%	96.76%	99.28%	97.18%
7	98.73%	96.72%	99.12%	96.9%	99.45%	97.26%
8	99.30%	97.25%	99.3%	97.35%	99.65%	97.87%

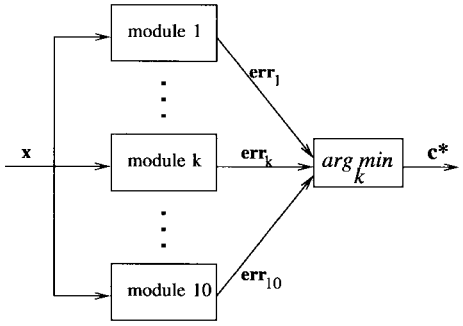
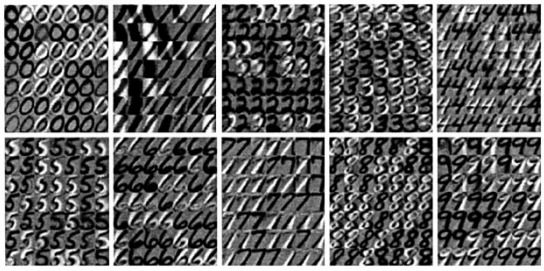
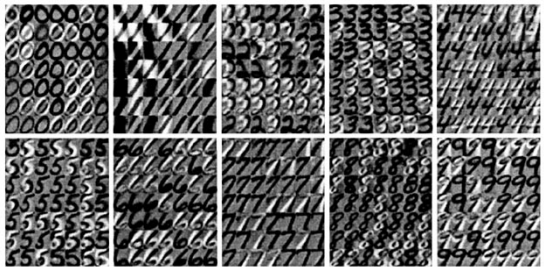


Fig. 1. Our proposed modular classification system for handwritten digits based on the ASSOM model.



(a)



(b)

Fig. 2. (a) and (b) The weight vectors corresponding to the first and second components in each class, respectively. There are 49 classes in each ASSOM module and $M = 2$ principal components in each class. The converged weight vectors are visualized in mask forms after being equalized to 256 greylevels.

the weighted average over all $\hat{\mathbf{x}}^{(kl)}$, i.e.,

$$\hat{\mathbf{x}}^{(l)} = \frac{\sum_{k=1}^{K} a_{kl}\hat{\mathbf{x}}^{(kl)}}{\sum_{k=1}^{K} a_{kl}}, \qquad l = 1, \cdots, 10. \tag{13}$$

A simple way of establishing a virtual response function a_{kl} is to use Gaussian functions with centers at $\hat{\mathbf{x}}^{(kl)}$, e.g.,

$$a_{kl} = \exp\left(-\frac{||\mathbf{x} - \hat{\mathbf{x}}^{(kl)}||^2}{2\kappa_{kl}^2}\right), \quad k = 1, \cdots, K, \quad l = 1, \cdots, 10 \tag{14}$$

where κ_{kl} is a parameter that controls the response range of the kth autoencoder.

Based on the above discussion, for a test sample $\mathbf{x}$, we compute the reconstruction error from each ASSOM module which measures the faithfulness the module in representing the data. The recognition process can then be simply performed by comparing the reconstruction errors err_l

$$\mathrm{err}_l = ||\mathbf{x} - \hat{\mathbf{x}}^{(l)}||^2, \qquad l = 1, \cdots, 10 \tag{15}$$

where l indicates the number of modules, $\hat{\mathbf{x}}^{(l)}$ is calculated using (13). Then, the class label is associated with the module with the smallest error, i.e., $\mathbf{x}$ is assigned to the class c^* if

$$c^* = \arg\min_c \mathrm{err}_c. \tag{16}$$

V. EXPERIMENTAL RESULTS

In the following, we turn to report results of experiments illustrating the recognition performance of the proposed classification. In Table I, we show the experimental results for the classification accuracy, which compares different number of classes in each module ($K = c^2$, where c is the size of a predefined square grid). The experiments are performed with $\beta = 0.1$. The parameter κ in (14) is taken as one. Intuitively, increasing the number of classes in each module can bring better recognition accuracy, but larger size of ASSOM modules will slow down the training and classification. As demonstrated in Table I, 64 classes ($c = 8$) in each module result in a satisfactory performance. Adding more principal components in each class may further improve the recognition accuracy when M is relatively small and an appropriate choice from our experience is $M = 2$–3. We also found that the sigmoidal nonlinearity parameter β has no significant influence on the recognition results when it is kept in a small range, e.g., $\beta < 1$.

In general, the parameter κ in (14) will also influence the classification accuracy, and as it controls the autoencoder response range, it should be chosen to be relatively small. We have assessed different values of κ and found that a smaller κ brings a better classification accuracy on the training set while increasing κ in a limited extent can improve generalization. In

practice, κ can be chosen within a reasonable range without significant difference.

In the handwritten digits samples, there are some samples with considerable variances in shapes, stroke widths, etc., which are harder to be correctly classified. In practice, a very small error rate is often acceptable. When a recognition system is established, error rate can be lowered by rejecting some test patterns. In our scheme, 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

$$\eta = 1 - \frac{\text{err}_j}{\text{err}_i} \tag{17}$$

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

$$\begin{aligned} &\mathbf{x} \text{ is rejected from classification if } \eta > \eta_T \\ &\mathbf{x} \text{ is accepted for classification if } \eta \leq \eta_T \end{aligned} \tag{18}$$

where η_T is a threshold which can be experimentally determined. Usually, the error rate is lowered by increasing the threshold η_T and a larger η_T means a higher rejection rate. The relationship between the error rate and the rejection is shown in Fig. 3(a), in which the rejection rates corresponding to varying threshold η_T from 0.01 to 0.1 are shown, and where the error rate and rejection rate are defined as follows:

$$\text{error rate} = \frac{\text{number of misrecognized test patterns}}{\text{number of test patterns}} \tag{19}$$

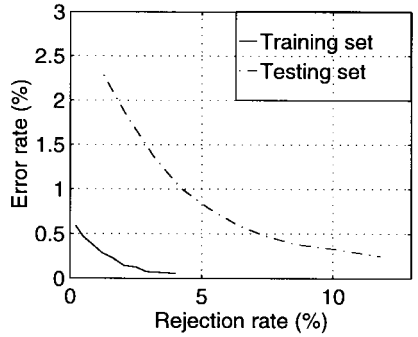
$$\text{rejection rate} = \frac{\text{number of rejected patterns}}{\text{number of test patterns}}. \tag{20}$$

In addition to error rate, another index for evaluating handwritten digit recognition is the reliability, which refers to the portion of correctly recognized patterns in all the test patterns. In Fig. 3(b), we illustrated the relationship between the reliability and the rejection rate. From Fig. 3 we can see that the lowest error rate is less than 0.5% with rejection rate of 10% on the testing data set. This shows that the ASSOM based recognition system can achieve a high recognition and low error rates. Reducing the rejection rate will cause an increase in the error rate.

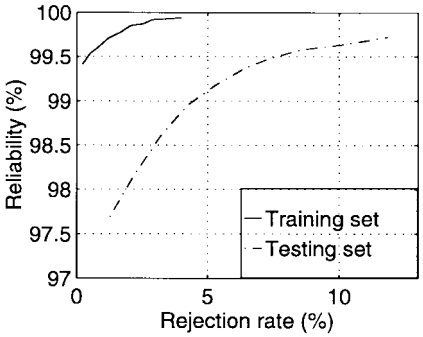
In practical recognition systems, we can set up a relatively high rejection rate and employ a multistage recognition scheme. Our proposed method can serve as the first stage which correctly recognize most of the test digits and make some rejections during the recognition process. For those rejected digits, another stage then proceeds which undertakes structural analysis or involves some other possibly more costly recognition methods.

VI. Discussion and Conclusion

In this paper, we have presented a modular classification scheme for handwritten digit recognition, in which each module is an ASSOM model for modeling the manifolds of handwritten digits bitmaps. Each module is composed of a number of topologically ordered autoencoders which corresponds to different subspaces 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. We addressed the importance of a PCA type learning which is based on the least square representation error principle in a nonlinear autoencoder network. Compared with exact PCA learning algorithm, this approximative subspace learning algorithm is numerically stable and robust to noise. The use of ASSOM model as a classifier produces promising results. With each module having 64 autoencoder ($M = 2$), the recognition rate is about 99.3% on the training set and over 97% on the testing set, with no rejection. The lowest error rate is less than 0.5% with rejection rate of 10% on the testing data-set.



(a)



(b)

Fig. 3. (a) Relationship between the error rate and the rejection rate. (b) Relationship between the reliability and the rejection rate. In the simulation, 64 classes in each ASSOM module and $M = 2$ principal components in each class are exploited.

References

[1] E. Oja, "Simplified neuron model as a principal component analyzer," *J. Math. Biol.*, vol. 15, pp. 267–273, 1982.

[2] ———, "Neural networks, principal components and subspace," *Int. J. Neural Syst.*, vol. 1, pp. 61–68, 1989.

[3] ———, *Subspace Methods of Pattern Recognition.* Letchworth, U.K.: Res. Studies, 1983.

[4] T. D. Sanger, "Optimal unsupervised learning in a single-layer linear feedforward neural network," *Neural Networks*, vol. 2, pp. 459–473, 1989.

[5] T. Hastie and W. Stuetzle, "Principal curves," *J. Amer. Statist. Assoc.*, vol. 84, pp. 502–516, 1989.

[6] A. R. Webb, "An approach to nonlinear principal components analysis using radially symmetric kernel functions," *Statist. Comput.*, vol. 6, pp. 159–168, 1996.
[7] M. Kramer, "Nonlinear principal component analysis using autoassociative neural networks," *AIChE J.*, vol. 37, pp. 233–243, 1991.
[8] R. D. Dony and S. Haykin, "Image segmentation using a mixture of principal components representation," *Proc. Inst. Elect. Eng.—Vis. Image Signal Processing*, vol. 144, pp. 73–80, 1997.
[9] N. Kambhatla and T. K. Lee, "Dimension reduction by local principal component analysis," *Neural Comput.*, vol. 9, pp. 1493–1516, 1997.
[10] G. E. Hinton, P. Dayan, and M. Revow, "Modeling the manifolds of images of handwritten digits," *IEEE Trans. Neural Networks*, vol. 8, pp. 65–74, 1997.
[11] T. Kohonen, "Emergence of invariant-feature detectors in the adaptive-subspace self-organizing map," *Biol. Cybern.*, vol. 75, pp. 281–291, 1996.
[12] T. Kohonen, S. Kaski, and H. Lappalainen, "Self-organized formation of various invariant-feature filters in the adaptive subspace SOM," *Neural Comput.*, vol. 9, pp. 1321–1344, 1997.
[13] T. Kohonen, E. Oja, O. Simula, A. Visa, and J. Kangas, "Engineering applications of the self-organizing map," *Proc. IEEE*, vol. 84, pp. 1358–1384, 1996.
[14] T. Kohonen, "Generalizations of the self-organizing map," in *Proc. Int. Joint Conf. Neural Networks*, Nagoya, Japan, 1993.
[15] ———, *Self-Organization and Associative Memory*. Berlin, Germany: Springer-Verlag, 1989.
[16] L. Xu, "Least mean square error reconstruction principle for self-organization," *Neural Networks*, vol. 6, pp. 627–648, 1993.
[17] J. Karhunen and J. Joutsensalo, "Representation and separation of signals using nonlinear PCA type learning," *Neural Networks*, vol. 7, pp. 113–127, 1994.
[18] ———, "Generalizations of principal component analysis, optimization problems, and neural networks," *Neural Networks*, vol. 8, pp. 549–562, 1995.
[19] C. Y. Suen, C. Nadal, R. Legault, T. A. Mai, and L. Lam, "Computer recognition of unconstrained handwritten numerals," *Proc. IEEE*, vol. 80, pp. 1162–1180, 1992.
[20] P. Simard, Y. L. Cun, and J. Denker, "Efficient pattern recognition using a new transformation distance," in *Advances in Neural Inform. Processing Syst. 5*, J. D. Cowan, S. J. Hanson, and C. L. Giles, Eds. San Mateo, CA: Morgan Kaufmann, 1993, pp. 50–58.
[21] M. Revow, C. K. I. Williams, and G. E. Hinton, "Using generative models for handwritten digit recognition," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 18, pp. 592–606, 1996.
[22] H. Bourlard and Y. Kamp, "Autoassociation by multilayer perceptrons and singular value decomposition," *Biol. Cybern.*, vol. 59, pp. 291–294, 1988.