Kernel Autoassociator with Applications to Visual Classification

Haihong Zhang, Weimin Huang
Insititute for Infocomm Research
Singapore 119613
{hhzhang,wmhuang}@i2r.a-star.edu.sg

Zhiyong Huang
School of Computing
National University of Singaore
huangzy@comp.nus.edu.sg

Bailing Zhang
School of Computer Science and Mathematics
Victoria University, Australia
bzhang@csm.vu.edu.au

Abstract

Autoassociator is an important issue in concept learning, and the learned concept of a particular class can be used to distinguish the class from the others. For nonlinear autoassociation, this paper presents a new model referred to as kernel autoassociator. Using kernel feature space as a potential nonlinear manifold, the model formulates the autoassociation as a special reconstruction problem from kernel feature space to input space. Two methods are developed to solve the problem. We evaluate the autoassociator with artificial data, and apply it to handwritten digit recognition and multiview face recognition, yielding positive experimental results.

1. Introduction

Autoassociator, also referred to as autoassociative memory or autoassociative network, is a brain-like distributed memory that learns from a set of samples $\{\mathbf{x}_i\}$ to perform the pattern reconstruction by ([1] pp.75)

$$F : \mathbf{x}_i \rightarrow \hat{\mathbf{x}}_i \tag{1}$$

Thereby, through learning dependencies among the samples in the autoassociation, the network can find the commonalities and capture the concept of the particular class [2, p.72]. For instance, Kohonen demonstrated in an early work that an autoassociative memory can be used to store and retrieve face images [3].

Autoassociators themselves do not produce explicit responses to classification tasks. Instead, a set of such networks can to be created for each class, and classifying a given pattern can be achieved by comparing the individual autoassociation results - as the network dedicated to the true class would produce the best results [4].

In this field, the linear networks such as correlation associative memory [3] exhibit the simplest form, but they have limitations in exploring high-order dependency among the data. Therefore, they are inappropriate for classification tasks involving multimodal and nonlinearly distributed patterns [5], such as face images under varying lighting conditions or pose angles.

Previous research usually addresses the nonlinear issue by using autoassociative Multi-Layer Perceptrons (AA-MLPs), which have been widely used in computer vision applications [6][7]. During learning, the hidden units in the network could build for the input pattern an internal representation that is useful for pattern reproduction through hidden-to-output layer connections.

However, it is known that for high-dimensional data, autoassociative MLPs may have difficulties in efficient training. Moreover, it has been found that autoassociative MLPs are somehow equivalent to linear PCA in many cases [6].

The goal of this paper is to propose an efficient approach to nonlinear autoassociation. We introduce kernel method [8] to nonlinear autoassociation, yielding a new network model referred to as kernel autoassociator. It is known that kernel method has been well-established as an efficient way to nonlinear analysis in recently years. For nonlinear autoassociation, we can use it as a generic and accessible method to obtain nonlinear features.

In particular, the kernel network achieves the autoassociation mapping F through kernel feature space in two steps: the mapping from input space to kernel feature space; the mapping from the feature space backwards to input space. The latter one is stressed in the present work. Two methods are developed for the backward mapping. One involves linear manipulations in kernel feature space, while the other

Proceedings of the 17th International Conference on Pattern Recognition (ICPR'04)
1051-4651/04 $ 20.00 IEEE

employs polynomial functions in a kernel feature subspace spanned by kernel principal components [9].

The proposed approach has been evaluated with simulations, and applied to face recognition (FR) (UMIST multiview face database [10]) and optical character recognition (OCR) (USPS handwritten digit database). The results demonstrate the effectiveness of this nonlinear autoassociator model, and suggest that the proposed approach can achieve high performance for classification, comparable to other state-of-the-art techniques.

2. Kernel Autoassociator

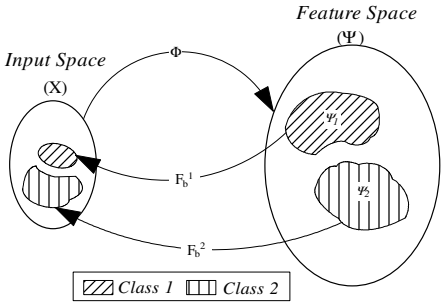

Figure 1. The kernel autoassociation

A conceptual depiction of kernel autoassociation is given in Fig. 1, where the autoassociator performs in two steps: first a pattern $\mathbf{x}$ is implicitly transferred to a kernel feature space Ψ by $\Phi(\mathbf{x})$, and is then mapped backwards to input space by

$$F_b^{(c)} : \Phi(\mathbf{x}) \rightarrow \hat{\mathbf{x}} \text{ , for } \mathbf{x} \in \text{ class } c \tag{2}$$

where $\Phi(\mathbf{x})$ is the kernel feature of $\mathbf{x}$. Φ together with F_b would constitute a complete kernel autoassociator. Strictly, the $\Phi(\mathbf{x})$ may be in implicit form, but it has an important property: the dot products of Φ can be performed by means of kernel functions in input space

$$k(\mathbf{x}_i, \mathbf{x}_j) = (\Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j)) \tag{3}$$

where $(\cdot)$ denotes a dot product, and k the kernel function in input space. By this, the problems in kernel feature space may be expressed and solved in forms of kernel functions.

Common kernel functions for k include Gaussian kernel, polynomial kernel and sigmoid kernel. Since the kernel mapping has been well addressed in the literature, this work will emphasis on another part of kernel autoassociator: the backward mapping F_b, for which a linear method and a polynomial method for F_b are developed in the following.

3. Linear Functions for F_b

In its simplest form, F_b would be a linear function on Φ. Because of the intrinsic nonlinearity of kernel feature space, the complete autoassociator is still nonlinear. This leads to a new expression of Eq. (2)

$$\hat{x} = F_b(\Phi(\mathbf{x})) = \vec{\beta}_\phi^T \Phi(\mathbf{x}) \tag{4}$$

where $\hat{x}$ is a component of $\hat{\mathbf{x}}$, $\vec{\beta}_\phi$ a kernel feature vector. Suppose $\vec{\beta}_\phi$ can be spanned by the given M examples.

$$\vec{\beta}_\phi = \sum_{i=1}^{M} b_i \Phi(\mathbf{x}_i) \tag{5}$$

then we can rewrite Eq. 4 as

$$\hat{x} = \sum_{i=1}^{M} b_i \Phi(\mathbf{x}_i)\Phi(\mathbf{x}) = \sum_{i=1}^{M} b_i k(\mathbf{x}_i, \mathbf{x}) = \mathbf{b}^T \mathbf{k} \tag{6}$$

where $\mathbf{b} = [b_1, \ldots, b_N]$ and $\mathbf{k}$ is the kernel product vector ($\mathbf{k}_i = k(\mathbf{x}_i, \mathbf{x})$). Hence for the complete output vector $\hat{\mathbf{x}}$, the backward mapping can be expressed in matrix form by

$$\hat{\mathbf{x}} = B\mathbf{k} \tag{7}$$

where $B = [\mathbf{b}_1, \ldots, \mathbf{b}_D]$, D the dimension of $\hat{\mathbf{x}}$. Interestingly, it is the same as the expression of kernel associative memory (KAM) [4] proposed earlier, which however is derived in a different way as an extension of correlation associative memory. This suggests that KAM can be viewed as a linear instance of kernel autoassociator. And we can just refer to [4] for training the network.

4. Polynomials for F_b

Let F_b consist of 2nd order multivariate polynomial functions in the kernel feature space

$$\hat{x} = F_b(\hat{\Phi}(\mathbf{x})) = \Phi^T(\mathbf{x}) W_\phi \hat{\Phi}(\mathbf{x}) + \vec{\beta}_\phi^T \hat{\Phi}(\mathbf{x}) + c_\phi \tag{8}$$

where W_ϕ,$\vec{\beta}_\pi$ and c_ϕ are the multivariate polynomial coefficients. $\hat{\Phi}$ is the centered feature vector given by $\Phi - \frac{1}{M}\sum_{i=1}^{M} \Phi(\mathbf{x}_i)$. The direct calculation of Eq. 8, however, may be inapplicable because the kernel feature vector Φ can be implicit. A practical solution is to find a low-dimensional representation of $\hat{\Phi}(\mathbf{x})$ such that the feature vector can be written as

$$\hat{\Phi}(\mathbf{x}) = \sum_{i=1}^{N} \alpha_i \mathbf{v}_i = (\mathbf{v}_1\ \mathbf{v}_2\ \mathbf{v}_3 \ldots)(\alpha_1\ \alpha_2\ \ldots)^T = V\vec{\alpha} \tag{9}$$

where V is the matrix with each column a basis of the subspace, and $\vec{\alpha}$ the projections of $\hat{\Phi}$ onto V.

And we can arrive at a new form of Equation 8.

$$\hat{x} = F_\alpha(\vec{\alpha})) = \vec{\alpha}^T W \vec{\alpha} + \vec{\beta}^T \vec{\alpha} + c \quad (10)$$

which is a polynomial funciton on the coefficient vector $\vec{\alpha}$, and W denotes $V^T W_\phi V$.

In practice, the low-dimensional representation in Eq. (9) can be obtained by KPCA which performs linear PCA in the kernel feature space by an elegant method [9]. In particular, the vector $\vec{\alpha}$ is obtained as $\vec{\alpha} = A\mathbf{k}_c$, where A is the KPCA projection matrix and $\mathbf{k}_c$ is given by

$$\mathbf{k}_c = \mathbf{k} - K_g 1'_m - 1_m \mathbf{k} + 1_m K_g 1'_m = J_m \mathbf{k} + \mathbf{k}_m \quad (11)$$

with K_g the gram matrix of prototypes: $K_g^{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$, I_M a $M \times M$ matrix with all entries equal to $1/M$, $1'_m$ a $1 \times M$ vector with all entries equal to $1/M$. And we also use $\mathbf{k}_m$ to represent the term $(K_g 1'_m + 1_m K_g 1'_m)$ independent of the kernel product vector $\mathbf{k}$, use J_m to represent $I - 1_m$. For details please refer to [9].

Importantly, from the above we can derive a new expression of Eq. 10 directly on kernel product vector $\mathbf{k}$

$$\hat{x} = F_k(\mathbf{k}) = \mathbf{k} W_q \mathbf{k} + \vec{\beta}_q \mathbf{k} + c_q \quad (12)$$

And β, W_q and c_q can be calculated as in Table 1 that gives the relationship between a polynomial in KPCA subspace $(\vec{\alpha})$ and the equivalent polynomial on kernel product vector $(\mathbf{k})$. By this, we can bypass the explicit KPCA calculations of $\vec{\alpha}$ to reduce computational complexity in running kernel autoassociator.

$F_\alpha(\vec{\alpha})$	$F_k(\mathbf{k})$
W	$W_q = J_m^T A^T W A J_m$
$\vec{\beta}$	$\vec{\beta}_q = 2K_m^T A^T W A J_m + \vec{\beta}^T A J_m$
c	$c_q = K_m^T A^T W A K_m + \vec{\beta}^T A K_m + c$

Table 1. Equivalent kernel polynomials

For training the autoassociator from a given set of samples, we have to determine those polynomial coefficients W,$\vec{\beta}$ and c. Although the function is nonlinear in the variable $\vec{\alpha}$, fortunately it is linear in the coefficients, so the solution can be achieved by linear approaches using the least mean square error criteria. The details are omitted here due to the space limitation.

5. Evaluation with Spiral Data

To examine the proposed method, we first generate a set of patterns on a 2D spiral as plotted in the leftmost panel of Fig. 2. The patterns are used to train an autoassociative MLP (AA-MLP), a kernel autoassociator with linear F_b (hereafter referred to as KAA-1) and a kernel autoassociator with polynomial F_b (hereafter referred to as KAA-2), respectively. Next, we evaluate the reconstruction error of each 2D pattern by each network, yielding error surfaces in the form of image in Fig. 2, where the reconstruction error is denoted by the pixel intensity. For the AA-MLP, we use a hidden layer of 12 hidden neurons, though more neurons yield similar result in our tests. For KAAs, we use Gaussian kernel functions k. KAA-2 employs 20 KPCA features.

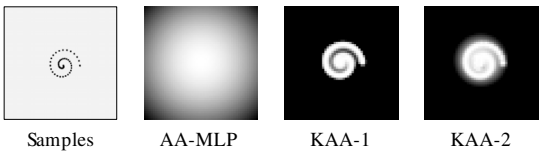

Figure 2. Evaluation with spiral data

Though the patterns exhibit a nonlinear and complex distribution thus posing a problem to AA-MLP, apparently the proposed kernel networks can produce reconstruction error surfaces that correctly capture the underlying structure of the data.

6. Application to OCR

US-Postal Service (USPS) handwritten digit database consists of 7291 training patterns and 2007 test patterns of 16×16 pixels. Fig. 3 shows some reconstruction examples, from left to right in (b,c) are results by the networks for '0' to '5'. It can be seen that each kernel autoassociator would perform well just on its intra-class patterns.

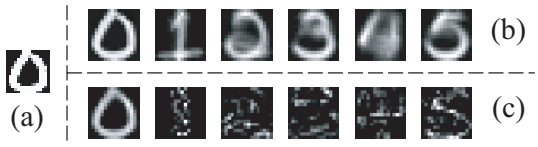

Figure 3. Pattern reconstruction by kernel autoassociators. (a) original pattern; (b) reconstructions by KAA-1s; (c) by KAA-2s.

Table 2 compares the kernel autoassociators with other techniques in USPS test. Here KDDA is a newly proposed kernel Fisher discriminant method [12], and KPCA-NN is the Nearest-Neighbour technique used in the KPCA subspace. Please note that no domain knowledge is used by any method in this study. The results suggest that kernel autoas-

sociators can offer satisfactory performance for handwritten digit recognition.

KAA-1	4.38%	KPCA-NN	6.15%
KAA-2	4.68%	AA-MLP	7.42%
KDDA	10.0%	SVM[11]	4.4%

Table 2. Recognition error rates on USPS.

7. Application to Face Recognition

The UMIST database consists of 575 gray-level face images of 20 subjects, each covering a wide range of poses [10]. The training set consists of 6 images per person, while test set is just the reminder. Both KPCA-NN and KAA-2 employ 40 principal components. Resulting recognition error are compared in Fig. 4, over the bandwidth σ of Gaussian kernel (Note that 1-Nearest-Neighbour (NN) and AA-MLP do not use Gaussian kernel). The performance of kernel autoassociators in the test are favourably compared to others, just slightly outperformed by KDDA (though they significantly outperformed KDDA in the previous experiment). Remarkably, KAA-1 demonstrates consistent performance over a wide range of σ.

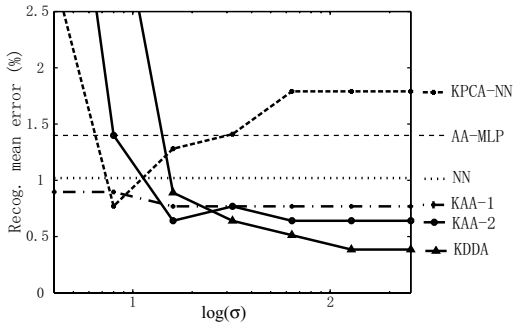

Figure 4. Face recognition results

8. Efficiency Consideration

Since kernel autoassociators can be trained using linear mean square error algorithms, in the training stage they generally take an advantage in computational time over AA-MLPs with back-propagation program. In the test stage, the overall cost can be approximated by that of $O(M)$ kernel operations plus some manipulations in Eq. 6 ($O(DM)$ linear) or Eq. 12 ($O(DM^2)$ linear). (Remind that D is the dimension of output, M the number of examples). Comparing with AA-MLPs which do no use large number of hidden nodes, the computational cost of kernel autoassociators could be higher. But the performance comparison described earlier demonstrates that kernel autoassociators are more effective.

9. Conclusion

This paper has presented a kernel autoassociator model for nonlinear autoassociation, and the model is capable of exploring nonlinear dependency among the data in an efficient way, by taking advantage of kernel techniques. The kernel autoassociator model has been evaluated using artificial data, as well as OCR and face recognition benchmark databases. Positive results attest to the excellent performance of the proposed model for visual classification.

References

[1] S. Haykin. An Introduction to Neural Networks – A Comprehensive Foundation. Prentice Hall, (1999)

[2] P., McLeod, K. Plunkett and E.T. Rolls. . An Introduction to Connectionist Modelling of Cognitive Processes, New York, Oxford University Press, (1998)

[3] T. Kohonen. Content-Addressable Memories. Springer-Verlag, (1980)

[4] B.L. Zhang, H.H. Zhang and S. Ge. Face recognition by applying wavelet subband representation and kernel associative memory. IEEE Trans. on Neural Network, 1(15), 166–177, (2004)

[5] P. Baldi, K. Hornik. Neural networks and principal component analysis: learning from examples without local minima, Neural Networks, 2 (1989) 53-58.

[6] G.W. Cottrell, P. Munro and D. Zipser. Learning internal representations of gray scale images: An example of extensional programming. Proc. 9th Annu. Cognitive Sci. Soc. Conf., Hillsdale, (1987) 462–473.

[7] H. Schwenk and M. Milgram. Transformation invariant autoassociation with application to handwritten character recognition. Neural Information Processing System (1995) 991-998.

[8] B. Schölkopf and A. Smola. Learning with Kernels. MIT Press, Cambridge, MA (2002).

[9] B. Schölkopf, A. Smola and K.-R. Müller. Nonlinear component analysis as a kernel eigenvalue problem. Neural Computation, 10 (1998) 1299-1319

[10] D.B. Graham and N.M. Allinson. Characterizing virtual eigensignatures for general purpose face recognition. Face Recognition: From Theory to Applications. (1998) 446-456

[11] Y. LeCun, L.D. Jackel *et al.* Comparison of learning algorithms for handwritten digit recognition. Proc. Int'l Conf. Artificial Neural Networks, Paris (1995) 53-60.

[12] L. Juwei, K.N. Plataniotis and A.N. Venetsanopoulos. Face recognition using kernel direct discriminant analysis algorithms. IEEE Trans. on Neural Networks, Vol. 14, No.1, (2003) 117-126.