Available online at www.sciencedirect.com
SCIENCE DIRECT®
Neurocomputing 61 (2004) 57–70
NEUROCOMPUTING
www.elsevier.com/locate/neucom

Nonlinear principal component analysis to preserve the order of principal components

Ryo Saegusa[a,*], Hitoshi Sakano[b], Shuji Hashimoto[a]

[a] *Department of Applied Physics, Waseda University, 3-4-1 Okubo, Shinjuku-ku, Tokyo 169-8555, Japan*
[b] *NTT Data Corporation, Kayaba-cho Tower 1-21-2, Shinkawa, Chuo-ku, Tokyo 104-0033, Japan*

Abstract

Principal component analysis (PCA) is an effective method of linear dimensional reduction. Because of its simplicity in theory and implementation, it is often used for analyses in various disciplines. However, because of its linearity, PCA is not always suitable, and has redundancy in expressing data. To overcome this problem, some nonlinear PCA methods have been proposed. However, most of these methods have drawbacks, such that the number of principal components must be predetermined, and also the order of the generated principal components is not explicitly given. In this paper, we propose a nonlinear PCA algorithm that nonlinearly transforms data into principal components, and at the same time, preserving the order of the principal components, and we also propose a hierarchical neural network model to perform the algorithm. Moreover, our method does not need to know the number of principal components in advance. The effectiveness of the proposed model will be shown through experiments.

Keywords: Nonlinear principal component analysis; Hierarchical structure; Sand-glass type multi-layered perceptron

1. Introduction

In the field of data analysis, it is important to reduce the dimensionality of data, because it will help to understand data and decreases the computational cost. As a method of dimensionality reduction, principal component analysis (PCA) [5] is often used in various areas such as pattern recognition and image processing [2,9].

* Corresponding author.
E-mail address: ryo@shalab.phys.waseda.ac.jp (R. Saegusa).

0925-2312/$ - see front matter
doi:10.1016/j.neucom.2004.03.004

PCA is an orthogonal transformation of a coordinate system in which we describe data. A basis of the objective coordinate system efficiently represents data distributed on a linear hyper plane as the coordinate value that is called a principal component. However, when n-dimensional data are distributed on a m ($< n$)-dimensional nonlinear manifold in a n-dimensional Euclidean space, more than the m dimensionality is required to describe the data in PCA which makes the dimensionality reduction inefficient.

In order to solve these problems, some methods of nonlinear principal component analysis (NLPCA) have been developed [1,3,7]. These methods can be classified into three categories. One is the application of a sandglass-type multi-layered perceptron (MLP) proposed by Irie and Kawato [6]. The other method is based of a partial linear approximation by Hastie et al.[4]. In these methods, the number of principal components has to be specified in advance. Unfortunately, these methods do not provide a way for deciding the number of principal components aside from inefficient trial and error. Another drawback of these methods is, even after the adequate number of principal component is given, they do not provide a parameter that corresponds to the eigenvalue in the conventional PCA, consequently, the ratio of contribution of the respective principal component cannot be explicitly determined. These drawbacks will limit the use of these methods in real world problems.

The third method is Kernel PCA recently proposed by Schölkopf [10]. Kernel PCA is a method that executes the linear PCA algorithm for the image of input data mapped by a nonlinear mapping function. Consequently, the method constructs the ordered principal components. However, the adequate way to determine the nonlinear mapping function for a given data set is not known. Moreover, the method has to solve an eigen-equation for a covariance matrix of the image of data in order to obtain an eigen base. The method also requires calculations of kernel functions between an objective data and all training data to calculate a principal component score.

In this paper, we propose a novel method of nonlinear principal component analysis that preserves the order of principal components based on their ratio of contributions [8]. Moreover, our method does not need to know the number of principal components in advance. In the proposed method, an array of neural networks are trained to build a set of nonlinear functions that map an input vector to its corresponding vector in the principal component space. The proposed model also has the ability to reconstruct the high dimensional data from its low dimensional representation in the principal component space. These functions are automatically adjusted in a training process. We also discuss the property of these functions by analyzing the experimental results. Section 2 of this paper describes the formulations of the proposed nonlinear PCA. Section 3 demonstrates the numerical experiments. In Section 4, we discuss the property of the proposed method. The conclusion and future studies are also given.

2. The formulation of nonlinear PCA

2.1. Extension of PCA to nonlinear PCA

Consider a random variable $\boldsymbol{x} \in R^n$ of $E[\boldsymbol{x}]=0$. In PCA, the feature (principal component) vector $\boldsymbol{y} \in R^m$ $(m \leqslant n)$ is an orthogonal transformation of data $\boldsymbol{x}$,

described by

$$\boldsymbol{y} = W^{\mathrm{T}}\boldsymbol{x} \tag{1}$$

$$= (\boldsymbol{e}_1^{\mathrm{T}}\boldsymbol{x}, \boldsymbol{e}_2^{\mathrm{T}}\boldsymbol{x}, \cdots, \boldsymbol{e}_m^{\mathrm{T}}\boldsymbol{x})^{\mathrm{T}}, \tag{2}$$

where columns of W are orthonormal bases $\{\boldsymbol{e}_i\}_{i=1,\ldots,m}$ that form an m-dimensional linear subspace L.

The reconstructed vector $\hat{\boldsymbol{x}} \in R^n$ from $\boldsymbol{y}$ is given as

$$\hat{\boldsymbol{x}} = W\boldsymbol{y} \tag{3}$$

$$= \sum_{i=0}^{m} \boldsymbol{e}_i(\boldsymbol{e}_i^{\mathrm{T}}\boldsymbol{x}). \tag{4}$$

$\{\boldsymbol{e}_i\}_{i=1,\ldots,m}$ are considered to minimize

$$E = E[\|\boldsymbol{x} - \hat{\boldsymbol{x}}\|^2].$$

The minimization is equivalent to the maximization of the variance of $\boldsymbol{y}$ [1].

In PCA, the mapping from the data space to the feature space is linear. The reverse mapping is also linear. We call the former a linear extraction function and the latter a linear reconstruction function.

The previously mentioned conventional PCA shows the best performance when the data $\boldsymbol{x}$ are distributed on an m-dimensional hyper plane. When the data are distributed on an m-dimensional nonlinear manifold embedded in an n-dimensional Euclidean space such as a curved hyper surface, the nonlinear PCA to provide nonlinear mapping functions should be introduced for non-redundant dimensionality reduction in order to represent the data with a curvilinear coordinate system.

We define the nonlinear extraction function from data $\boldsymbol{x}$ onto the feature vector $\boldsymbol{y}$ as

$$\boldsymbol{y} = \boldsymbol{\phi}(\boldsymbol{x}), \quad \boldsymbol{\phi} \in S_{\mathrm{e}}, \tag{5}$$

and a nonlinear reconstruction function from $\boldsymbol{y}$ onto the reconstructed vector $\hat{\boldsymbol{x}}$ as

$$\hat{\boldsymbol{x}} = \boldsymbol{\psi}(\boldsymbol{y}), \quad \boldsymbol{\psi} \in S_{\mathrm{r}}, \tag{6}$$

where S_{e} and S_{r} are the sets of nonlinear functions. The data nonlinearly correspond to the principal components through the nonlinear extraction function and the nonlinear reconstruction function.

Our problem is to minimize the mean square reconstruction error:

$$E = E[\|\boldsymbol{x} - \hat{\boldsymbol{x}}\|^2] \tag{7}$$

$$= E[\|\boldsymbol{x} - \boldsymbol{\psi}(\boldsymbol{\phi}(\boldsymbol{x}))\|^2]. \tag{8}$$

When we find the optimal $\boldsymbol{\psi}$, and $\boldsymbol{\phi}$, we can efficiently describe the data with fewer principal components than that of PCA.

2.2. The order of nonlinear principal components

In PCA, a reconstruction function is a linear combination of the bases. The bases $\{\boldsymbol{e}_i\}_{i=1,\ldots,m}$ are obtained from the minimization of the mean square reconstruction error under the constraint that they are normalized $\|e_i\| = 1$.

After the bases were obtained, we generate the reconstruction function from a linear combination of the bases. We can design the number to combine the bases for a specific purpose, such as the high contribution ratio, the low cost of memory and the best recognition rate. Since the bases are combined in the order of significance, the description efficiency is maximized for the number of principal components to be used.

In our method, we introduce nonlinear reconstruction functions that map a vector in the principal component space into the original input space as follows:

$$\hat{\boldsymbol{x}}_1 = \boldsymbol{\psi}_1(y_1), \tag{9}$$

$$\hat{\boldsymbol{x}}_2 = \boldsymbol{\psi}_2(y_1, y_2), \tag{10}$$

$$\vdots \tag{11}$$

$$\hat{\boldsymbol{x}}_m = \boldsymbol{\psi}_m(y_1, \ldots, y_m), \tag{12}$$

where y_i represents the score of the input vector with respect to the ith principal component, and $\hat{\boldsymbol{x}}_i$ is the n- dimensional vector reconstructed by the ith reconstruction function utilizing the representation of the original input vector $\boldsymbol{x}$ in the i-dimensional principal component space.

The ith component of the extraction function ϕ_i and the ith reconstruction function $\boldsymbol{\psi}_i$ are paired in the following manner:

$$\boldsymbol{\psi}_1(\boldsymbol{x}) = \boldsymbol{\psi}_1(\phi_1(\boldsymbol{x})), \tag{13}$$

$$\boldsymbol{\psi}_2(\boldsymbol{x}) = \boldsymbol{\psi}_2(y_1, \phi_2(\boldsymbol{x})), \tag{14}$$

$$\vdots \tag{15}$$

$$\boldsymbol{\psi}_m(\boldsymbol{x}) = \boldsymbol{\psi}_m(y_1, \ldots, y_{m-1}, \phi_m(\boldsymbol{x})). \tag{16}$$

The functions of each pair are adjusted to minimize the mean square reconstruction error in the above order. Therefore, y_i can be regarded as the ith significant nonlinear principal component. In order to obtain the extraction functions $\{\phi_i\}_{i=1,\cdots,m}$ and the reconstruction functions $\{\boldsymbol{\psi}_i\}_{i=1,\cdots,m}$, we propose a hierarchical nonlinear principal component network (HNPCN) composed of MLPs that are hierarchically arranged.

In the proposed method, we define the order of the principal components as the order to combine the principal components. The proposed method adjusts the parameters of the kth nonlinear extraction function ϕ_k to perform the best expressor combined with the upper nonlinear extraction functions $\phi_1, \phi_2, \ldots, \phi_{k-1}$. Consequently, when we choose a nonlinear extraction function from $\phi_k, \phi_{k+1}, \ldots, \phi_m$ for the function combining with $\phi_1, \phi_2, \ldots, \phi_{k-1}$, the expression ability of the kth function ϕ_k is the best function,

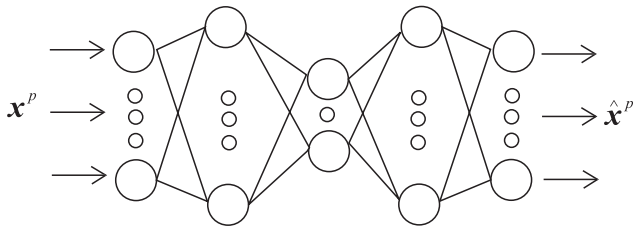

Fig. 1. A sandglass-type MLP.

while the lower functions $\phi_{k+1}, \phi_{k+2}, \ldots, \phi_m$ are equal or less significant from ϕ_k. The order of $\phi_1, \phi_2, \ldots, \phi_k, \ldots, \phi_m$ corresponds to the best order to combine the principal components for data expression.

2.3. The hierarchical nonlinear principal component network

The most sandglass-type MLPs for NLPCA are composed of five layers as shown in Fig. 1. The first and the fifth layers are the input and output layers, respectively. It is expected that principal components will be extracted in the third layer provided that the number of units in the third layer is less than the number of units in the first layer. The part from the first layer to the third layer has the role of data extraction, while the part from the third layer to the fifth layer has the role of data reconstruction. It should be noted that the number of principal components, which is the number of units in the third layer, must be determined before the training and there are no differences in the significance among the third layer units in this type of NLPCA.

We propose a HNPCN composed of a number of independent sub-networks that can extract ordered nonlinear principal components as explained in the previous section. The number of sub-networks corresponds to the number of principal components to be extracted. The number of layers in each sub-network is greater than five, and the number of input and output units are equally set to the size of the input vector's dimension, while the number of units in the middle layer (extraction layer) corresponds to the index of the principal component extracted by the corresponding sub-network. The structure of the model is shown in Fig. 2.

The activation function of the units in the input, output, and extraction layers is

$$f(u) = u, \tag{17}$$

while the activation function of the units in the other layers is

$$f(u) = \frac{1}{1 + \exp(-u/T)}. \tag{18}$$

Each sub-network is trained to reconstruct an input vector $\boldsymbol{x}$ in its output layer by producing $\hat{\boldsymbol{x}}$. As we want to extract the first principal component in the first sub-network, we set one unit in its extraction layer. The output of this unit is the representation point in the first principal component of that input. The extraction layer of the second sub-network receives the first principal component of the input vector computed in the

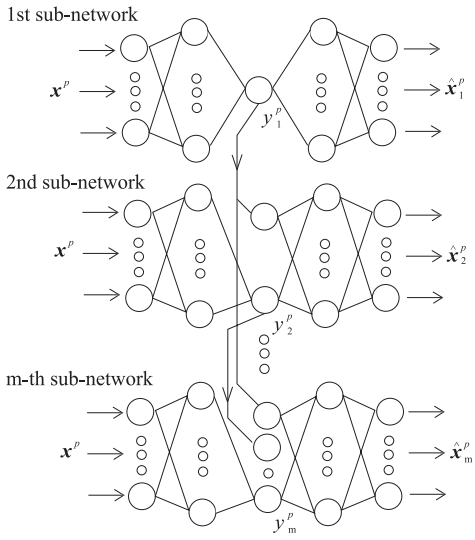

Fig. 2. Structure of HNPCN.

first sub-network and independently learns to generate a function that maps the input layer into the second principal component. This action is hierarchically executed in all of the sub-networks. It is obvious that the proposed model executes the nonlinear principal component mapping explained in the previous section.

When n-dimensional data $\boldsymbol{x}^p$ numbered p is given to the first layer of the ith sub-network, the ith unit in the extraction layer outputs one-dimensional data

$$y_i^p = \phi_i(\boldsymbol{x}^p) \in R^1, \tag{19}$$

where the function ϕ_i is the ith component of the extraction function from $\boldsymbol{x}^p$ onto a principal component y_i^p. ϕ_i corresponds to the extraction part that is from all units in the input layer to the ith unit of the extraction layer, while the principal components $y_1^p, y_2^p, \ldots, y_{i-1}^p$ from all the upper sub-networks are fed to this layer.

The function $\boldsymbol{\psi}_i$ is the ith reconstruction function from the principal components $y_1^p, y_2^p, \ldots, y_i^p$ onto the reconstructed data $\hat{\boldsymbol{x}}_i^p$. $\boldsymbol{\psi}_i$ corresponds to the reconstruction part that is from all units of the extraction layer to all units of the output layer.

The calculation of all outputs mentioned above is carried out in the increasing order of the sub-network number. Each connection weight of the ith sub-network is then adjusted in order to obtain an identical mapping by the criterion of the mean square error

$$E_i^p = \|\boldsymbol{x}^p - \hat{\boldsymbol{x}}_i^p\|^2. \tag{20}$$

As for the input data $\boldsymbol{x}^p$, the additional correction Δw_i^p of a weight coefficient w_i^p is

$$\Delta w_i^p = -\eta \frac{\partial E_i^p}{\partial w_i^p} \tag{21}$$

that is adjusted by a back propagation algorithm. η is a constant learning parameter.

The w_i^p of the ith sub-network is adjusted only inside the ith sub-network. Error signals of the ith sub-network are not propagated into the extraction part of all upper sub-networks. As a result, adjustment of the weight coefficients in the ith sub-network is dependent from the upper sub-networks to the lower ones. It is expected that this dependency forces one to construct the lower extraction function to be different from any of the upper $1, 2, \ldots, (i-1)$th functions and to be efficient to reconstruct the data.

3. Numerical experiments

We conducted some experiments to examine the efficiency of the proposed method. In this section we present two experimental results. In the first experiment, we utilize three-dimensional artificial data, the distribution of which is easy to understand. In the second experiment, we utilize high-dimensional data that correspond to a waveform.

3.1. Experiment with three-dimensional artificial data

Training data and test data are the coordinates of points on a parabolic surface in a three-dimensional Euclidean space given as follows:

$$x_3 = \frac{x_1^2}{a_1^2} + \frac{x_2^2}{a_2^2} \tag{22}$$

in which $(x_1, x_2, x_3) \in R^3$ and $a_1 = 1.0$, $a_2 = 3.0$.

The network in this experiment has two sub-networks. Each sub-network has five layers, where number of units in the first and fifth layers of each sub-network is three, and the number in the second and fourth layers is 10. The parameters for the network are set to $\eta = 0.05$ and $T = 0.1$. The number of training data is 20 000, randomly chosen on the parabolic surface. The number of test data is 400 from lattice points on the surface as shown in Fig. 3.

Fig. 4(a) shows the first principal component, generated from the first sub-network, while Fig. 4(b) shows the second principal component generated from the second sub-network without using the output of the unit in the third layer of the first sub-network.

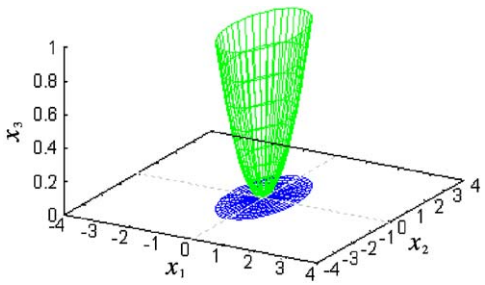

Fig. 3. Test data.

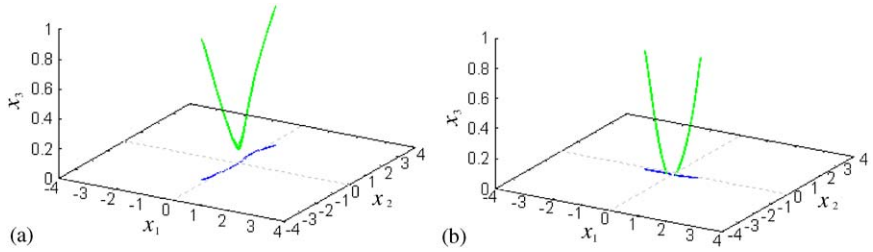

Fig. 4. The reconstructed data from the first principal component (a) and the second principal component (b).

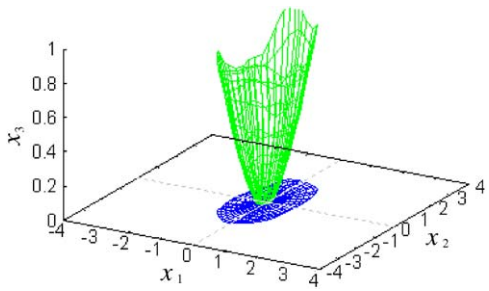

Fig. 5. Reconstructed data by the second sub-network.

From Eq. (22), the distribution of the data has the greatest variance along

$$x_3 = \frac{x_2^2}{3^2}, \tag{23}$$

which corresponds to Fig. 4(a) and the second greatest variance is along

$$x_3 = x_1^2, \tag{24}$$

which is finely reflected in Fig. 4(b).

Fig. 5 shows the reconstruction of the data from the second sub-network. It is obvious from this experiment that the proposed model is able to extract the nonlinear principal components from the nonlinearly distributed data.

Fig. 6 shows the principal component scores of the inputs. The horizontal axis indicates the first principal component score, while the vertical axis is for the second principal component score.

Next, we constructed three principal components with three sub-networks in the proposed network. The condition of the experiments is the same as that of the above experiment except for the number of sub-networks. Table 1 shows the mean square error (MSE) of each sub-network for 1000 test data points.

As shown in Table 1, the MSEs of the first, second, and third sub-networks were 0.18319, 0.00473, and 0.00257, respectively. It is considered that the reconstruction error is smaller when the number of available principal components is larger.

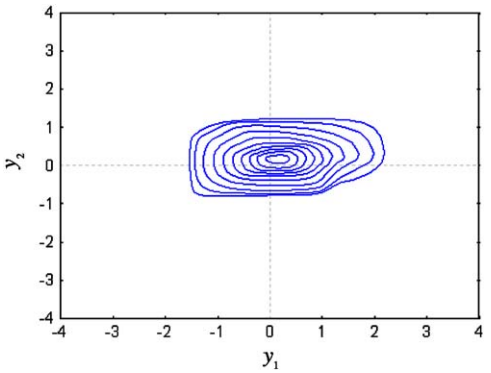

Fig. 6. A distribution of the first and second principal component.

Table 1
MSEs of sub-networks

# Sub-network	MSE	Diminution of MSE
1st	0.18319	—
2nd	0.00473	0.17846
3rd	0.00257	0.00216

However, when we increased the number of a principal component from one to two, the MSE decreased by 0.17846, but when we increased the number of principal components from two to three, the MSE decreased by the small value of 0.00216. Since the MSE is not much improved by the addition of the third principal component, we can consider that it is sufficient to describe the data in this experiment by two principal components, and the upper principal components have a higher ability to describe the data than the lower components.

We also conducted the experiments with the above parabolic surface data embedded in four, five, and six dimensional space. In these high-dimensional cases, the network could almost reconstruct the input data from two principal components as well as the three-dimensional case.

3.2. Experiment with waveform data

In this experiment, the input is n-dimensional vectors, whose components are periodically sampled points of function that is a superposition of three sinusoidal functions. The input vector can be written as follows:

$$(x_1, x_2, \ldots, x_n) = (f(\tau + \theta), f(2\tau + \theta), \ldots, f(n\tau + \theta)), \tag{25}$$

$$f(s\tau + \theta) = \sum_{k=1}^{3} a_k \sin(\omega_k(s\tau + \theta)) \quad s = 1, 2, \ldots, n, \tag{26}$$

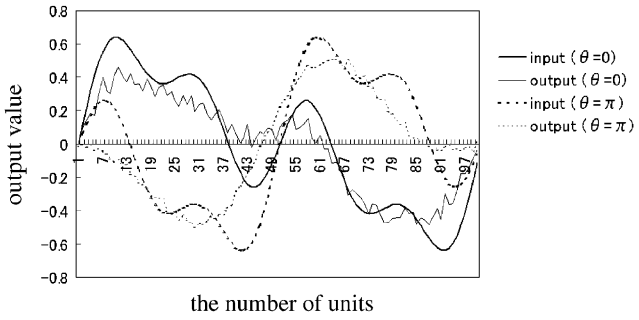

Fig. 7. Output wave of the first sub-network.

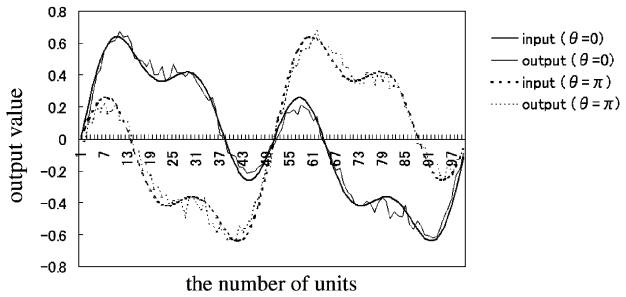

Fig. 8. Output wave of the fifth sub-network.

where the number of sampled points $n = 100$, the frequencies $(\omega_1, \omega_2, \omega_3)$=(1.0, 2.0, 3.0), the amplitudes $(a_1, a_2, a_3) = (0.5, 0.3, 0.2)$, and $\tau = 2\pi/100$. The initial phases θ of the training data are at random. The number of training data is 50,000. The number of test data is 100. The number of sub-networks is five and the number of layers is seven through out this experiment. We enhanced the description ability of the network by increasing one layer in the extraction part and the reconstruction part in each sub-network. The number of units in the first and seventh layers is 100, and the number in the second, third, fifth and sixth layers is 200. The parameters are set to $\eta = 0.001$ and $T = 0.1$.

In Figs. 7 and 8, we show the reconstructed data with the initial phases of $\theta = 0$ and π. The data are reconstructed from the test data by the trained network.

It is clear from Figs. 7 and 8 that the reconstructed wave of the fifth sub-network that used five principal components is better than that of the first sub-network that used only one principal component. These results verify that although we do not have parameters that correspond to the eigen value as in the conventional PCA, each sub-network is able to extract one principal component while keeping their order. In Fig. 7, the reconstructed data of the first sub-network are similar to a sinusoidal wave. The first

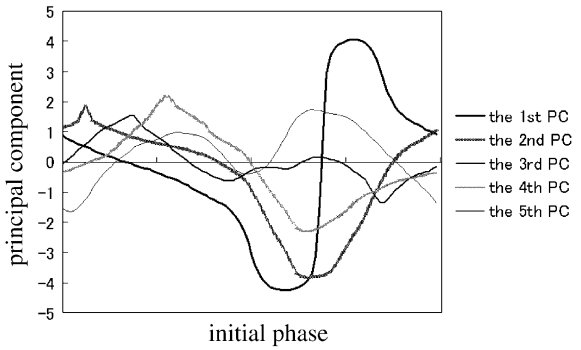

Fig. 9. The principal components as to the initial phases.

extraction function seems to obtain the lowest-frequency sinusoidal function of f. The first extraction function works most efficiently to reconstruct the objective waveform because the sinusoidal function has the largest amplitude.

Fig. 9 shows the form of the respective principal components. The figure shows that the first principal component is a low frequency component, and the frequency gradually grows bigger in the latter principal components. This result is comparable to the Fourier Series, but the proposed model has more independence of expressing the principal components, therefore, it can be expected that the proposed method will have a more efficient principal component representation of some signals that cannot be efficiently expressed by Fourier Series, i.e., signals with discontinuity.

4. Discussion and conclusions

We proposed a method of nonlinear principal component analysis that preserves the order of principal components with the hierarchical neural network model composed of a number of MLPs. In some numerical experiments, we demonstrated that the proposed network constructs the extraction functions in order of the reconstruction efficiency as to the objective data. We also confirmed that the network obtains the effective and significant functions for high-dimensional data.

The proposed model has a high independence in generating a function (extraction function) to map input vectors into their corresponding vectors in the principal component space, while at the same time, the model is also able to build a reconstruction function that maps the points in the principal component space into the original input space. The forms of the extraction and the reconstruction functions are determined by a number of factors, such as the structure of the sub-networks, the initial condition of each sub-network and the learning algorithm.

We also examined the robustness of the proposed model with respect to the initial condition of the sub-networks with different structures. This is done with the same data as in Section 3.1.

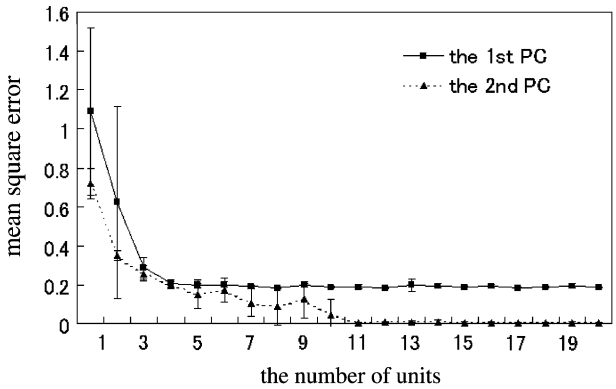

Fig. 10. The mean square errors with various numbers of units in the second and fourth layers.

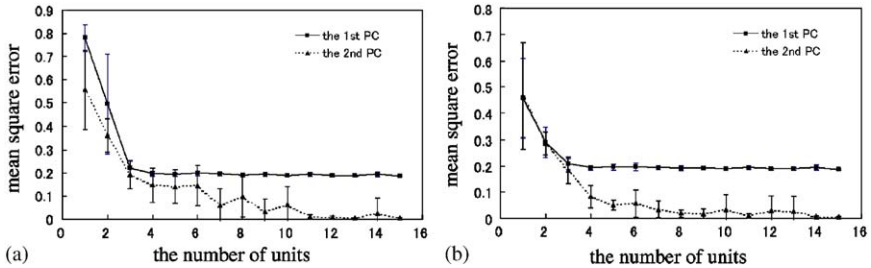

Fig. 11. The mean square errors with various numbers of units in the second layer by fixing the number in the fourth layer at 15 (a), and the mean square errors with various numbers of units in the fourth layer by fixing the number in the second layer at 15 (b).

Fig. 10 shows the learning MSE of the model, regarding sub-networks with a different number of units in the second and the fourth layers. Sub-networks with a greater number of units in the second and the fourth layers have better mapping abilities.

We also examined the performance of the proposed model by setting different numbers of units for the second and fourth layers. In Fig. 11(a), the number of units in the fourth layer is fixed at 15, while the number of units in the second layer varies. On the other hand, in Fig. 11(b), the number of units in the second layer is fixed at 15, while the number of units in the fourth layer varies. From Fig. 11, we can draw the conclusion that the extraction function is valued more than the reconstruction function in this mode. The reason is that a "weak" extraction function will fail to provide a good representation of input vectors in the principal component space, so the reconstructed vectors will not be precise even if the reconstruction function's performance is strong.

Although we do not have any explicit parameter that corresponds to the ratio of the contributions as in the conventional PCA, the learning error of each sub-network can be loosely used as a guidance in determining the cumulative ratio of contribution from all the sub-networks.

In regard to the constructed nonlinear principal components, the limitation of the data description ability is considered to depend on the structure of the sub-networks in the proposed model. If the structure is changed, the maximum description ability of the principal components will change.

In the proposed method, the sub-networks are trained until the mean square error converges so that the obtained nonlinear principal components are expected to have the best performance in the given network structure.

The promising applications that we will consider in the near future, encompasses nonlinear pattern classifications, nonlinear data compressions, etc.

Acknowledgements

We would like to thank Dr. Pitoyo Hartono, Mr. Hiroki Yoshizawa and Dr. Kenji Suzuki for their valuable advices.

This work was supported in part by The 21st Century Center Of Excellence Program, "The innovative research on symbiosis technologies for human and robots in the elderly dominated society", Waseda University, in part by the JSPS Research for the Future Program in the Area of Kansei (Intuitive & Affective) Human Interfaces and in part by a grant-in-aid from the Japanese Ministry of Education, Science, Sports and Culture (A-11305021).

References

[1] K.I. Diamantaras, S.Y. Kung, Principal Component Neural Networks Theory and Applications, Wiley, New York, 1996.

[2] R. Duda, P. Hart, Pattern Classification Theory and Systems, Springer, Berlin, 1988.

[3] R. Gnanadesikan, Methods for Statistical Data Analysis of Multivariate Observations, Wiley, New York, 1977.

[4] T. Hastie, W. Stuetzle, Principal curves, J. Am. Statist. Assoc. 84 (406) (1989) 502–516.

[5] H. Hotelling, Analysis of complex statistical variables into principal components, J. Educ. Psychol. 24 (1933) 498–520.

[6] B. Irie, M. Kawato, Acquisition of internal representation by multi-layered perceptrons, J. IEICE J73-D2 (8) (1990) 1173–1178 (in Japanese).

[7] J. Karhunen, J. Joutsensalo, Generalization of principal component analysis, optimization problems, and neural network, Neural Networks 8 (4) (1995) 549–562.

[8] R. Saegusa, S. Hashimoto, Nonlinear principal component analysis using neural networks, Proceedings of the 64th IPSJ General Conference, Vol. 2, 2002, pp. 205–206 (in Japanese).

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

[10] B. Schölkopf, A. Smola, K. Müller, Nonlinear component analysis as a kernel eigenvalue problem, Neural Comput. 10 (5) (1998) 1299–1319.

Ryo Saegusa received his B.Eng. and M.Eng. degrees in Applied Physics from Waseda University, Tokyo, Japan, in 1999 and 2001, respectively. He was a Ph.D. student in the Graduate School of Science and Engineering, Waseda University, from 2001 to 2004. Since 2004, he has been a research associate at the Department of Applied Physics of Waseda University. His research interests include neural computing, data analysis and image processing.

Hitoshi Sakano received his B.S. degree in Physics from Chuo University, Tokyo, Japan, in 1988 and his M.S. degree in Physics from Saitama University, Saitama, Japan, in 1990. Since 1990, he has been at NTT Data Co., Tokyo, Japan. His research interests include pattern recognition such as character recognition and biometrics. Since 2000, he is a visiting researcher at the Advanced Research Institute for Science and Engineering, Waseda University, Tokyo, Japan.

Shuji Hashimoto (Member, IEEE) received his B.S., M.S. and Dr.Eng. degrees in Applied Physics from Waseda University, Tokyo, Japan, in 1970, 1973 and 1977, respectively. He is currently a Professor in the Department of Applied Physics, School of Science and Engineering, Waseda University. Since 2000, he has been a director at the Humanoid Robotics Institute, Waseda University. From 1979 to 1991, he was with the Faculty of Science, Toho University, Funabashi. He was a Vice President of the International Computer Music Association until 2001. His research interests are in human communication and Kansei information processing including image processing, music systems, neural computing and humanoid robotics. Since 2003, he serves as the chair of the chapter promotion committee, IEEE Tokyo Section.