Proceedings of IEEE International Conference on Neural Networks 1994, Invited Paper, June , 26-July 2, Orlando, Florida, Vol.II, pp1252-1257.

Theories for Unsupervised Learning: PCA and Its Nonlinear Extensions

(Invited Paper)

Lei Xu

The Chinese University of Hong Kong and Peking University

Abstract— *Several theories are proposed for unsupervised learning in one layer nonlinear network. It has been shown that all the learning rules developed under the theories merge at performing PCA type tasks when the network reduces into linear one. However, for nonlinear networks the performances of these rules become different, which indicates many possibilities for nonlinear extensions of PCA. These theories provide a number of potential guidelines for further explorations on nonlinear PCA type learning. Moreover, the relations between these proposed theories as well as to some existing theories have also been discussed.*

I. Introduction

Since Oja's pioneer work on Principal Component Analysis by a single neuron[18], a large volume of papers have been published on PCA networks in the literature[1]. The developments can be roughly summarized into five directions: (1) to a layer of many neurons, including various asymmetrical and symmetrical architectures and algorithms for true PCA and subspace analyses [34] [6][16][17][7] [8][40] [9][29][30][14][15]; (2) to constrained PCA[13][26][10]; (3) to MCA (Minor Component Analysis)[36][37][23][33]; (4) to robust PCA [36] [37] [38] [33]; (5) to nonlinear networks [21] [35] [36][37][39] [31] [25] [24] [32] [12] [11] [27]; as well as a great number of applications.

The development from the single neuron to a layer of many neurons encountered two crucial problems. The first problem is how the learning rule for single neuron can be extended to let different neurons in the layer to response different features orthogonally such that the network can perform true PCA. One type of efforts is to externally well design some asymmetrical network structure plus lateral interaction[29][30]. Although these efforts solve the problem, the asymmetrical structure needs externally helps and non-parallel implementation, as well as has the weak point of increasing error accumulation in computation. The other possibility is to use a symmetrical or homogeneous network instead. Oja (1989) proposed a generalization of his single neuron constrained Hebbian rule for such symmetrical single layer linear network. However, only a partial success has been achieved: the network can only perform principal subspace analysis but can not separate the component eigenvectors that span the subspace[20]. The second problem is whether we can find some global principles that guide the unsupervised learning such that PCA emerges automatically instead of basing on heuristically proposed local Hebbian-type rules. There are some such principles for a single neuron[16][17][8][40][36], but seldomly see such principle for a layer of neurons, especially for Oja subspace learning on the symmetrical structure.

At the beginning of 1990, Oja believed that one growing point for studies on PCA-type networks should be the extensions to nonlinear networks. Sharing his opinion and also motivated to find a global interpretation for Oja subspace rule, I also fallen the interest of studying PCA-type networks[2]. In the beginning of 1991, Oja published two papers which explicitly proposed his views for extending PCA nets to nonlinear cases[21][22]. Paper [21] proposed a number of nonlinear constrained Hebbian learning rules and also demonstrated through experiments that the introduction of nonlinearity can let the learning resist very strong noises or outerliers. In the same time, Oja found that in one our joint work on a single neuron for Minor Component Analysis (MCA) and curve fitting (The work was later published in [36]), the success can also be explained as that some nonlinear factor has been introduced into the learning although the derivation of learning role was based on an other interpretation.

In May 1991, another work was completed by the present author[35]. This work proposed the LMSER(Least Mean Square Error Reconstruction) principle[3] for self-organizing nets and particularly studied the properties of LMSER for a single layer net $\vec{z} = S(\vec{y}), \vec{y} = W^t\vec{x}$ with $S(\vec{y}) = [s(y_1), \cdots, s(y_m)]^t$ and $s(r)$ being a nonlinear sigmoid function. Several interesting results were been obtained. For the linear case $s(r) = r$, it has be proved that by the global guide of LMSER the PCA automatically emerges for $m = 1$ and that the principal subspace analysis PCA automatically emerges for $m > 1$; it has also been proved that the evolution direction of Oja rule (for both single neuron and subspace) has a positive projection on that of LMSER and thus two rules have the similar convergence properties and the same solutions. This result provides a global convergence analysis for Oja rule, which was believed to be a difficulty task[10]. For the nonlinear cases that $s(r)$ is a sigmoid function, it was clearly pointed out and also demonstrated through experiments that the introducing of the nonlinearity can automatically break the symmetry of the homogeneous networks and let the weight vectors separate the different features (i.e., it seems the network performs the true PCA). Unfortunately, due to mathematical difficulties I could not prove this viewpoint. Later in [39], in help of Brockett's result[7], I only proved that a modified LMSER as well as a modified Oja subspace rule performs the true PCA in the semi-asymmetrical linear case of $S(\vec{y}) = [a_1y_1, \cdots, a_my_m)]^t$ with $a_1, \cdots a_m$ being different positive numbers. A few months after the work [35], Oja proposed his weighted subspace rule[25] which performs the true PCA for linear networks. This rule, the Brockett rule [7], the modified LMSER and the modified Oja subspace rule given in [39] are closely related, but different from each other, as well as derived from different aspects. Motivated by the finding of [35] on the function of nonlinearity, Oja also demonstrated in [25] through experiments that a nonlinear subspace algorithm, obtained by introducing sigmoid nonlinearity to his weighted subspace rule, can also separate the weight vectors to the different feature directions and perform

The author's address: Dept. of Computer Science, HSH ENG BLDG, Room 1006, The Chinese University of Hong Kong, Shatin, New Territories, Hong Kong

[1] A long reference list is given in [39]; and a much more detailed reference list is provided recently by Oja's research group.

[2] During 1989.2-1990.5, I was worked at his laboratory as a visiting senior researcher. It is this visit and the discussions with him that arose my interest and gave me many insights on PCA-type networks.

[3] Its differences from the related works [34] [3] [6] are given in [39], and will be further discussed later in Section VI.

0-7803-1901-X/94 $4.00 ©1994 IEEE

similarly to his weighted subspace rule. Later in [24], Oja has further theoretically proved that under certain conditions the nonlinear PCA rules given in [35] [25] can indeed perform the true PCA. No doubt, it is an important result.

Recently, the research on nonlinear PCA has gained an increasing interest in the literature. Some intersting results have also been reported on using nonlinear PCA for signal separation and image [11][12][33]. In WCNN-1993, a self-association method similar to the LMSER principle [35] [39] has been proposed by Palmieri [28]. The method applies the LMSER principle and the sigmoid nonlinear neurons to an asymmetrical architecture that resembles to those given in [29][13]. It consists of the Hebbian type rule that is the same as the modified oja rule proposed in [35] [39] and the anti-Hebbian type rules for lateral connections which is an essential difference from the methods by [35] [39]. A preliminary experimental result has also been given by [28] to demonstrate the potential of this approach in function decomposition. Moreover, there are also many other works which are closely related to nonlinear PCA [36][37][38][31][32].

The present paper will propose several theories as some global guiding principles for unsupervised learning in one layer nonlinear network. In the special case of linear network, we show that all the learning rules developed under the theories merge at performing automatically PCA or subspace analysis. However, for nonlinear networks the performances of these rules become different, and thus indicates that the extensions of PCA from linear cases to nonlinear cases may have many possibilities, and that the theories proposed in this paper provide for nonlinear PCA type learning a number of potential rules that may deserve further explorations. Moreover, the relations between these proposed theories as well as to some other existing theories have also been discussed.

II. Theories and Their Related Rules

A. *Quasi-Linear Networks*

We consider a quasi-linear networks as follows

$$\vec{z} = S(\vec{y}), \quad \vec{y} = W^t\vec{x}, \tag{1}$$

where $\vec{z} = [z_1, \cdots, z_m]^t$, $\vec{y} = [y_1, \cdots, y_m]^t$, $\vec{x} = [x_1, \cdots, x_n]^t$. Moreover, W^t is a $m \times n$ matrix with $W = [\vec{w}_1, \cdots, \vec{w}_m], \vec{w}_i = [w_{1i}, \cdots, w_{ni}]^t$; $(\vec{y}) = [s_1(y_1), \cdots, s_m(y_m)]^t$ with $s_i(r)$ being a nonlinear sigmoid function. So, this network consists of a linear system and a sigmoid nonlinear transform following the outputs of the linear system. Thus we call the network a quasi-linear networks. This is a slight different version of the network studied in (Xu, 1991, 1993) [35][39]

Particularly, when $s_i(x)$ is a linear function $s_r(r) = r$, eq.(1) becomes an one-layer linear network used in Oja subspace rule[20]. In the special case of $m = 1$, we return back to a single PCA neuron originally studied by (Oja, 1982)[18].

In this paper, we will base on the network eq.(1) to show our theories. It may deserve to mention that the theories and rules as well as the related analyses given in this paper may be able to be extended to various other existing PCA nets [29][30][27][13][10]; however, we would like to leave this work elsewhere.

B. *Best Reconstruction Theory and Min-Distorted Reflection Theory*

From the output of eq.(1) (i.e., $\vec{z}$), we let

$$\vec{u} = W\vec{z}, \tag{2}$$

as a reconstruction of input $\vec{x}$. Here, *Our theory for unsupervised learning on eq.(1) is that the reconstructed $\vec{u}$ should approximate $\vec{x}$ best.*

This theory can be realized by the LMSER (*Least Mean Square Error Reconstruction*) theory proposed by the present author in 1991 [35][39]. That is, we determine the parameters W in eq.(1) by minimizing[4]

$$e_2(\vec{x}, \vec{u}) = E\|\vec{x} - \vec{u}\|^2, \tag{3}$$

As shown in [35][39], using gradient descent technique, both *batch* way and *on line or adpative* way algorithms can be obtained to implement the minimization, and the resulted solutions are the same as Oja subspace rule when $s(.)$ is linear. It was firstly suggested and demonstrated through experiments by [35][39] that the measure $e_2(\vec{x}, \vec{u})$ can used in the network eq.(1) to break naturally its symmetrical structure and to separate the different features.

Let us further consider the pair $(\vec{x}, \vec{u})$. Imagining that the input layer and the output layer form two boundaries. Then $\vec{u}$ can be regarded as the reflection of $\vec{x}$ bounced back by the output boundary. Similarly, we can have $\vec{z}_1$ being the reflection of $\vec{z}$, bounced back by the input boundary, and $\vec{u}_2$ being the reflection $\vec{u}$, ... Denoting $\vec{z}$ by $\vec{z}_0$, and $\vec{u}$ by $\vec{u}_1$ and $\vec{x}$ by $\vec{u}_0$, for $i = 0, 1, 2, \cdots$, we can generally have

$$\vec{z}_{i+1} = S(W^tW\vec{z}_i), \vec{u}_{i+1} = WS(W^t\vec{u}_i), \tag{4}$$

Ideally, we hope that $\vec{u}_0 = \vec{u}_1 = \cdots \vec{u}_i = \cdots$ and $\vec{z}_0 = \vec{z}_1 = \cdots \vec{z}_i = \cdots$. But this is not true since $S(.)$ is nonlinear and W is not full rank. Instead, our theory is that W *is decided such that the distortions of reflections can be minimized.* For this purpose, we can minimize either of the following measures

$$d_1^u = \sum_0^k E\|\vec{u}_{i+1} - \vec{u}_i\|^2, \quad d_2^u = \sum_0^k E\|\vec{u}_{i+1} - \vec{x}\|^2, \tag{5}$$

The minimizations can be implemented by gradient descent approach. One can also observe easily that d_1^u, d_2^u reduce into $e_2(\vec{x}, \vec{u})$ when $k = 0$. In other words, eq.(3) is a speical case of eq.(5).

C. *Maximum Relative Uncertainty Theory*

A random variable $\vec{x}$ takes a value with uncertainty. The degree of this uncertainty depends on the distribution $P(\vec{x})$. For two different $P(\vec{x}), P(\vec{\xi})$, the degrees of uncertainty will also be not same. As given below, we can use different types of measures for describing the degrees.

First, when $P(\vec{x})$ is a normal, the uncertainty can be fully described by its covariance matrix $\Sigma_x = E[(\vec{x} - E\vec{x})(\vec{x} - E\vec{x})^t]$. Therefore, its determinant $J_d(\vec{x}) = det(\Sigma_x)$ or more generally

$$J_d^f(\vec{x}) = f(det(\Sigma_x)), \tag{6}$$

can be used as the measeure for describing the degree of uncertainty, where $f(r) \geq 0$ is any monotonuously increasing differentiable function for $r \geq 0$, e.g., $f(r) = r, r^2, r^p (p \geq 1), ln(r), \ldots$ The measure describes how $P(\vec{x})$ scattering. The larger the measure J_d^f is, the larger the scattered range, and the larger the uncertainty. Thus, even when $P(\vec{x})$ is not a normal distribution, we can still use the measure eq.(6) for describing its related uncertainty.

[4] It may also be written as an apparently more general form $e_2^f(\vec{x}, \vec{u}) = f(E\|\vec{x} - \vec{u}\|^2)$, where $f(r)$ is an arbitrary monotonuously increasing differentiable function for $r \geq 0$. However, it is obvious that the solutions for minimizing $e_2(\vec{x}, \vec{u})$ and $e_2^f(\vec{x}, \vec{u})$ are the same. In this paper, we will not discuss this trivial extension. However, all our results apply to such extension.

Second, we can use the entropy

$$J_e(\vec{x}) = -E\ln p(\vec{x}) = -\int p(\vec{x})\ln p(\vec{x})d\vec{x}$$

or more generally

$$J_e^f(\vec{x}) = f(-E\ln p(\vec{x})), \tag{7}$$

as a uncertainty measure, where $f(r)$ is the same as in eq.(6).

With the above measures, for two distributions $P(\vec{x}), P(\vec{\xi})$ we further call $\rho_d(\vec{z},\vec{\eta}) = J_d^f(\vec{x})/J_d^g(\vec{\xi})$, $\rho_e(\vec{x},\vec{\xi}) = J_e^f(\vec{x}) - J_e^f(\vec{\xi})$ as the *relative uncertainty measures* of $P(\vec{x})$ to $P(\vec{\xi})$, with $g(r) \neq f(r), g() \geq 0$ being monotonuously increasing and differentiable for $r \geq 0$. The measures indicate how scattering or uncertain of $\vec{x}$ in comparison with $\vec{\xi}$.

Through the network eq.(1) $\vec{x}$ will become the transformed variable $\vec{z}$ with $P(\vec{z})$. Let us also consider another variable $\vec{\xi}$ of the same dimension as $\vec{x}$. Assume that $\vec{\xi}$ comes from the standard normal distribution $N(\vec{0}, I)$, then $\vec{\xi}$ will become $\vec{\eta}$ with $P(\vec{\eta})$ after the network eq.(1). *Our theory here is that we maximize the relative uncertainty measure $\rho(\vec{z},\vec{\eta})$ to determine the parameters W in eq.(1)*, where $\rho(\vec{z},\vec{\eta})$ can be either of the follwoing two:

$$\rho_d(\vec{z},\vec{\eta}) = f(det(\Sigma_z))/g(det(\Sigma_\eta)), \tag{8}$$

$$\rho_e(\vec{z},\vec{\eta}) = f(-E\ln p(\vec{z})) - g(-E\ln p(\vec{\eta})). \tag{9}$$

Moreover, considering that f, g are arbitrary, we actually have two spectrums of different learning rules.

All the rules obatained under this theory can be implemented by the gradient ascent algorithm in the batch way, i.e., $W(k+1) = W(k) + \alpha\frac{d\rho(\vec{z},\vec{\eta})}{dW}$ with an appropraite learning stepsize $\alpha > 0$.

Finally, let us explain the motivation behind the theory. On one hand, the uncertainty of $\vec{x}$ reflects the complexity of its distribution $P(\vec{x})$ or the richness of information contained in $P(\vec{x})$, and we naturally hope that the our network's output can retain this complexity; thus we want to maximize the uncertainty of output $\vec{z}$. On the other hand, when we talk about how large or small of the uncertainty of $\vec{x}$, we actually compare it with some reference $\vec{\xi}$ implicitly in our mind. Through the network, the uncertainty of $\vec{\xi}$ may also change, and the uncertainty measure of output $\vec{z}$ may also contain this type of changes. So in eq.(8)(9), we use the uncertainty of $\vec{\xi}$ to re-calibrate our reference. Here, we take $\vec{\xi}$ from the standard normal distribution $N(\vec{0}, I)$ as our reference.

D. Other Theories

We also give some more heuristic theories. The first one is to determine the parameters W in eq.(1) by maximizing the followin measure:

$$J_c^o = tr(\Sigma_z\Sigma_\eta^{-1}), \tag{10}$$

where the notations are the same as in the previous subsection. It is motivated by the reason simliar to the above one. The uncertainties of $\vec{z}$, $\vec{\xi}$ are described by Σ_z, Σ_η, and and we use the uncertainty of reference ξ to discount the related part in the uncertainty of $\vec{z}$.

The second one is to maximize one of the following two measures:

$$\begin{aligned} J_d^o &= J_d^f(\vec{x}) + cg(|tr(\Sigma_\eta - I)|) \\ &= f(det(\Sigma_z)) + cg(|tr(\Sigma_\eta - I)|), \end{aligned} \tag{11}$$

$$\begin{aligned} J_e^o &= J_e^f(\vec{x}) + cg(|tr(\Sigma_\eta - I)|) \\ &= f(-E\ln p(\vec{z})) + cg(|tr(\Sigma_\eta - I)|), \end{aligned} \tag{12}$$

where $g(r) \neq f(r)$ is the same as in eq.(8)(9), and $c > 0$ is an arbitrary constant. The motivation is also related the one in the previous subsection. We want to maximize the uncertainty of $\vec{z}$, while in the sametime, and to keep the change of uncertainty from ξ to η being minimum.

One step further, we can simply constrain that the the uncertainty of the reference keeps invariable, i.e., $\Sigma_\eta = I$. This leads to the third choices–the constrained maximizations as follows

$$J_d^f = f(det(\Sigma_z)), \ \ s.t.\ \Sigma_\eta = I,, \tag{13}$$

$$J_e^f = f(-E\ln p(\vec{z})), \ \ s.t.\ \Sigma_\eta = I, \tag{14}$$

III. Analyses on the Linear Cases: Converged at PCA

When $s_i(r)$ is a linear function $s_i(r) = r$, eq.(1) becomes the one-layer linear network studied in [20]. In this case, $\vec{z} = \vec{y} = W^t\vec{x}$, and $\Sigma_z = W^t\Sigma_x W$, $\Sigma_\eta = W^tIW = W^tW$.

A. On the Rules for Best Reconstruction and Min-Distorted Reflection

In this linear case, the rules eq.(3) and eq.(5) become

$$\begin{aligned} e_2 &= E\|\vec{x} - WW^t\vec{x}\|^2 = E\|(I-P)\vec{x}\|^2, \\ d_1^u &= \sum_0^k E\|(I-P)\vec{u}^i\|^2 = \sum_0^k E\|(I-P)P^i\vec{x}\|^2, \\ d_2^u &= \sum_0^k E\|(I-P^{i+1})\vec{x}\|^2, \end{aligned} \tag{15}$$

where $P = WW^t$ is called as projection operator.

With some mathematical work, we can derive

$$\nabla_W e_2 = 2(W^t\Sigma_x - W^t\Sigma_x P + W^t\Sigma_x - PW^t\Sigma_x), \tag{16}$$

When $m < n$ and Σ_x is nonsingular. Let $\Phi = [\vec{\phi}_1, \cdots, \vec{\phi}_n]$ and $\Lambda = diag[\lambda_1, \cdots, \lambda_n]$ be the matrices of eigenvectors and eigenvalues of Σ_x respectively. As proved in [35][39], we have the solutions of $\nabla_W e_2 = 0$ are $W = \Phi\Pi DR$, R is an arbitrary $m \times m$ rotation matrix, i.e., $R^tR = I$. $D = [D_1|0]$ is $m \times n$ matrix with D_1 being an $m \times m$ diagonal matrix and its diagonal elements only taking value of $+1, -1, 0$. $\Pi_{n\times n}$ is an arbitrary permutation matrix. Furthermore, among these solutions, the one consisting of the eigenvectors that correspond to the m largest eigenvalues makes e_2 reach its minimum. In other words, the learning rule eq.(3) performs *Principal Subspace Analysis (PSA)*. Particularly, when $m = 1$, the weight vector of the single neuron will be the eigenvector corresponding to the largest eigenvalue, i.e., it performs PCA.

In the similar way, we can also derive $\nabla_W d_1^u$ and $\nabla_W d_2^u$, then via $\nabla_W d_1^u = 0$ and $\nabla_W d_2^u = 0$ prove that the minimizations of d_1^u, d_2^u also emerge PSA automatically. We will not give the detail mathematics here. One can also observe this point in such a way: from $W = \Phi\Pi DR$ we have $W^tW = I$, $P^i = P, i > 1$ and $(I-P)P = 0, P(I-P) = 0$, put these into eq.(15), we find that e_2, d_1^u, d_2^u become indentical.

B. On the Rules for Maximum Relative Uncertainty

In this linear case, the rule eq.(8) becomes

$$\rho_d(\vec{z},\vec{\eta}) = f(det(W^t\Sigma_x W))/g(det(W^tW)).$$

Throught some calculation, we can obtain

$$\begin{aligned} \nabla_W\rho_d = 2[&f'g\ det(W^t\Sigma_x W)\Sigma_x W(W^t\Sigma_x W)^{-1} \\ &-g'f\ det(W^tW)W(W^tW)^{-1}]/g^2, \end{aligned}$$

where $f = f(det(W^t\Sigma_x W))$, $f' = f'(det(W^t\Sigma_x W))$, $g =$

$g(det(W^tW))$ and $g' = g'(det(W^tW))$. From $\nabla_W \rho_d = 0$, we further have

$$\begin{aligned} \Sigma_x W &= W\Lambda,\ \Lambda = a(W^tW)^{-1}W^t\Sigma_x W, \\ a &= g'f det(W^tW)/[gf' det(W^t\Sigma_x W)]. \end{aligned} \quad (17)$$

Its solutions must satisfy $W^tW = b^2 I$ and $a = 1$ with b being a constant. Thus, by single value decomposition we have $W = b\Phi R$, where Φ is a $n \times m$ matrix $\Phi^t\Phi = I$ and R is $m \times m$ orthogonal mtarix. Putting this W in ρ_d, we get $\rho_d = f(b^2 det(\Phi^t\Sigma_x\Phi))/g(b^2))$. Since $f(r)$ is monotonuously increases, we see that it arrives its maximum when Φ consists of the eigenvectors that correspond to the m largest eigenvalues of Σ_x. Moreover, we can also show that all the other solution for Φ are saddle points for ρ_d. Next, we further check whether it is possible for $a = 1$. The second equation in eq.(17) is now reduced into

$$a(b) = g'(b^2)f(b^2\lambda_0)/[\lambda_0 g(b^2)f'(b^2\lambda_0)]$$

with $\lambda_0 = det(\Phi^t\Sigma_x\Phi) > 0$. $a(b)$ is continuous on $[0,\infty]$ since $f(r), g(r)$ are differentiable functions. So, there exists a b' such that $a(b') = 1$ as long as $a(b) - 1$ can change its sign on $[0,\infty]$. This is a very weak condition which is generally satisfied. In addition, we can also check that $a(b) = 1$ for $b = 1$ when $f(r) = g(r) = r^p$ with $p \geq 1$.

Therefore, we prove that the learning rule eq.(8) performs PSA. Particularly, when $m = 1$, the weight vector of the single neuron will be the eigenvector corresponding to the largest eigenvalue.

Let us further consider the rule eq.(9). Assume that $\vec{x}$ is from normal distribution, thus $\vec{z} = \vec{y} = W^t\vec{x}$ is also normal. In this case, after some calculation, eq.(9) will become

$$\rho_e(\vec{z}, \vec{\eta}) = f(0.5 \ln det(W^t\Sigma_x W) + h) - g(0.5 \ln det(W^tW) + h)$$

with h being a constant. It further follows that

$$\nabla_W \rho_e = f'\Sigma_x W(W^t\Sigma_x W)^{-1} - g'W(W^tW)^{-1}$$

with $f' = f(0.5 \ln det(W^t\Sigma_x W) + h)$ and $g' = g(0.5\ \ln det(W^tW) + h)$. From $\nabla_W \rho_e = 0$, we again obtain eq.(17), but now $a = g'/f'$. Using similar argments for the rule eq.(8), we see the learning rule eq.(9) perform PSA too.

C. On the Other Rules

In this linear case, the rule eq.(10) becomes

$$J_c^o = tr(W^t\Sigma_x W(W^tW)^{-1})$$

Via derivation, we have

$$\nabla_W J_c^o = 2[\Sigma_x W - W(W^tW)^{-1}W^t\Sigma_x W](W^tW)^{-1}$$

From $\nabla_W J_c^o = 0$, we again obtain eq.(17), but now $a = 1$. So, we see the learning rule eq.(10) performs PSA too.

The rule eq.(11) becomes

$$J_d^o = f(det(W^t\Sigma_x W)) + cg(|tr(W^tW - I))|,$$

and we have

$$\begin{aligned} \nabla_W J_d^o = {} & 2f' det(W^t\Sigma_x W)\Sigma_x W(W^t\Sigma_x W)^{-1} \\ & -2cg' sign(tr(I - W^tW))W, \end{aligned}$$

with $f' = f'(det(W^t\Sigma_x W))$, $g' = g'(|tr(W^tW - I))|$, and $sign(x) = 1, x \geq 0, sign(x) = -1, x < 0$.

From $\nabla_W \rho_d = 0$, we further have

$$\begin{aligned} & \Sigma_x W = W\Lambda,\ \Lambda = aW^t\Sigma_x W, \\ & a = cg' sign(tr(I - W^tW)/[f' det(W^t\Sigma_x W)]. \end{aligned} \quad (18)$$

Similar to the arguments for eq.(17), we now need to show $ab^2 = 1$ or

$$a(b) = [cg'(b^2)sign(n(1 - b^2))/[b^4 f'(b^2\lambda_0)\lambda_0] = 1/b^2$$

When $b < 1$, $a(b)$ is a positive continuous on $[0,1]$. So, as long as $a(b) - 1/b^2$ can change its sign on $[0,1]$, we can prove that the learning rule eq.(11) performs PSA by the arguments similar to those for eq.(17).

The rule eq.(12) becomes

$$J_e^o = f(0.5 \ln det(W^t\Sigma_x W) + h) + cg(|tr(W^tW - I))|$$

with h being a constant. It follows that

$$\nabla_W J_e^o = f'\Sigma_x W(W^t\Sigma_x W)^{-1} - 2cg' sign(tr(I - W^tW))W$$

with $f' = f'(0.5 \ln det(W^t\Sigma_x W) + h)$. The other notations are the same as above.

From $\nabla_W \rho_d = 0$, we again obtain eq.(18), but now $a = 2cg' sign(tr(I - W^tW))/f'$. Using similar argments for the rule eq.(18), we see the learning rule eq.(12) perform PSA too.

The rule eq.(13) becomes

$$J_d^f = f(det(W^t\Sigma_x W)),\quad s.t.\ W^tW = I,$$

By Lagrange law, the solution is give by

$$2f' det(W^t\Sigma_x W)\Sigma_x W(W^t\Sigma_x W)^{-1} - 2WD = 0$$

with $f' = f'(det(W^t\Sigma_x W))$ and D being a diagonal matrix. Let $D = I$, we again get eq.(18), but now $a = 1/[f' det(W^t\Sigma_x W)]$.

The rule eq.(14) becomes

$$J_e^f = f(0.5 \ln det(W^t\Sigma_x W) + h),\quad s.t.\ W^tW = I,$$

By Lagrange law, the solution is give by

$$f'\Sigma_x W(W^t\Sigma_x W)^{-1} - 2WD = 0$$

with $f' = f'(0.5 \ln det(W^t\Sigma_x W) + h)$ and D being a diagonal matrix. Let $D = I$, we get eq.(18) too, but now $a = 2/f'$. Using similar argments for the rule eq.(18), we see the learning rule eqs.(13)(14) also perform PSA.

IV. Nonlinear Cases: Several Viewpoints

Although the learning rules based each theory poposed in Section II perfom PCA or PSA tasks for the linear cases studied in Section III, the theoires and rules will function differently in general cases that $s_i(r)$ are sigmoid nonlinear functions $s_i(r) = r$. In addition, stricly speaking, they no longer perform PCA or PSA.

Let us observe the points in the simple special case of single neuron. Assume that $s(0) = 0, s(-r) = -s(r)$ and $\vec{x}$ from a normal distribution $N(0, \Sigma_x)$. Thus $E(z) = E(s(\vec{y})) = 0$ since $\vec{y}$ is from a normal $N(0, \vec{w}^{\,t}\Sigma_x\vec{w})$ too.

The rule eq.(3) becomes $e_2 = E||\vec{x} - \vec{w}s(\vec{w}^{\,t}\vec{x})||^2$ and $\nabla_W e_2 = 0$ is given by

$$E[\vec{x}s]\vec{x} - E[s^2]\vec{w} + E[s'\vec{x}^{\,t}\vec{x}] - E[s's\vec{w}^{\,t}\vec{x}]\vec{w} = 0, \quad (19)$$

where $s = s(\vec{w}^{\,t}\vec{x})$ and $s' = s'(\vec{w}^{\,t}\vec{x})$.

The rule eq.(8) becomes $\rho_d = \frac{f(E[s^2(\vec{w}^{\,t}\vec{x})])}{g(E[s^2(\vec{w}^{\,t}\vec{\xi})])}$, where ξ is a reference from $N(0, I)$. From $\nabla_W \rho_d = 0$, we have

$$f'E[s(\vec{w}^{\,t}\vec{x})s'(\vec{w}^{\,t}\vec{x})\vec{x}] - g'E[s(\vec{w}^{\,t}\vec{\xi})s'(\vec{w}^{\,t}\vec{\xi})\vec{\xi}]) = 0, \quad (20)$$

where $f' = f'(E[s^2(\vec{w}^{\,t}\vec{x})]), g = g'(E(s^2[\vec{w}^{\,t}\vec{\xi})])$.

Comparing the two equations eqs.(19)(20), one observes that generally their solution will not be the same due to the nonlinearity of $s(.)$. Even for the rule eq.(8) itself, the solutions will be different when f, g change. It follows from eq.(20) that its solutions actually are not longer the principle component vector in the common sense.

So, the so called *nonlinear PCA* really means "the extensions of PCA to nonlinear cases". As shown in previous sections, there can be many rules that perform PCA or PSA tasks in the linear case. In the linear case, the PCA or PSA has an unique and well-defined meaning. While in the nonlinear cases, all the rules will function differetly and thus the nonlinear PCA or PSA has no unique or well defined meaning. Vaguely, it means *a class of learning rules for nonlinear networks which functions somewhat similar to PCA or PSA and will reduce into PCA and PSA in linear cases.*

As shown in papers [35][39], the learning rule eq.(3) based on the theory given in section II.B can let the weight vectors in the nonlinear network eq.(1) separate the different feature directions. It was believed in [35][39] that the rule performs the true PCA. This opinion is actually not accurate. In fact, the rule performs some type of function which can be regarded as a good approximation to the true PCA. This is why the results of experiments in [35][39] demonstrate some derivations between the rule and the true PCA. There remain some important open problems to be explored. For examples, what advantages will be gained by using nonlinear PCA ? do they outperform the true PCA ? what kind of performance each of the rules given in this paper will produce ? which theory or rule is the best ?

Generally speaking, for all the theories and rules proposed in this paper, the implementation is actually a nonlinear optimization. Thus, gradient ascent or descent techniques can be used to tackle the problem. However, the computing costs will be different for different rules. For some rules, e.g., eq.(3) and eq.(10), we can also get adaptive or on-line as well as local algorithms [35][39]. Thus, the computing aspects of the proposed theories and rules are also deserve to be further explored.

V. Some Other Extensions

As shown in Section III, all the rules under the proposed theories perform PSA in the linear network with $s_i(r) = r$ in eq.(1), instead of the true PCA. As argued in [35][39], the true PCA has some advantages over PSA. It has been shown the the best reconstruction rule eq.(3) can be modified slightly to let the network perform the true PCA in the linear case that $s_i(r) = a_i r$ with any constants $a_1 \neq a_2 \neq \cdots \neq a_m$.

Here we show that for this special linear case the rule eq.(10) can also be modified to perform the true PCA. The key point is to let $\vec{x}$ pass the operator $S(.)$ while the reference $\vec{\xi}$ does not pass $S(.)$. That is $\vec{z} = S(\vec{y}) = A\vec{y}$, $\vec{y} = W^t\vec{x}$, $\vec{\eta} = W^t\vec{\xi}$, where $A = diag[a_1, \cdots, a_m]$. As a result, the rule becomes

$$J_c^o = tr(AW^t\Sigma_x WA(W^tW)^{-1}). \qquad (21)$$

From $\nabla_W J_c^o = 0$, we have

$$\Sigma_x WA(W^tW)^{-1}A - W(W^tW)^{-1}AW^t\Sigma_x WA(W^tW)^{-1} = 0$$

Similar to the analysis of eq.(17), we see that its solutions must satisfy $W^tW = b^2I$. That is, $W = b\Phi R$, $R^tR = I$. Putting this into the above equation, we have

$$\Sigma_x \Phi RAA - \Phi RAR^t\Phi^t\Sigma_x\Phi RA = 0 \quad .$$

which further reduces into

$$(R^t\Phi^t\Sigma_x\Phi R)A - A(R^t\Phi^t\Sigma_x\Phi R) = 0 \quad .$$

It holds only when $(R^t\Phi^t\Sigma_x\Phi R)$ is diagonal matrix, and thus only when $R = I$ and Φ consists of the m eigenvectors of Σ_x. Moreover, J_c^o reaches the maximum when the m eigenvectors coressponding to the m largest eigenvalues. Moreover, we can also show that all the other solutions for Φ are saddle points for ρ_c^o. Therefore, we have that Φ consists of eigenvectors that correspond to the m largest eigenvalues of Σ_x. In the other words, this modified rule performs the true PCA.

Another possible extension is to modify the rules eq.(8) eq.(9) eq.(10) eq.(11) and eq.(12) in such a way: we let the reference variable $\vec{\xi}$ from a normal $N(0, \Sigma_\xi)$ instead of $N(0, \Sigma_\xi)$ for the the linear network with $s_i(r) = r$ in eq.(1). As a result, the rules eq.(8) eq.(9) eq.(10) eq.(11) eq.(12) becomes

$$\begin{aligned} \rho_d &= f(det(W^t\Sigma_x W))/g(det(W^t\Sigma_\xi W)), \\ \rho_e &= f(0.5\ln det(W^t\Sigma_x W) + h) - g(0.5\ln det(W^t\Sigma_\xi W) + h), \\ J_c^o &= tr(W^t\Sigma_x W(W^t\Sigma_\xi W)^{-1}), \\ J_d^o &= f(det(W^t\Sigma_x W)) + cg(|tr(W^t\Sigma_\xi W - I))|, \\ J_e^o &= f(0.5\ln det(W^t\Sigma_x W) + h) + cg(|tr(W^t\Sigma_\xi W - I))|. \end{aligned}$$

We can find that the rules will performs the task of generalized PSA or PCA in the sense of $\Sigma_\xi^{-1}\Sigma_x W = W\Lambda$

VI. Relations Between the Theories and To Other Works

The rules developed based on the proposed theories actually can be grouped under two principles. One is the *self-reconstruction* of a system's output through the inverse mapping of the same system such that the reconstruction approximates the input best. The rules given in Section II.B are all under this principle. The other one is the *best preservation* of some intrinsic items of the input such that we can describe, based on the system's output, as good as possible the key features of the objects that the original input represents. The rules given in Section II.C&D are all under this principle. The two principles are also related. When the output catched the intrinsic items of the input, it is more easy for us to reconstruct the input based on these the intrinsic items. Conversely, when the *self-reconstruction* is a good approximation of the input, the output that the reconstruction is based on should contain the enough intrinsic features about the input. This is the reason why all the rules merge at performing PSA or PCA task for the linear network. The other common point that may deserve to mention is that when we talk about the best in the both principles, we actually make comparision with some thing—a reference, either implicitly or explicitly. For the *self-reconstruction* principle, the reference is explicitly the input itself. While for the principle of *best preservation*, in Section II.C&B we have taken the data from the standard normal distribution $N(0, I)$ and its transformation by the system as the reference for the input and output respectively.

Next, we discuss the relations of these proposed theories to some existing works in the literature.

The error measure eq.(3) is a more general rule which inculdes one of the cost function studied in [4] and into the symmetrical learning rule of [34] as its special cases. The error measure eq.(3) will reduce into the cost in [4] for the special single linear neuron case that $m = 1$ and $s(r) = r$, and the symmetrical learning rule of [34] for the special linear case of $s_i(r) = r$ for all i. The rule eq.(3) also relates to the auto-association back propagation learning of feedfarward network with one hidden layer [6]. The different point is that in this feedfarward net the reconstruction of the input from the hidden layer is implemented by the system of output layer which is not the same as the input layer. While our rule eq.(3) reconstructs the input through the input system itself reversely. In addition, in our paper [35] [39] the role of nonlinear sigmoid function, especially its roleon true PCA type tasks, has been explicitly uncovered, and this kind of nonlinearity has been deliberately suggested to break the different feature directions.

The rules eq.(8) and eq.(9) are related to those methods based information transfer theory, especially mutual information by [16] [1] [5] [28] [2] [17]. However, the rules eq.(8) and eq.(9) are differet from these methods in four aspects. First, all these mentioned methods consider a linear system, while the rules eq.(8) and eq.(9) also cover the nonlinear system given by eq.(1). Second, in section II the so called uncertainty measures are indeed closely related to the measure of information, i.e., the entropy. The measure can exactly be the entropy, but it can also be some other measure which is not exactly entropy or some functions (linear or nonlinear) of entropy. Third, we consider the system's performance in comparison with an explicit reference and our relative uncertainty measure is actually a ratio of some nonlinear function of the entropy of the output transfered from the input that is not corrupted by noise to some other nonlinear function of the entropy of output transfered from the reference. While those mentioned methods consider noise either in the input or in the output (or in both) and maximize the mutual information between the input and the noise-corrupted output under certain equality constraint. Fourth, the cost functions of those methods are also not same as eq.(8) and eq.(9). Moreover, the rules eq.(8) and eq.(9 are also different from the one proposed in [15]. However, all the rules have one thing in common: they will perform PSA or PCA tasks for the linear systems.

Furthermore, for the special cases of a single linear neuron and togather with the condition that $f(r) = g(r) = r$, we can also see that eq.(8) will collapse into the cost function given in [19] [36], and eq.(11) will collapse into the cost function given in [8][40]. In addition, for the special linear case of $s_i(r) = r$ the rule eq.(21) can also be regarded as an unconstrained version of Brockett's constrained cost function for his gradient flow[7].

Acknowledgements

I would like to acknowledge partial support from CUHK by a DIRECT GRANT for Research 93-94.

References

[1] J.J. Atick and A.N.Redlich. *Neural Computation*, vol. 2, 1990, pp. 308-320.

[2] J.J. Atick and A.N.Redlich. *Neural Computation*, vol. 5, 1993, pp. 45-60.

[3] P. Baldi and K. Hornik. *Neural Networks*, vol. 2, 1989, pp. 53-58.

[4] P. Baldi and K. Hornik. "Back-Propagation and Unsupervised Learning in Linear Networks". In Y. Chauvin and D.E. Rumelhart (Eds.), *Back Propagation: Theory, Architectrues and Applications*. Lawrence Erlbaum, 1991.

[5] S. Becker. *Int. J. of Neural Systems*, vol. 2, 1991, pp. 17-33.

[6] H. Bourlard and Y. Kamp. *Biological Cybernetics*, vol. 59, 1988, pp. 291-294.

[7] R.W. Brockett. *Linear Algebra Appl.*, vol. 146. 1991, pp. 79-91.

[8] Y. Chauvin, in *Proc. of IEEE International Conf. on Neural networks*, Washington D.C., Vol I, 373-380, New York:IEEE Press.

[9] P. Földiák. In *Proc. of the Int. Joint Conf. on Neural Networks*, Washington DC, 1989, pp. I-401-406.

[10] K. Hornik and C.-M. Kuan. *Neural Networks*, vol. 5, 1992, pp. 229-240.

[11] J. Karhunen and J. Joutsensalo. In I. Aleksander and J. Taylor (Eds.) *Artificial Neural Networks, 2* (Proc. ICANN'92, Brighton, United Kingdom, September 1992). North-Holland, Amsterdam, 1992, pp. 1099-1102.

[12] J. Karhunen and J. Joutsensalo. In *Proc. Int. Joint Conf. Neural Networks*, Nagoya, Japan, October 1993, pp. 2599-2602.

[13] S.Y. Kung and K.I. Diamantaras. In *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, Albuquerque, New Mexico, April 1990, pp. 861-864.

[14] T.K. Leen. *Network*, vol. 2, 1991, pp. 85-105.

[15] R. Lenz and M. Österberg. *Neural Computation*, vol. 4, 1992, pp. 382-392.

[16] R. Linsker. *Proc. of Int. Joint Conf. on Neural Networks*, 1990, pp. II-291-297.

[17] R. Linsker. *Neural Computation*, vol. 4, 1992, pp. 691-702.

[18] E. Oja. *J. of Mathematical Biology*, vol. 16, 1982, pp. 267-273.

[19] E. Oja and J. Karhunen. *J. of Math. Analysis and Applications*, vol. 106, 1985, pp. 69-84.

[20] E. Oja. *Int. J. of Neural Systems*, vol. 1, 1989, pp. 61-68.

[21] E. Oja, H. Ogawa, and J. Wangviwattana. In T. Kohonen et al. (Eds.), *Artificial Neural Networks* (Proc. ICANN'91, Espoo, Finland, June 1991). North-Holland, Amsterdam, 1991, pp. 385-390.

[22] E. Oja. In T. Kohonen et al. (Eds.), *Artificial Neural Networks* (Proc. ICANN'91, Espoo, Finland, June 1991). North-Holland, Amsterdam, 1991, pp. 737 - 745.

[23] E. Oja. *Neural Networks*, vol. 5, 1992, pp. 927-936.

[24] E. Oja. "A Nonlinear Symmetrical PCA Network Learns True Principal Components". *Research Report 40*, Lappeenranta Univ. of Technology, Dept. of Information Technology, 1993.

[25] E. Oja, H. Ogawa, and J. Wangviwattana. *IEICE Trans. on Information and Systems (Japan)*, vol. E75-D, 1992, pp. 366 - 375 (part I) and pp. 376-382 (Part II).

[26] F. Palmieri, J. Zhu, and C. Chang. "Anti-Hebbian Learning in Topologically Constrained Linear Networks: A Tutorial". To appear in *IEEE Trans. on Neural Networks.*

[27] F. Palmieri. *Proc. of the World Congress of Neural Networks*, Portland, Oregon, July 1993, pp. II-339-343.

[28] M.D. Plumbley. *Neural Networks*, vol. 6, 1993, pp. 823-833.

[29] J. Rubner and P. Tavan. *Europhysics Letters*, vol. 10, 1989, pp. 693-698.

[30] T.D. Sanger. *Neural Networks*, vol. 2, 1989, pp. 459-473.

[31] W.R. Softky and D.M. Kammen. *Neural Networks*, vol. 4, 1991, pp. 337-347.

[32] J.G. Taylor and S.G. Coombes. *Neural Networks*, vol. 6, 1993, pp. 423-427.

[33] L.-Y. Wang and E. Oja. In *Proc. 8th Scandinavian Conf. on Image Analysis*, Tromso. Norway, May 1993, pp. 531-537.

[34] R.J. Williams. "Featrue Discovery Through Error-Correction Learning". *Technical Report 8501.* Institute of Cognitive Science, University of California, San Diego, CA, 1985.

[35] L. Xu. In *Proc. Int. J. Conf. on Neural Networks*, Singapore, November 1991, pp. 2362-2367 (part I) and pp. 2368-2373 (part II).

[36] L. Xu, E. Oja, and C.Y. Suen, *Neural Networks*, vol. 5, 1992, pp. 441-457.

[37] L. Xu and A. Yuille, In *Proc. Int. J. Conf. on Neural Networks*, Baltimore, Maryland, June 1992, pp. I-812-817, also in S.J. Hanson, J.D. Cowan, and C.L. Giles (Eds.), *Advances in Neural Information Processing Systems 5.* Morgan Kaufmann, San Mateo, California 1993, pp. 467-474.

[38] L. Xu and A. Yuille. "Robust Principal Component Analysis by Self-Organizing Rules Based on Statistical Physics Approach". *Harvard Robotics Laboratory: Technical Report.* No.92-3, February 1992; Also in *IEEE Trans. Neural Networks*, in press, 1994.

[39] L. Xu, *Neural Networks*, vol. 6, 1993, pp. 627-648.

[40] A.L. Yuille, D.M. Kammen, and D.S. Cohen, *Biological Cybernetics*, vol. 61, 1989, pp. 183-194.