Verleysen M. (ed.). *ESANN'96. Proceedings.*
4th European Symposium on Artificial Neural Networks.
D facto Publ., Brussels, 1996. pp. 223–228.

Recurrent Least Squares Learning for Quasi–Parallel Principal Component Analysis

Włodzimierz KASPRZAK Andrzej CICHOCKI [1]

Frontier Research Program RIKEN
Laboratory for Artificial Brain Systems
2–1 Hirosawa, Wako–shi, Saitama 351–01, JAPAN
E-mail: *cia@kamo.riken.go.jp*

Abstract. The *recurrent least squares* (RLS) learning approach is proposed for controlling the learning rate in parallel *principal subspace analysis* (PSA) and in a wide class of *principal component analysis* (PCA) associated algorithms with a quasi–parallel extraction ability. The purpose is to provide a useful tool for applications where the learning process has to be repeated in an on–line self–adaptive manner. The methods are compared with a sequential PCA method for image compression.

1. Introduction

Independently of the learning algorithm that is applied for neural network based *principal component analysis* (PCA) [2,5,8] a higher order principal component m can be estimated if and only if all the previous components $(1, 2, ..., m-1)$ are already extracted or exactly estimated. This means that we are not able to extract all required principal components in a fully *parallel way* (i.e. simultaneously).

An alternative approach to signal (or image) compression and feature extraction is the *principal subspace analysis* (PSA) [6]. Its advantage is a fully parallel working ability, i.e. a simultaneous calculation of the subspace spanned by specified number of principal components. Instead of a relatively simple scalar algebra, like in sequential PCA, a computationally more expensive but compact matrix algebra is required for PSA. From a parallel method we usually expect that good quality results will be available in a very fast manner. This is not automatically guaranteed by the original PSA algorithm which converges very slowly. A class of quasi–parallel PCA algorithms, which we call *ordered* PSA can also be considered in this context [3,7]. These methods are given in matrix form, like the PSA method, but they do not work fully parallel due to additional control by specific *ordering operators* (matrix).

In order to provide the fast learning convergence of parallel PSA and quasi–parallel PCA methods in this paper a recurrent least squares (RLS) learning rate adaptation is proposed for them. The performance to speed trade–off

[1] On leave from Warsaw University of Technology, Department of Electrical Engineering, Warsaw, Poland.

of such fast converging methods will be compared with the recently proposed CRLS method, that is a reliable and fast sequential PCA algorithm [4].

2. The CRLS method for PCA

The standard PCA, called also Karhunen–Loeve transformation, (KL) determines an optimal linear transformation $\boldsymbol{y} = \boldsymbol{W}\boldsymbol{x}$ of an input vector $\boldsymbol{x}$, where $\boldsymbol{x} \in \mathcal{R}^n$ is a zero–mean input vector, $\boldsymbol{y} \in \mathcal{R}^m$ is the output vector and $\boldsymbol{W} = [\boldsymbol{w}_1, \boldsymbol{w}_2, ..., \boldsymbol{w}_m]^T \in \mathcal{R}^{m \times n}$ is a desired transformation matrix. The orthogonal vectors $\boldsymbol{w}_j = [w_{j1}, w_{j2}, ..., w_{jn}], (j = 1, 2, ..., m)$, are called *principal components* (usually $m << n$).

The task of principal component extraction can be accomplished in a sequential manner by using a *cascade neural network*. A recently proposed learning algorithm, called CRLS [4], combines advantages of three techniques: RLS learning rate adaptation, Hebbian–like learning rule and signal reduction (*deflation*).

Let $k(k = 1, 2..., N)$ be the index of signal samples $\boldsymbol{x}(k)$ and $j(j = 1, 2, ...m)$ be the index of principal components. For the first principal component extraction the signal $\boldsymbol{e}_1(k) = \boldsymbol{x}(k)$ is used. The RLS approach of learning rate adaptation allows an automatic setting according to current signal energy:

$$\eta_j(0) = \frac{\sum_{i=1}^{n} \sum_{k=1}^{N} e_{j(ik)}^2}{N}, \quad \eta_j(k) = \eta_j(k-1) + y_j(k)^2, \tag{1}$$

where $y_j(k) = \boldsymbol{w}_j^T(k-1)\boldsymbol{e}_j(k)$. The synaptic weight vector $\boldsymbol{w}_j(k)$ is updated according to following formula:

$$\boldsymbol{w}_j(k) = \boldsymbol{w}_j(k-1) + \frac{y_j(k)}{\eta_j(k)}(\boldsymbol{e}_j(k) - \boldsymbol{w}_j(k-1)y_j(k)). \tag{2}$$

The signal reduction for next component extraction $\boldsymbol{w}_{j+1}$ is as follows:

$$\boldsymbol{e}_{j+1}(k) = \boldsymbol{e}_j(k) - y_j(k)\boldsymbol{w}_j(k) \tag{3}$$

where $y_j(k) = \boldsymbol{w}_j^T(k)\boldsymbol{e}_j(k)$.

3. PSA and associated PCA with RLS

The adaptive algorithm of *principal subspace analysis* (PSA) has been developed by Oja and Karhunen [6]. It can be written in generalized (modified) form as

$$\boldsymbol{W}(t+1) = \boldsymbol{W}(t) + \boldsymbol{\eta}(t)[\boldsymbol{y}(t)\boldsymbol{x}^T(t) - \boldsymbol{y}(t)\boldsymbol{y}^T(t)\boldsymbol{W}(t)] \tag{4}$$

where $\boldsymbol{y}(t) = \boldsymbol{W}(t)\boldsymbol{x}(t)$ and $\boldsymbol{\eta}(t)$ is a suitable positive–definite matrix.

The PSA algorithm is able to learn only a rotated basis of the PC's subspace, i.e. PSA determines the subspace spanned by the first m $(m < n)$ principal eigenvectors with imposed constraint

$$\boldsymbol{w}_j^T \boldsymbol{R}_{xx} \boldsymbol{w}_i = 0; \quad \text{for } i \neq j; \quad \text{where } \boldsymbol{R}_{xx} = E\{\boldsymbol{x}\boldsymbol{x}^T\}. \tag{5}$$

In order to extract true principal components some non-symmetry must be introduced in the learning rule or some nonlinearity must be incorporated. We extend two learning algorithms from a class called here *ordered PSA*: the Brockett subspace algorithm (BSA) and Sanger's generalized Hebbian algorithm (GHA).

The Brockett learning rule differs from the PSA rule by the introduction of a nonsingular diagonal matrix $\boldsymbol{D}$ [3]:

$$\boldsymbol{W}(t+1) = \boldsymbol{W}(t) + \boldsymbol{\eta}(t)[\boldsymbol{D}\boldsymbol{y}(t)\boldsymbol{x}^T(t) - \boldsymbol{y}(t)\boldsymbol{y}^T(t)\boldsymbol{D}\boldsymbol{W}(t)], \quad (6)$$

$$\boldsymbol{D}(t) = diag(d_1, d_2, ..., d_m); \quad \text{where } 1 > d_1 > d_2 > ... > d_m > 0 \quad (7)$$

The second modified form of ordered PSA in our experiments is the generalized Hebbian algorithm (GHA), proposed by Sanger [7]:

$$\boldsymbol{W}(t+1) = \boldsymbol{W}(t) + \boldsymbol{\eta}(t)[\boldsymbol{y}(t)\boldsymbol{x}^T(t) - LT(\boldsymbol{y}(t)\boldsymbol{y}^T(t))\boldsymbol{W}(t)]. \quad (8)$$

$LT(.)$ means the Lower Triangular operation, i.e. it sets the above diagonal entries of the matrix to zero.

4. RLS technique in matrix form

The initial learning rate is a diagonal matrix with values on the diagonal equal to $\boldsymbol{\eta}(0) = (\sigma_{all}^2)^{-1}\boldsymbol{I}$; where σ_{all}^2 is the variance of the input signal. After t steps the learning rate can be computed as follows:

$$\boldsymbol{\eta}(t) = (\sum_{k=1}^{t} \boldsymbol{y}(k)\boldsymbol{y}^T(k))^{-1} = \boldsymbol{\eta}(t-1) - \frac{\boldsymbol{\eta}(t-1)\boldsymbol{y}(t)\boldsymbol{y}^T(t)\boldsymbol{\eta}(t-1)}{\boldsymbol{y}^T(t)\boldsymbol{\eta}(t-1)\boldsymbol{y}(t)}. \quad (9)$$

Let us notice that in our formulation $\boldsymbol{\eta}(t)$ is a matrix and not a scalar. The elements of this matrix are converging to zero with different speed during the learning process according to the above rule.

5. Computer simulation results

Several grey scale images have been used for tests of the PCA methods. The images with resolution 512×512 are divided into 4096 blocks of 8×8 pixels each and converted into vector samples of 64 elements. The quality of applied methods is tested by using the extracted PC-s or PS-s, given by the weight matrix $\boldsymbol{W}_j$ (of size $j \times 64$, where $j = 1, 2, ..., 64$), directly for reconstruction of the images. This procedure can be described by a sequence of two following steps: 1) $\boldsymbol{y}(t) = \boldsymbol{W}_j\boldsymbol{x}(t)$; 2) $\widehat{\boldsymbol{x}}(t) = \boldsymbol{W}_j^T\boldsymbol{y}(t)$; where $\boldsymbol{W}_j \in R^{j \times 64}, \boldsymbol{x}(t) \in R^{64}, \boldsymbol{y}(t) \in R^j$, $j = 1, 2, ..., 64$.

In Fig. 1 and 2 two image reconstruction results are shown, by applying 8 principal components or 8 subspace vectors respectively extracted by three methods with RLS learning (CRLS, PSA, BSA). Two cases of the learning time are considered: 1) the epoch number is equal to the number of requested

Original | For 8 PCs in 8 epochs PSNR=24.92 *dB* (CRLS) | For 8 PCs in 1 epoch PSNR=22.83 *dB* (BSA)

Fig. 1: The original image *Girl* (left image) and its reconstruction on the basis of eight PC/PS-s, that are learned either in 8 epochs (center image) or in 1 epoch (right image) of image data. The best reconstruction quality was PSNR=24.96 *dB* after 22.51 epochs of CRLS learning.

Original | For 8 PCs in 8 epochs PSNR=33.11 *dB* (CRLS) | For 8 PCs in 1 epoch PSNR=25.79 *dB* (GHA)

Fig. 2: The original image *Road* (left image) and its reconstruction on the basis of eight PC/PS-s, that are learned either in 8 epochs (center image) or in 1 epoch (right image). The best reconstruction quality was PSNR=33.41 *dB* after 42.9 epochs of CRLS learning.

PC/PS-s (e.g. 8 epochs for 8 PS/PC-s) and 2) there is only one epoch independently of the number of requested PS/PC-s.

For quality judgment the best possible reconstruction quality should also be known. This optimum was found by the CRLS method in a relatively long learning process. Usually more than one epoch of the image data for every PC is required in order to fulfill both the weight stability condition $\Delta \boldsymbol{w}_j < 10^{-5}$ and the normalization to unit length condition, i.e. $|1 - ||\boldsymbol{w}_j||| < 10^{-2}$.

Quantitative results of image reconstruction related to above tests are provided in Fig. 3, 4. The peak signal to noise ratio (PSNR) is shown in all drawings. From the above figures it is clearly evident, that the sequential extraction of the principal components by the cascaded PCA method ensures very high quality. The sequential CRLS method achieves the best performance among all of the tested methods if the training time is one epoch for each com-

ponent. As far as the first 4 PC–s are concerned this method can also learn them in one epoch (i.e. 1/4 of the image data can be used for learning one component) with well quality. A further time shortage for one component by requesting more than four PC-s in one epoch, leads to worser quality of the CRLS results. If the neural network has to learn in an on–line manner in one epoch of image data and more than 6.25 % PC/PS-s (i.e. more than 4 from the set of 64) are needed then a quasi–parallel PCA method gives better results.

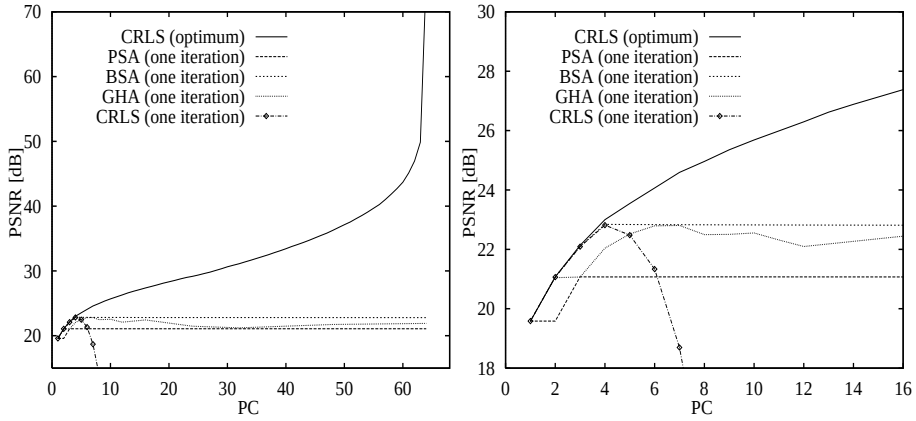

Fig. 3: The PSNR of the reconstructed image *Girl* for considered methods while learning in one epoch. Optimum means the best possible reconstruction performance.

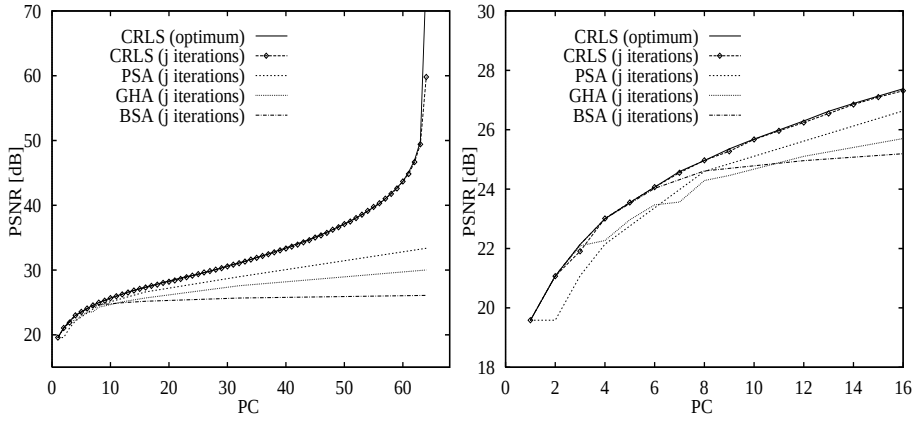

Fig. 4: The PSNR of the reconstructed image *Girl* for considered methods if the learning time is j epochs for total number of j PC/PS-s.

6. Conclusions

In this paper the RLS based adaptation of the learning rate for PSA and associated quasi–parallel PCA methods was proposed. It was searched for a speed to quality trade–off between sequential and parallel working manner, depending on the number of extracted PC/PS-s, while learning on natural image data.

In our experiments the sequential CRLS method has outperformed the parallel PSA and quasi–parallel ordered PCA methods for any number of PC/PS-s if the time of learning was in proportion to the number of PC/PS-s, where the learning time for one principal component was longer than the time required for visiting a subset of 25% of the whole image data. Although for the first 25% of PC/PS-s the quality difference is usually relatively small, but in this case there is also no speed advantage of the quasi–parallel and parallel methods over the sequential CRLS one.

In case of limited learning time the quasi–parallel PCA methods are a better choice than the PSA method. In the Brockett algorithm the vector $\boldsymbol{D}$ controls the *parallelity behavior* of learning (i.e. if $\boldsymbol{D} = \boldsymbol{I}$ it simplifies the PSA algorithm). This matrix can always be set according to the number of requested PC-s in such a way, that in a longer learning case BSA will perform with at least the same quality as the parallel PSA method does.

References

[1] S. Amari. Theory of adaptive pattern classifiers. *IEEE Transactions on Electric Computing*, EC–16:299–307, 1967.

[2] S. Amari. Neural theory of association and concept formation. *Biological Cybernetics*, 26:175–185, 1977.

[3] R.W. Brockett. Dynamical systems that sort lists, diagonalize matrices, and solve linear programming problems. *Linear Algebra Applications*, 146:79–91, 1991.

[4] A. Cichocki, W. Kasprzak, and Skarbek W. Adaptive learning algorithm for principal component analysis with partial data. In *Thirteenth European Meeting on Cybernetics and Systems Research*, Vienna, April 1996 (in print).

[5] A. Cichocki and R. Unbehauen. Robust estimation of principal components by using neural networks. *Electronic Letters*, 29:1869–1870, 1993.

[6] E. Oja and J. Karhunen. On stochastic approximation of the eigenvectors and eigenvalues of the expectation of a random matrix. *Journal of Mathematical Analysis and Applications*, 106:69–84, 1985.

[7] T.D. Sanger. Optimal unsupervised learning in a single–layer linear feedforward neural network. *Neural Networks*, 2:459–473, 1989.

[8] J.G. Taylor and S. Coombes. Learning of higher order correlations. *Neural Networks*, 6:423–427, 1993.