IEEE Signal Processing Letters, vol.15, pp.677-680, 2008.

A Multipurpose Linear Component Analysis Method Based on Modulated Hebb Oja Learning Rule

Marko V. Jankovic, *Senior Member, IEEE,* and Masashi Sugiyama

Department of Computer Science, Tokyo Institute of Technology, Japan

Abstract –This letter presents a Hebb-type learning algorithm for on-line linear calculation of principal components. The proposed method is based on a recently proposed cooperative-competitive concept, named the time-oriented hierarchical method. The algorithm performs deflation on the signal power rather than on the signal itself. It will be also shown when, or how, this algorithm can be used as a blind signal separation algorithm. The proposed synaptic efficacy learning rule does not need the explicit information about the value of the other efficacies to make individual efficacy modification. The number of necessary global calculation circuits is one.

Index terms - adaptive algorithm, principal/independent component analysis

EDICS: SAS-STAT Detection, estimation and classification theory and methods, statistical signal processing
SAS-ADAP Adaptive systems and adaptive filtering

Corresponding author:

Marko Jankovic,

Tokyo Institute of Technology, Department of Computer Science,

Ookayama 2-12-1, Meguro-ku, Tokyo, 152-8552, Japan;

Tel: +81-3- 5734-2699, FAX: +81-3-5734-2699

e-mail: marko@sg.cs.titech.ac.jp

1. Introduction

Neural networks provide a way for parallel on-line computations of principal component analysis (PCA) or principal subspace analysis (PSA) [1]. The goal of PSA or PCA is to extract from a stationary, random, zero-mean process $\boldsymbol{x}(i) \in \Re^K$, with covariance matrix $\boldsymbol{C}=\mathrm{E}\{\boldsymbol{x}(i)\boldsymbol{x}(i)^{\mathrm{T}}\}$, the subspace spanned by the N principal eigenvectors of $\boldsymbol{C}$ or principal vectors themselves, respectively. Within last years various PCA and PSA learning algorithms have been proposed and mathematically investigated (see e.g. [1]). Most of the proposed algorithms are based on local Hebbian learning - due to locality it has been argued that these algorithms are biologically plausible. However, it is not frequently analyzed how those algorithms could be related to the known biological neural architectures or could they have multiple purposes.

In [2], biologically inspired PSA methods, named Modulated Hebbian (MH) and Modulated Hebb-Oja (MHO) learning rules have been introduced. Major objectives for the methods' derivation were to obtain a network which has a learning rule for individual synaptic efficacy that requires the least possible amount of explicit information about the other synaptic efficacies, especially those related to other neurons and to minimize the neural hardware that is necessary for implementation of the proposed learning rule.

In this paper modification of the MHO algorithm, named the MultIpurpose LInear Component Analysis MHO (MILICA-MHO) algorithm is proposed and analyzed. Generally speaking the MILICA-MHO algorithm performs PCA. If the input signal is prewhitened the same algorithm can perform blind signal separation (BSS). The new algorithm is obtained by implementation of the modified (generalized) time-oriented hierarchical method [5,6] on the MHO method. Comparing to bigradient or related nonlinear PCA algorithms [7] which can be used in a similar way, or Generalized Hebbian Learning (GHA) [9] the proposed algorithm is much more based on local calculations.

2. Theory

Let $\boldsymbol{x} \in \Re^K$ denote the input random variables with mean zero, and let $\boldsymbol{y} = \boldsymbol{W}^{\mathrm{T}}\boldsymbol{x} \in \Re^N$ denotes the output, where $\boldsymbol{W} \in \Re^{KxN}$ denote the synaptic weight matrix. In scalar form, for the n-th output, we have $y_n = \boldsymbol{w}_n^{\mathrm{T}}\boldsymbol{x}$ ($n = 1, \ldots, N$), where $\boldsymbol{w}_n \in \Re^K$ denotes the column vector of $\boldsymbol{W}$.

The Modulated Hebbian (Oja) (MHO) learning rule is introduced in [2] and analyzed in [3, 4]. The MH(O) rule can be derived as a gradient descent learning rule for minimization of the following cost function:

$$J(\boldsymbol{W}) := \mathrm{E}\left(\left\|\boldsymbol{x}-\boldsymbol{W}\boldsymbol{y}\right\|^4\right) = \mathrm{E}\left(\left(\left\|\boldsymbol{x}-\boldsymbol{W}\boldsymbol{W}^{\mathrm{T}}\boldsymbol{x}\right\|^2\right)^2\right) \qquad (1)$$
$$= \mathrm{E}\left(\left(\mathrm{tr}(\boldsymbol{x}\boldsymbol{x}^{\mathrm{T}}) - 2\mathrm{tr}\left(\boldsymbol{W}^{\mathrm{T}}\boldsymbol{x}\boldsymbol{x}^{\mathrm{T}}\boldsymbol{W}\right) + \mathrm{tr}\left(\boldsymbol{W}^{\mathrm{T}}\boldsymbol{x}\boldsymbol{x}^{\mathrm{T}}\boldsymbol{W}\boldsymbol{W}^{\mathrm{T}}\boldsymbol{W}\right)\right)^2\right), \quad \|\boldsymbol{w}_n\| = 1$$

or, under assumption $\boldsymbol{W}^{\mathrm{T}}\boldsymbol{W}=\boldsymbol{I}$ ($\boldsymbol{I}$ is identity matrix), in compact notation

$$J_N^{MHPSA}(\boldsymbol{W}) := \mathrm{E}\left\{\left(\boldsymbol{x}^{\mathrm{T}}\boldsymbol{x}-\boldsymbol{y}^{\mathrm{T}}\boldsymbol{y}\right)^2\right\} = \mathrm{E}\left\{\left(\boldsymbol{x}^{\mathrm{T}}\left(\boldsymbol{I}-\boldsymbol{W}\boldsymbol{W}^{\mathrm{T}}\right)\boldsymbol{x}\right)^2\right\} \qquad (2)$$
$$= \mathrm{E}\left\{\left(\sum_{k=1}^{K} x_k^2 - \sum_{n=1}^{N} y_n^2\right)^2\right\}, \quad \|\boldsymbol{w}_n\| = 1.$$

As it was explained in [2-4] MHO learning algorithm represents the learning rule in which individual synaptic efficacy modification does not require the explicit information about the other synaptic efficacies, especially those related to other neurons. In order to obtain a new PCA algorithm with the same feature, we will introduce the following set of cost functions:

$$J_n^{MHPCA}(\boldsymbol{W}) := J_N^{MHPSA}(\boldsymbol{W}) + f(a)\mathrm{E}\left\{\left(\sum_{k=1}^{K} x_k^2 - \sum_{j=1}^{n} y_j^2\right)^2\right\}, \quad \|\boldsymbol{w}_n\| = 1,$$

$$f(a) = \begin{cases} a, & n \neq N, \\ 0, & n = N. \end{cases}$$

It can be noticed that for every weight vector $\boldsymbol{w}_n$ we have a different learning rule which consists of one common part and one part which is specific for the n-th output neuron. By minimization of the proposed cost function the new learning rule for the n-th weight vector can be written in the form

$$\boldsymbol{w}_n(i+1) = \boldsymbol{w}_n(i) + g(i)\left(\boldsymbol{x}(i)y_n(i) - \boldsymbol{w}_n(i)y_n^2(i)\right)\left(\|\boldsymbol{x}(i)\|^2 - \|\boldsymbol{y}(i)\|^2\right) \qquad (3)$$
$$+ f(a)g(i)\left(\boldsymbol{x}(i)y_n(i) - \boldsymbol{w}_n(i)y_n^2(i)\right)\left(\|\boldsymbol{x}(i)\|^2 - \sum_{j=1}^{n} y_j^2(i)\right), \quad n = 1,..,N,$$

where $\gamma(i)$ is the learning factor, and i defines steps in time (t). We actually have a system of equations that have the same family (common) part of the learning rule, while all individual (specific) parts of

the learning rules are different. While the GHA algorithm [9] for modification of synaptic vector $\boldsymbol{w}_n$ (related to output neuron n) requires explicit information about all synaptic efficacies related to output neurons 1 to n-1, the proposed algorithm does not need any explicit information about the synaptic efficacies related to any other neurons.

Now we can state the following theorem (proof is diverted to Appendix):

Theorem 1: Columns of $\boldsymbol{W}$ evolve to span the principal subspace of $\boldsymbol{C}$ under the assumption that the input is bounded and a is properly chosen (in a sense that it is possible to achieve stable solution). If rank[$\boldsymbol{W}$] = N, then the space of locally-asymptotically-stable stationary points satisfy $\boldsymbol{W}^{\mathrm{T}}\boldsymbol{W} = \boldsymbol{I}$ and columns of $\boldsymbol{W}$ will be equal to principal eigenvectors of $\boldsymbol{C}$.

3. Numerical experiments

Here, we will examine the small scale numerical simulations results. The number of inputs was set at K = 5 and the number of output neurons was N = 3. Artificial zero-mean vectors with uncorrelated elements were generated by the following equations:

$$
\begin{aligned}
&x(1,i) = .45\sin(i/2);\\
&x(2,i) = .45\,((\mathrm{rem}(i,23)-11)/9).\hat{}5;\\
&x(3,i) = .35\sin(i/17.8);\\
&x(4,i) = .145\,((\mathrm{rand}(1,1) < .5)*2-1).\cdot\log(\mathrm{rand}(1,1)+.5);\\
&x(5,i) = .18\ \ \mathrm{randn}(1,1)\,,
\end{aligned}
$$

where rem, rand and randn represent standard MATLAB functions (rem- remainder after integer division, rand – random number generator from uniform distribution and randn – random number generator from normal distribution). The input signal is constructed as $\boldsymbol{s}$ = 0.47*$\boldsymbol{mix}$*$\boldsymbol{x}$, where the mixing matrix $\boldsymbol{mix}$ is defined as $\boldsymbol{mix}$ = -.5 + rand(K).

In Fig.1 cosine of the angles between column vectors of matrix $\boldsymbol{W}$ and numerically calculated eigenvectors of the input signal covariance matrix were used as illustration of the algorithm efficiency. The learning rate was chosen constant, g_i = 3.45, for the first 15000 iterations, and then was set at g_i = 0.115. The individual part was taken as α=0.5. The initial value for matrix $\boldsymbol{W}$ was selected as $\boldsymbol{W}_{\mathrm{init}}$=-.5+rand(5,3).

In Fig. 2 we can see that proposed method can be efficiently used for extraction of the deterministic input signals (under the assumption that mixing matrix is orthogonal). The first column in Fig. 2 represents original signals, the second column represents the input signals obtained from the original signals by multiplication with the orthonormal matrix ***mix***, and the last column represents the output obtained by implementation of MILICA-MHO.

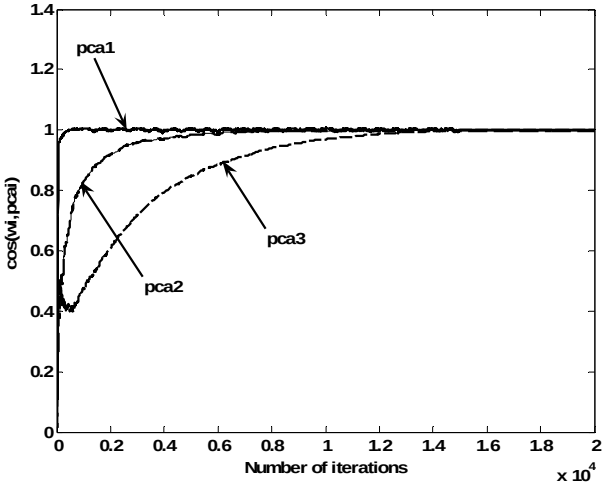

Fig. 1. Cosine of angles between column vectors of ***W*** and three principal eigenvectors of input covariance matrix versus the number of iterations

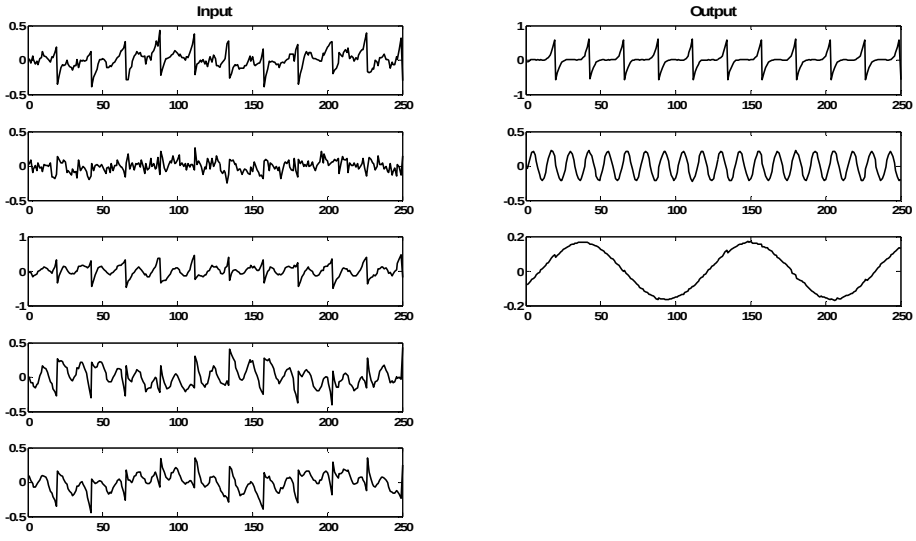

Fig. 2. Blind signal extraction of deterministic components (mixing matrix is orthonormal)

4. Can we use the proposed method for Blind Signal Separation?

In order to show how the algorithm proposed in this paper can be used as a BSS algorithm in some cases and in order to explain results obtained in Fig. 2, we will consider the simplest case $k = n = 2$. Also, we will assume that the input data is prewhitened. Then, the MILICA-MHO algorithm can be represented by the following equations:

$$\begin{aligned}\boldsymbol{w}_1(i+1) &= \boldsymbol{w}_1(i) + g(i)\left(\boldsymbol{x}(i)y_1(i) - \boldsymbol{w}_1(i)y_1^2(i)\right)\left(\|\boldsymbol{x}(i)\|^2 - \|\boldsymbol{y}(i)\|^2\right) \\ &\quad + ag(i)\left(\boldsymbol{x}(i)y_2(i) - \boldsymbol{w}_2(i)y_2^2(i)\right)\left(\|\boldsymbol{x}(i)\|^2 - y_1^2(i)\right) \\ \boldsymbol{w}_2(i+1) &= \boldsymbol{w}_2(i) + g(i)\left(\boldsymbol{x}(i)y_2(i) - \boldsymbol{w}_2(i)y_2^2(i)\right)\left(\|\boldsymbol{x}(i)\|^2 - \|\boldsymbol{y}(i)\|^2\right).\end{aligned} \tag{4}$$

that are obtained by minimization of the following two cost functions

$$\begin{aligned}J_1^{MHPCA}(\boldsymbol{W}) &= J_2^{MHPSA}(\boldsymbol{W}) + a\mathrm{E}\left\{\left(\left(x_1^2 + x_2^2\right) - y_1^2\right)^2\right\} \\ J_2^{MHPSA}(\boldsymbol{W}) &= \mathrm{E}\left\{\left(\sum_{k=1}^{2} x_k^2 - \sum_{j=1}^{2} y_j^2\right)^2\right\}.\end{aligned} \tag{5}$$

We can see that both cost functions contain a part which is used for symmetric orthogonalization while J_1^{MHPCA} contains also an additional part which directs the solution in some particular direction. Since we know that the input data is priwhitened and that the final solution will be an orthogonal matrix, we can approximately rewrite (5) as

$$\begin{aligned}J_1^{MHPCA}(\boldsymbol{W}) &\approx J_2^{MHPSA}(\boldsymbol{W}) + a\mathrm{E}\left\{\left(\sum_{k=1}^{2} y_k^2 - y_1^2\right)^2\right\} \\ J_2^{MHPSA}(\boldsymbol{W}) &= \mathrm{E}\left\{\left(\sum_{k=1}^{2} x_k^2 - \sum_{j=1}^{2} y_j^2\right)^2\right\}.\end{aligned} \tag{6}$$

or

$$\begin{aligned}J_1^{MHPCA}(\boldsymbol{W}) &\approx J_2^{MHPSA}(\boldsymbol{W}) + a\mathrm{E}\left(y_2^4\right) \\ J_2^{MHPSA}(\boldsymbol{W}) &= \mathrm{E}\left\{\left(\sum_{k=1}^{2} x_k^2 - \sum_{j=1}^{2} y_j^2\right)^2\right\}.\end{aligned} \tag{7}$$

since, in the vicinity of the solution it holds

$$\sum_{k=1}^{2} x_k^2 \cong \sum_{j=1}^{2} y_j^2 . \qquad (8)$$

So, if at most one of the input signals is supergaussian our method will perform signal separation from their prewhitened mixture, since the algorithm (7) is going to minimize the 4th moment of the output [7]. This can be easily extended to dimensions higher than 2.

If all signals, except one, are supergaussian, then the algorithm can perform BSS, given that the output is calculated as $y_n = -\boldsymbol{w}_n^{\mathrm{T}}\boldsymbol{x}$ ($n = 1, \ldots, N$), where $\boldsymbol{w}_n \in \Re^K$ denotes the column vector of $\boldsymbol{W}$. This can also be extended to higher dimensions.

5. Conclusion

In this letter, we have proposed a novel PCA/BSS algorithm based on modification of the PSA MH(O) algorithm. The algorithm employs a deflation technique that is implemented on the signal power, which could be seen as a more natural solution (especially in the PCA case). Locality of calculations could be seen as a desirable property for possible implementation in parallel hardware. If we know in advance the number of supergaussian and subgaussian signals, the algorithm can be easily adjusted to perform BSS of any kind of signals. If it is not the case, the algorithm and network structure have to be additionally modified. Theory and simulations indicate that the algorithm works well.

Appendix

Proof of Theorem 1:

Equation (3) can be written in the following form

$$\boldsymbol{w}_n(i+1) = \boldsymbol{w}_n(i) + g(i)\left(\boldsymbol{C}(i)\left(\|\boldsymbol{x}(i)\|^2 - \|\boldsymbol{y}(i)\|^2 + f(a)\left(\|\boldsymbol{x}(i)\|^2 - \sum_{j=1}^{n} y_j^2(i)\right)\right)\boldsymbol{w}_n(i) - \boldsymbol{w}_n(i)q(i)\right), \qquad (9)$$

where $\boldsymbol{C}(i) = \boldsymbol{x}(i)\boldsymbol{x}(i)^{\mathrm{T}}$, and $\theta(i)$ is defined as

$$q(i) = y_n^2\left(\|\boldsymbol{x}(i)\|^2 - \|\boldsymbol{y}(i)\|^2 + f(a)\left(\|\boldsymbol{x}(i)\|^2 - \sum_{j=1}^{n} y_j^2(i) \right) \right). \tag{10}$$

Differential counterpart of this difference equation can be written in the form [8, 9]

$$\frac{\mathrm{d}\boldsymbol{w}_n}{\mathrm{d}t} = \boldsymbol{F}_n\boldsymbol{w}_n - \boldsymbol{w}_n\Theta_n \qquad n = 1,2,\ldots,N\,, \tag{11}$$

where $\boldsymbol{F}_n$ is defined as

$$\begin{aligned} \boldsymbol{F}_n &= \mathrm{E}\left(\boldsymbol{C}(i)\left(\|\boldsymbol{x}(i)\|^2 - \|\boldsymbol{y}(i)\|^2 + f(a)\left(\|\boldsymbol{x}(i)\|^2 - \sum_{j=1}^{n} y_j^2(i) \right) \right) \right) \\ &= \mathrm{E}\left(\boldsymbol{C}(i)\left(\|\boldsymbol{x}(i)\|^2 - \|\boldsymbol{y}(i)\|^2 \right) \right) + f(a)\mathrm{E}\left(\boldsymbol{C}(i)\left(\left(\|\boldsymbol{x}(i)\|^2 - \sum_{j=1}^{n} y_j^2(i) \right) \right) \right) \\ &= \boldsymbol{F}\boldsymbol{f} + \boldsymbol{F}\boldsymbol{i}_n\,, \end{aligned} \tag{12}$$

where expectation is over $\boldsymbol{x}$ and Θ_n is defined as

$$\begin{aligned} \Theta_n &= \mathrm{E}\left(y_n^2\left(\|\boldsymbol{x}(i)\|^2 - \|\boldsymbol{y}(i)\|^2 + f(a)\left(\|\boldsymbol{x}(i)\|^2 - \sum_{j=1}^{n} y_j^2(i) \right) \right) \right) \\ &= \mathrm{E}\left(y_n^2\left(\|\boldsymbol{x}(i)\|^2 - \|\boldsymbol{y}(i)\|^2 \right) \right) + f(a)\mathrm{E}\left(y_n^2\left(\left(\|\boldsymbol{x}(i)\|^2 - \sum_{j=1}^{n} y_j^2(i) \right) \right) \right) \\ &= \Theta f_n + \Theta i_n. \end{aligned} \tag{13}$$

It is not difficult to conclude that, at stationary points, equation (11) represents system of equations for calculations of eigenvector of matrices $\boldsymbol{F}_n$. It is possible to prove that all matrices $\boldsymbol{F}_n$ have same eigenvectors as matrix $\boldsymbol{C}$=E($\boldsymbol{x}\boldsymbol{x}^{\mathrm{T}}$) – it can be done by straight calculation if we use empirical expectations and than show that matrices $\boldsymbol{F}_n$ commute with matrix $\boldsymbol{C}$. Also, it can easily be shown that all matrices $\boldsymbol{F}_n$ are symmetric – they represent expectation of the symmetric matrix multiplied by some scalar. Since $\boldsymbol{F}_n$ are real and symmetric then, if matrix $\boldsymbol{W}$ is of maximum rank for all i (for all time t), it is easy to see that matrix $\boldsymbol{W}$ is an orthonormal matrix.

In the analysis that follows, stable points of equation (11) will be investigated. In order to prove that original algorithm (9) will converge toward the same stable points, it should be proved that outputs are bounded and that (9) visits infinitely often, almost surely, a compact subset of the domain

of attraction of the asymptotically stable solution of (11) [8]. By implementation of the same method that was proposed in [4], it can be shown that synaptic vectors have bounded norms under some reasonable assumptions. This, together with assumptions of the boundness of the input signals, means that outputs are bounded. However, it must be said that proving that (9) visits infinitely often, almost surely, a compact subset of the domain of attraction of the asymptotically stable solution of (11) is very difficult. Such kind of proof does not exist for many of the known algorithms. Based on experience, generally can be said that if the desired limit of the discrete algorithm is not an asymptotically stable point of the averaged differential equation, then the convergence will not take place and the behavior of the discrete algorithm is not good [4]. Here will be said that in all simulations in which (9) converged it had the same stable points as (11).

Now, we will rewrite the equation (11) in the following form

$$\begin{aligned}\frac{\mathrm{d}\boldsymbol{w}_n}{\mathrm{d}t} &= (\boldsymbol{Ff} + \boldsymbol{Fi}_n)\boldsymbol{w}_n - \boldsymbol{w}_n(\Theta f_n + \Theta i_n) \\ &= (\boldsymbol{Ff}\ \boldsymbol{w}_n - \boldsymbol{w}_n \Theta f_n) + (\boldsymbol{Fi}_n\ \boldsymbol{w}_n - \boldsymbol{w}_n \Theta i_n) \\ &= \frac{\mathrm{d}\boldsymbol{wf}_n}{\mathrm{d}t} + \frac{\mathrm{d}\boldsymbol{wi}_n}{\mathrm{d}t} \qquad n = 1,2,\ldots,N\,.\end{aligned} \tag{14}$$

Without a loss of generality, we can assume that coordinate system coincides with eigenvectors of matrix $\boldsymbol{C}$. Then, the input data is decorrelated and covariance matrix of input data is diagonal. In that case analysis that was carried on in [4] could be applied. From [4, Theorem 4.3] we know that if $\boldsymbol{W}$ spans any subspace defined by some of the eigenvectors of matrix $\boldsymbol{C}$ then family part of the learning equations (11), denoted by $\mathrm{d}\boldsymbol{wf}_n/\mathrm{d}t$, is zero, so equation (14) can be written as

$$\frac{\mathrm{d}\boldsymbol{w}_n}{\mathrm{d}t} = \frac{\mathrm{d}\boldsymbol{wi}_n}{\mathrm{d}t} = \boldsymbol{Fi}_n\,\boldsymbol{w}_n - \boldsymbol{w}_n \Theta i_n \quad n = 1,2,\ldots,N\,. \tag{15}$$

In the case n=1 equation (15) represents the equation for calculation of the principal eigenvector of matrix $\boldsymbol{C}$ (having in mind [4]). Then, for n=2, this equation represents the equation for calculation of the eigenvector which is orthonormal to the first principal eigenvector $\boldsymbol{w}_1$ of $\boldsymbol{C}$ ($\boldsymbol{W}$ is an orthonormal matrix) and corresponds to the maximum possible eignvalue. So, it represents eqution for calculation of eigenvector that corresponds to the second largest eigenvalue of the matrix $\boldsymbol{C}$. This can be seen as a deflation procedure in which the data are not back projected and then subtracted from the original

data, but rather the energy of input signal is decreased by the amount that corresponds to the energy contained in the direction of the principal component. So, for $n > 2$ we can continue the same way of reasoning and conclude that $\boldsymbol{w}_n$ represents the eigenvector of matrix $\boldsymbol{C}$ which corresponds to n-th biggest eigenvalue of the matrix $\boldsymbol{C}$. This concludes the proof. Proper range of the parameter a can be assessed trough stability analysis.

REFERENCES

[1] A. Chichocki, S.–I. Amari, (2003) "Adaptive Blind Signal and Image Processing – Learning Algorithms and Applications", John Wiley and Sons, New York.

[2] M. Jankovic (2001) "A new modulated Hebbian learning rule – Method for local computation of a principal subspace", ICONIP 2001: 470-475

[3] M. Jankovic and H. Ogawa (2003) "A new modulated Hebb learning rule – Biologically plausible method for local computation of principal subspace", Int. J. Neural Systems 13: 215-224.

[4] M. Jankovic and H. Ogawa (2006) "Modulated Hebb-Oja learning rule – A method for principal subspace analysis", IEEE Trans. on Neural Networks 17: 345-356.

[5] M. Jankovic and B. Reljin (2005) "Neural learning on Grassman/Stiefel principal/minor submanifold", EUROCON 2005: 249-252.

[6] M. Jankovic and H. Ogawa (2004) "Time-Orineted Hierarachical Method for Computation of Principal Components using Subspace Learning Algorithm", Int. J. Neural Systems 14: 313-324.

[7] J. Karhunen, E. Oja, L. Wang, R. Vigario, J. Joutensalo (1997) "A Class of Neural Networks for Independent Component Analysis", IEEE Trans. on Neural Networks 8: 486-504.

[8] L. Ljung (1977) "Analysis of recursive stochastic algorithms", IEEE Trans. Automat. Contr. 22: 551-575.

[9] T. Sanger (1989) "Optimal Unsupervised Learning in a Single-Layer Linear Feedforward Neural Network", Neural Networks 2: 459-473.