Letter to the Editor
**A SELF-STABILIZING LEARNING RULE
FOR MINOR COMPONENT ANALYSIS**

RALF MÖLLER
*Computer Engineering Group, Faculty of Technology, Bielefeld University
D-33594 Bielefeld, Germany
WWW: www.ti.uni-bielefeld.de*

Received (to be inserted
Revised by Publisher)

The paper reviews single-neuron learning rules for minor component analysis and suggests a novel minor component learning rule. In this rule, the weight vector length is self-stabilizing, i.e. moving towards unit length in each learning step. In simulations with low- and medium-dimensional data, the performance of the novel learning rule is compared with previously suggested rules.

1. Introduction

Minor component analysis (MCA) determines the directions of smallest variance in a distribution. They are corresponding to the directions of those eigenvectors of the covariance matrix of the data which have the smallest eigenvalues. MCA has several applications in adaptive signal processing, parameter estimation, and computer vision.^{1,2} In neural network techniques for MCA, estimates of the minor eigenvectors are updated by sequentially processing data vectors drawn from the distribution. Neural MCA methods are advantageous for high-dimensional data, since they avoid the computation of the covariance matrix, and are suitable for the tracking of non-stationary distributions. Eigenvector estimates are interpreted as the synaptic weights of a neuron with usually linear response, and the update procedure is called a learning rule.

While neural network techniques for *principal* component analysis (PCA) can meanwhile be considered as fully developed, the search for robust learning rules for *minor* component analysis is still ongoing. This paper reviews single-neuron MCA rules. The review is based on the study by Cirrincione

et al.², where the rules OJAn,³ LUO,^{4,5,6} MCA-EXIN (here EXIN for short),² OJA+,⁸ and FENG⁹ were analyzed. In addition, this paper includes the MCA rules DOUG,¹⁰ CHEN,¹¹ AMEX,^{12,1} ZLlog and ZLdiff,¹³ WANG,¹⁴ as well as two rules derived from PCA rules, OJA*^{15,17} and XU*¹⁶.

Zhang and Leung¹³ define two generic MCA rules, each containing a scalar function of the weight vector for which certain conditions have to hold. Examples of appropriate functions are given, all including some norm of the weight vector. Since here we relate to the Euclidean norm, we only include two of the rules from the work of Zhang and Leung,¹³ in the following called ZLlog and ZLdiff; a third rule coincides with FENG. ZLdiff is closely related to the MCA rule WANG suggested by Wang and Karhunen¹⁴.

Additional single-neuron MCA rules can be obtained from PCA rules by replacing the covariance matrix $\mathbf{C}$ with $\beta\mathbf{I} - \mathbf{C}$, where $\mathbf{I}$ is a unit matrix and β is larger than the minor eigenvalue of $\mathbf{C}$. The modified PCA rules extract the principal component of $\beta\mathbf{I} - \mathbf{C}$ which is the minor component of $\mathbf{C}$.¹⁷ In this paper, only two such rules, XU*¹⁶ and OJA*¹⁵, are

included as examples. Note that the rules OJA* and OJA+ coincide for $\beta = 1$. In these rules, the range of the minor eigenvalue is restricted; outside this range, the rules diverge.²

Cirrincione et al.² distinguish between two classes of learning rules according to the time course of the length of the eigenvector estimate. In the rules of the first class, OJAn, LUO, and EXIN, the length of the weight vector in the fixed point is undetermined. In an exact solution of the differential equation, the weight vector length would not deviate from its initial value. However, when a numerical procedure (like Euler’s method) is applied, all these rules are plagued by “sudden divergence” of the weight vector length.^{7,2} In the rules of the second class, DOUG, CHEN, XU*, OJA*, OJA+, FENG, AMEX, ZLlog, ZLdiff, and WANG, the weight vector length converges towards a fixed value.

Within the second class, some rules are *self-stabilizing* with respect to the time course of the weight vector length. In these rules, the weight vector length converges towards a fixed value (here always 1) independently of the presented input vector. The rules DOUG and CHEN are *strictly* self-stabilizing, while OJA+, OJA*, and XU* show this property in the vicinity of the fixed point. All rules *lacking* self-stabilization are potentially prone to fluctuations and divergence in the weight vector length.

In this work, we present a novel MCA learning rule belonging to the second class. The rule bears a structural resemblance with the DOUG rule and can be derived from the same framework as both the DOUG and the CHEN rule. With the latter rules, the novel rule shares the property of being strictly self-stabilizing. In contrast to OJA+, OJA*, and XU*, the rule has no limitation in the allowed range of the minor eigenvalue.

After an overview of previously presented MCA rules and the methods for their analysis in Sec. 2, we introduce the novel learning rule in Sec. 3, and demonstrate its performance in simulations in Sec. 4. In Sec. 5.1 we discuss the relation of the novel rule to the “unified stabilization approach”.¹¹ We analyze the self-stabilization property of the tested rules in Sec. 5.2. A coupled learning system improving the self-stabilization of the novel rule is derived in Sec. 5.3. Convergence speed problems of MCA rules are analyzed in Sec. 5.4.

2. Overview of MCA Rules

All learning rules discussed in this work are based on a linear neuron model which computes its output y from the scalar product of the weight vector $\mathbf{w}$ and the input vector $\mathbf{x}$, both of dimension n , thus $y = \mathbf{w}^T \mathbf{x}$. In Table 1, the learning rules are given as differential equations in two forms. The first form (in the second column) is the averaged form $\dot{\mathbf{w}} = \mathbf{F}(\mathbf{C}, \mathbf{w})$, with a right-hand side depending on the covariance matrix $\mathbf{C}$. This form is amenable to the standard tools for the analysis of deterministic ordinary differential equations. Used for the training, however, is the second form (in the third column), the online rule $\dot{\mathbf{w}} = \mathbf{f}(\mathbf{x}, \mathbf{w})$. The online rule is a stochastic differential equation with a random input $\mathbf{x}$. It is usually solved numerically by Euler’s method, $\mathbf{w}_{t+1} = \mathbf{w}_t + \gamma_t \mathbf{f}(\mathbf{x}_t, \mathbf{w}_t)$, where γ_t is a time-dependent step size, $\mathbf{x}_t$ a random vector drawn from the distribution, and t the time step. Informally, the transition from averaged to online rule can be done by replacing $\mathbf{C}$ with $\mathbf{x}\mathbf{x}^T$. Stochastic approximation theory provides the conditions under which averaged and online rule have the same convergence goal; important is a step size γ_t that decreases to zero over time.^{18,19}

All averaged rules in Table 1 share the form

$$\dot{\mathbf{w}} = -f(\mathbf{w})\mathbf{C}\mathbf{w} + g(\mathbf{C}, \mathbf{w})\mathbf{w}, \quad (1)$$

accordingly also the online rules are of the same form. The fixed points are in all rules specified by $\mathbf{C}\bar{\mathbf{w}} = \bar{\lambda}\bar{\mathbf{w}}$, thus either $\bar{\mathbf{w}} = \mathbf{0}$, or $\bar{\mathbf{w}}$ is an eigenvector of $\mathbf{C}$ corresponding to the eigenvalue $\bar{\lambda}$. In rules of the second class, the fixed points in the eigenvectors of $\mathbf{C}$ are in addition characterized by a specific value for the squared weight vector length $\bar{\mathbf{w}}^T \bar{\mathbf{w}}$, given in the fourth column of the table. The temporal derivative of the squared weight vector length $\Xi = \frac{1}{2}d(\mathbf{w}^T \mathbf{w})/dt = \mathbf{w}^T \dot{\mathbf{w}}$ for the differential equation is found in the fifth column. If Euler’s method is applied, the squared weight vector length changes according to

$$\mathbf{w}_{t+1}^T \mathbf{w}_{t+1} = \mathbf{w}_t^T \mathbf{w}_t + 2\gamma_t \mathbf{w}_t^T \dot{\mathbf{w}}_t + \gamma_t^2 \dot{\mathbf{w}}_t^T \dot{\mathbf{w}}_t, \quad (2)$$

where Ξ is part of the second-order term. Ξ is zero for the learning rules of the first class. For the learning rules of the second class, however, Ξ determines how

Table 1. MCA learning rules. The rule in the bottom row is introduced in this work.

rule	$\dot{\mathbf{w}}$, averaged form	$\dot{\mathbf{w}}$, online form	$\bar{\mathbf{w}}^T \bar{\mathbf{w}}$	$\Xi = \mathbf{w}^T \dot{\mathbf{w}}$
OJAn	$-\mathbf{C}\mathbf{w} + \frac{\mathbf{w}^T \mathbf{C}\mathbf{w}}{\mathbf{w}^T \mathbf{w}} \mathbf{w}$	$\frac{y}{\mathbf{w}^T \mathbf{w}} (y\mathbf{w} - [\mathbf{w}^T \mathbf{w}]\mathbf{x})$	0	
LUO	$-(\mathbf{w}^T \mathbf{w})\mathbf{C}\mathbf{w} + (\mathbf{w}^T \mathbf{C}\mathbf{w})\mathbf{w}$	$y(y\mathbf{w} - [\mathbf{w}^T \mathbf{w}]\mathbf{x})$	0	
EXIN	$-\frac{1}{\mathbf{w}^T \mathbf{w}} \mathbf{C}\mathbf{w} + \frac{\mathbf{w}^T \mathbf{C}\mathbf{w}}{(\mathbf{w}^T \mathbf{w})^2} \mathbf{w}$	$\frac{y}{(\mathbf{w}^T \mathbf{w})^2} (y\mathbf{w} - [\mathbf{w}^T \mathbf{w}]\mathbf{x})$	0	
DOUG	$-(\mathbf{w}^T \mathbf{w})^2 \mathbf{C}\mathbf{w} + (\mathbf{w}^T \mathbf{C}\mathbf{w})\mathbf{w}$	$y(y\mathbf{w} - [\mathbf{w}^T \mathbf{w}]^2 \mathbf{x})$	1	$y^2 \mathbf{w}^T \mathbf{w} (1 - \mathbf{w}^T \mathbf{w})$
CHEN	$-(\mathbf{w}^T \mathbf{w})\mathbf{C}\mathbf{w} + (\mathbf{w}^T \mathbf{C}\mathbf{w} + 1 - \mathbf{w}^T \mathbf{w})\mathbf{w}$	$y(y\mathbf{w} - [\mathbf{w}^T \mathbf{w}]\mathbf{x}) + (1 - \mathbf{w}^T \mathbf{w})\mathbf{w}$	1	$\mathbf{w}^T \mathbf{w} (1 - \mathbf{w}^T \mathbf{w})$
OJA+	$-\mathbf{C}\mathbf{w} + (\mathbf{w}^T \mathbf{C}\mathbf{w} + 1 - \mathbf{w}^T \mathbf{w})\mathbf{w}$	$y(y\mathbf{w} - \mathbf{x}) + (1 - \mathbf{w}^T \mathbf{w})\mathbf{w}$	1	$(\mathbf{w}^T \mathbf{w} - y^2)(1 - \mathbf{w}^T \mathbf{w})$
OJA*	$-\mathbf{C}\mathbf{w} + (\mathbf{w}^T \mathbf{C}\mathbf{w} + \beta[1 - \mathbf{w}^T \mathbf{w}])\mathbf{w}$	$y(y\mathbf{w} - \mathbf{x}) + \beta(1 - \mathbf{w}^T \mathbf{w})\mathbf{w}$	1	$(\beta \mathbf{w}^T \mathbf{w} - y^2)(1 - \mathbf{w}^T \mathbf{w})$
XU*	$-(2 - \mathbf{w}^T \mathbf{w})\mathbf{C}\mathbf{w} + (\mathbf{w}^T \mathbf{C}\mathbf{w} + 2\beta[1 - \mathbf{w}^T \mathbf{w}])\mathbf{w}$	$y(y\mathbf{w} - [2 - \mathbf{w}^T \mathbf{w}]\mathbf{x}) + 2\beta(1 - \mathbf{w}^T \mathbf{w})\mathbf{w}$	1	$2(\beta \mathbf{w}^T \mathbf{w} - y^2)(1 - \mathbf{w}^T \mathbf{w})$
FENG	$-(\mathbf{w}^T \mathbf{w})\mathbf{C}\mathbf{w} + \mathbf{w}$	$-(\mathbf{w}^T \mathbf{w})y\mathbf{x} + \mathbf{w}$	$\bar{\lambda}^{-1}$	$(1 - y^2)\mathbf{w}^T \mathbf{w}$
AMEX	$-\mathbf{C}\mathbf{w} + \frac{1}{\mathbf{w}^T \mathbf{w}} \mathbf{w}$	$-y\mathbf{x} + \frac{\mathbf{w}}{\mathbf{w}^T \mathbf{w}}$	$\bar{\lambda}^{-1}$	$1 - y^2$
ZLlog	$-\mathbf{C}\mathbf{w} - \ln(\mathbf{w}^T \mathbf{w})\mathbf{w}$	$-y\mathbf{x} - \ln(\mathbf{w}^T \mathbf{w})\mathbf{w}$	$e^{-\bar{\lambda}}$	$-\ln(\mathbf{w}^T \mathbf{w})\mathbf{w}^T \mathbf{w} - y^2$
ZLdiff	$-\mathbf{C}\mathbf{w} + (\beta - \mathbf{w}^T \mathbf{w})\mathbf{w}$	$-y\mathbf{x} + (\beta - \mathbf{w}^T \mathbf{w})\mathbf{w}$	$\beta - \bar{\lambda}$	$(\beta - \mathbf{w}^T \mathbf{w})\mathbf{w}^T \mathbf{w} - y^2$
WANG	$-\mathbf{C}\mathbf{w} + \beta(1 - \mathbf{w}^T \mathbf{w})\mathbf{w}$	$-y\mathbf{x} + \beta(1 - \mathbf{w}^T \mathbf{w})\mathbf{w}$	$1 - \frac{\bar{\lambda}}{\beta}$	$\beta(1 - \mathbf{w}^T \mathbf{w})\mathbf{w}^T \mathbf{w} - y^2$
here	$-(2\mathbf{w}^T \mathbf{w} - 1)\mathbf{C}\mathbf{w} + (\mathbf{w}^T \mathbf{C}\mathbf{w})\mathbf{w}$	$y(y\mathbf{w} - [2\mathbf{w}^T \mathbf{w} - 1]\mathbf{x})$	1	$y^2(1 - \mathbf{w}^T \mathbf{w})$

the squared weight vector length converges towards the value $\bar{\mathbf{w}}^T \bar{\mathbf{w}}$.²

Depending on Ξ , the rules of the second class may differ in their convergence properties. In DOUG and CHEN, the *sign* of the temporal derivative of the weight vector length is independent of the random vector y , but only depends on the sign of $1 - \mathbf{w}^T \mathbf{w}$. This shows the self-stabilization property: if $\mathbf{w}^T \mathbf{w} < 1$, the weight vector length increases, if $\mathbf{w}^T \mathbf{w} > 1$, it decreases, thus in each step, the weight vector moves towards unity. In OJA+, OJA*, XU*, FENG, AMEX, ZLlog, ZLdiff, and WANG, however, the sign depends on the random vector y , so that deviations in the weight vector length from the value in the fixed points are not reduced in every step.

A closer inspection of the temporal development of the weight vector length is necessary in these rules.² Near convergence, when $\mathbf{w} \approx \bar{\mathbf{w}}$, MCA rules have $E\{y^2\} = E\{(\mathbf{w}^T \mathbf{x})^2\} = \mathbf{w}^T \mathbf{C}\mathbf{w} \approx \bar{\lambda}$. Since $\bar{\lambda}$ is the minor eigenvalue of $\mathbf{C}$, $E\{y^2\}$ is small. OJA+, OJA*, and XU* show self-stabilization at least near the fixed point: as long as $\bar{\lambda} < \beta$ (OJA+: $\beta = 1$), the first factor of the equation is on average positive and Ξ is dominated by the factor $1 - \mathbf{w}^T \mathbf{w}$ as in DOUG and CHEN. For FENG, this analysis reveals the possibility of a sudden divergence of the weight vector length. For $\bar{\lambda} \ll 1$, we can neglect the term y^2 in Ξ , since $E\{y^2\} \approx \bar{\lambda}$. We obtain

$\Xi \approx \mathbf{w}^T \mathbf{w} \approx \bar{\lambda}^{-1}$, thus Ξ is large and therefore the weight vector length diverges.² This problem is supposedly mended in AMEX since the term $\mathbf{w}^T \mathbf{w}$ is missing in Ξ . ZLlog, ZLdiff, and WANG require further analysis.

3. Novel MCA Learning Rule

In the bottom row of Table 1, the novel MCA learning rule introduced in this paper is shown. From the averaged form

$$\dot{\mathbf{w}} = -(2\mathbf{w}^T \mathbf{w} - 1)\mathbf{C}\mathbf{w} + (\mathbf{w}^T \mathbf{C}\mathbf{w})\mathbf{w} \quad (3)$$

the fixed points of this rule can be obtained:

$$\mathbf{C}\bar{\mathbf{w}} = \bar{\lambda}\bar{\mathbf{w}}, \quad \text{with} \quad \bar{\lambda} = \frac{\bar{\mathbf{w}}^T \mathbf{C}\bar{\mathbf{w}}}{2\bar{\mathbf{w}}^T \bar{\mathbf{w}} - 1}. \quad (4)$$

We either have $\bar{\mathbf{w}} = \mathbf{0}$, or $\bar{\mathbf{w}}$ is an eigenvector of $\mathbf{C}$. By merging these two equations we obtain $\bar{\mathbf{w}}^T \bar{\mathbf{w}} = 1$ if $\bar{\mathbf{w}}$ is an eigenvector of $\mathbf{C}$, thus (3) has fixed points in the unit-length eigenvectors of $\mathbf{C}$. The Jacobian $\mathbf{J}(\mathbf{w}) = \partial \dot{\mathbf{w}} / \partial \mathbf{w}$ of (3) is

$$\mathbf{J}(\mathbf{w}) = -4\mathbf{C}\mathbf{w}\mathbf{w}^T - 2\mathbf{w}^T \mathbf{w}\mathbf{C} + \quad (5)$$

$$\mathbf{C} + (\mathbf{w}^T \mathbf{C}\mathbf{w})\mathbf{I} + 2\mathbf{w}\mathbf{w}^T \mathbf{C}. \quad (6)$$

In all eigenvector fixed points, we get

$$\mathbf{J}(\bar{\mathbf{w}}) = -\mathbf{C} + \bar{\lambda}\mathbf{I} - 2\bar{\lambda}\bar{\mathbf{w}}\bar{\mathbf{w}}^T. \quad (7)$$

$\mathbf{J}$ has the same eigenvalues as $\mathbf{J}^* = \mathbf{W}^T \mathbf{J} \mathbf{W}$, where $\mathbf{W}$ is the orthonormal $n \times n$ matrix containing all n eigenvectors of $\mathbf{C}$ in its columns. With $\bar{\lambda} = \lambda_1$ we obtain

$$\mathbf{J}^* = -\bar{\lambda} \mathbf{I} + \lambda_1 \mathbf{I} - 2\lambda_1 \mathbf{e}_1 \mathbf{e}_1^T. \quad (8)$$

The diagonal $n \times n$ matrix $\bar{\lambda} \mathbf{I}$ contains all eigenvalues of $\mathbf{C}$, and $\mathbf{e}_1 = (1, 0, \dots, 0)^T$. The eigenvectors of $\mathbf{J}^*$ are determined from

$$\det(\mathbf{J}^* - \alpha \mathbf{I}) = (-2\lambda_1 - \alpha) \prod_{j=2}^n (-\lambda_j + \lambda_1 - \alpha) = 0, \quad (9)$$

which yields an eigenvalue $\alpha_1 = -2\lambda_1 < 0$, and an MCA condition $\alpha_j = -\lambda_j + \lambda_1 < 0$, i.e. $\lambda_1 < \lambda_j$, $j = 2, \dots, n$. Only fixed points where the MCA condition is fulfilled are attractors, all others are repellers or saddle points. For the remaining fixed point $\bar{\mathbf{w}} = 0$, we obtain $\mathbf{J}(\bar{\mathbf{w}}) = \mathbf{C}$ and $\mathbf{J}^*(\bar{\mathbf{w}}) = \bar{\lambda} \mathbf{I}$. If at least one eigenvalue of $\mathbf{C}$ is strictly positive, this fixed point is not stable. We conclude that the rule converges towards the minor eigenvector $\bar{\mathbf{w}} = \mathbf{w}_1$ corresponding to the minor eigenvalue λ_1 .

The online version of the novel rule is obtained as described in Sec. 2:

$$\dot{\mathbf{w}} = y(y\mathbf{w} - [2\mathbf{w}^T \mathbf{w} - 1]\mathbf{x}). \quad (10)$$

We see that $\Xi = \mathbf{w}^T \dot{\mathbf{w}} = y^2(1 - \mathbf{w}^T \mathbf{w})$. As the rules DOUG and CHEN, the novel rule is strictly self-stabilizing. This holds for an exact solution of the differential equation, and to the first order in γ_t if solved by Euler's method; see equation (2).

4. Simulations

In this letter, we demonstrate that the novel MCA learning rule actually shows the predicted behavior, and compare the performance of the novel rule with the performance of OJA+, DOUG, and CHEN, which show the closest structural similarities and comparable behavior. We present results for synthetic three-dimensional and natural 64-dimensional data.

4.1. Synthetic Data

In the three-dimensional space ($n = 3$), 1000 data points were drawn from a uniform distribution inside a cuboid of size $500 \times 100 \times 200$ which was rotated around the x , y , and z axis by angles of 20° , 10° ,

and 15° , in that order. The distribution's covariance matrix has the eigenvalues 20451, 863, and 3173.

The weight vector was initialized with a random unit-length vector and $t_{max} = 10\,000$ training steps were executed for both rules. With the novel rule and the rule DOUG, the step size γ_t starts at $\gamma_0 = 10^{-5}$ and decays exponentially towards zero with a final value $\gamma_{t_{max}} = 10^{-7}$. The small values of the step size are necessary, since the distribution has large eigenvalues and in both rules the eigenvalues of the Jacobian linearly depend on the eigenvalues of the distribution (see discussion in Sec. 5.4). Since the smallest eigenvalue violates the condition $\lambda_1 < 1$ of OJA+, the data vectors had to be scaled by a factor 0.01 for the experiment with OJA+; in this case, the learning rate range was $\gamma_0 = 10^{-1}$ and $\gamma_{t_{max}} = 10^{-3}$. The same step size and scaling as for OJA+ were also used for CHEN, since the rule proved to be unstable with the parameters used for the novel rule and DOUG. Although a local analysis will not be sufficient to study this problem, it could be related to the fact that in the CHEN rule, one of the eigenvalues of the Jacobian does not depend on the eigenvalues of $\mathbf{C}$, so that the convergence speed in the corresponding eigendirection may differ considerably from the other eigendirections of the Jacobian (see discussion in Sec. 5.4).

Fig. 1. Synthetic data: Time course of the weight vector length.

Fig. 1 shows the time course of the length of the weight vector $\|\mathbf{w}_t\|$ for the four rules (the fifth rule labeled “coupled rule” is discussed in Sec. 5.3). In all rules, the length slightly deviates from 1 in an initial phase, but afterwards approaches 1. The initial deviation is an effect of the second-order terms (γ_t^2) in equation (2); later, with decreasing step size,

the first-order terms become dominating. OJA+ and CHEN generally produce smaller deviations than the novel rule and DOUG, and the deviations are decreasing earlier. The slow return to 1 in the novel rule and in DOUG can be explained by the fact that with $E\{y^2\}$ being small near convergence the changes of the weight vector length become smaller and initial deviations are only slowly reduced; see Sec. 5.2.

Fig. 2. Synthetic data: Time course of the projection of the weight vector.

Fig. 2 shows the projection $\mathbf{w}_1^T \mathbf{w}_t / \|\mathbf{w}_t\|$ of the normalized weight vector onto the real minor eigenvector (with unit length) of the distribution for the four rules. The behavior does not differ noticeably between the rules. The projection quickly approaches a value of 1, i.e. the weight vector aligns with the real minor eigenvector. From a comparison of Fig. 1 and 2 it is visible that in the novel rule and in DOUG the direction of the weight vector converges much earlier than its length.

4.2. Image Data

The learning rules were also applied to a medium-dimensional data distribution formed by 20 000 windows of size 8×8 ($n = 64$) extracted from a gray-scale image of a natural scene (with gray values in $[0, 1]$). It is known that the eigenvalues of this distribution extend over a range of 3-4 orders of magnitude, from approximately $5 \cdot 10^{-4}$ to 2. Here we used the same training parameters $\gamma_0 = 0.1$, $\gamma_{t_{max}} = 0.001$, and $t_{max} = 200\,000$ for the four rules without scaling of the data vectors.

The time course of the weight vector length $\|\mathbf{w}_t\|$ for these data is visualized in Fig. 3 (sampling dis-

tance: 1000 steps). As in the previous example, the weight vector length initially deviates from unity and then returns to unity in both the novel rule and in DOUG, but stays close to 1 for OJA+ and CHEN (the curves for OJA+, CHEN, and the coupled rule are overlapping); see discussion in Sec. 5.2.

Fig. 3. Image data: Time course of the weight vector length.

Fig. 4 shows the final weight vectors for 10 different runs with the same data set and the same presentation order but different weight vector initialization. All final weight vectors exhibit the high image frequencies characteristic for minor eigenvectors of image data. A qualitative difference between the four rules is not visible. The differences between the runs can be explained by the fact that the minor eigenvalues of the distribution are very close to each other, so that the corresponding eigenvalues of the Jacobian $\alpha_j = -\lambda_j + \lambda_1$ are small and therefore the movement in the corresponding directions slow. Moreover, MCA learning rules are generally affected by the problem that the eigenvalues of the Jacobian vary over the same (relative) range as the eigenvalues of the covariance matrix; see the discussion in Sec. 5.4. Thus practically the final weight vector seems to be an arbitrary vector in the subspace spanned by some of the minor eigenvectors, and depends on its initial value.

5. Discussion

5.1. Relation to the Unified Stabilization Approach

Chen and Amari¹¹ presented a unified stabiliza-

Fig. 4. Image data: Final weight vectors of 10 runs.

tion approach which subsumes a number of learning rules for principal and minor component (or subspace) analysis; the following discussion is restricted to single-neuron minor component analyzers. As mentioned in Sec. 1, in the learning rules of the first class (like LUO) the weight vector length does not deviate from its initial value. However, this only holds for the differential equation, but not for the numerical solution which is plagued by “sudden divergence”.^{11,2} As a solution to this problem, Chen and Amari suggest to add suitable penalty terms to learning rules of the first class. The task of these penalty terms is to reduce the deviation of the weight vector length from unity. The rules resulting from this modification belong to the second class.

Starting from the LUO rule, the DOUG rule is obtained by adding the penalty term $(1 - \mathbf{w}^T \mathbf{w})(\mathbf{w}^T \mathbf{w})\mathbf{C}\mathbf{w}$, and the CHEN rule can be derived by adding the penalty term $(1 - \mathbf{w}^T \mathbf{w})\mathbf{w}$.¹¹ The novel rule presented here turns out to be the sum of LUO and a penalty term $(1 - \mathbf{w}^T \mathbf{w})\mathbf{C}\mathbf{w}$. Chen and Amari show that adding the same penalty term to a PCA rule which is equal to LUO but has the opposite sign leads to Oja’s PCA rule.¹⁵ Because of this relationship, the methods for stability analysis used by Chen and Amari can also be applied to the novel rule suggested in this paper.

5.2. Self-Stabilization Property

A characteristic property influencing the behavior of rules of the second class is the temporal derivative of the squared weight vector length Ξ (see Table 1). If the *sign* of Ξ depends on the random output

y — as it is the case for all rules of the second class with the exception of DOUG, CHEN, and the rule presented here —, the weight vector length may in principle fluctuate or diverge,² and a closer inspection of the temporal development of the weight vector length is necessary. In DOUG, CHEN, and the novel rule, the sign of Ξ only depends on $1 - \mathbf{w}^T \mathbf{w}$, so that, in an exact solution of the differential equation, deviations from unity are reduced in each step, independently of the currently presented input vector.

In the numerical solution, however, we see an initial phase in which the vector length slightly deviates from unity. This can be explained by including the second-order terms in the analysis which can obviously not be fully neglected at the beginning of the training process when the step size is still large; see equation (2). Moreover, in both DOUG and the novel rule there appears to be a residual deviation of the weight vector length from unity (see Fig. 1 and 3). In both rules, Ξ includes a factor y^2 ; in contrast, this factor is not contained in Ξ for the CHEN rule. Since near convergence we have $E\{y^2\} \approx \bar{\lambda}$ (with $\bar{\lambda}$ being the minor eigenvalue), and thus the factor y^2 is on average small, DOUG and the novel rule are slower in reducing deviations of the weight vector length from unity than CHEN and fail to fully compensate for the initial deviation.

5.3. A Coupled MCA Rule

This disadvantage of the novel rule can be overcome by choosing a penalty term $\lambda^{-1}(1 - \mathbf{w}^T \mathbf{w})\mathbf{C}\mathbf{w}$, where λ is an eigenvalue estimate λ updated along

with the eigenvector estimate in a coupled learning rule system²⁰:

$$\dot{\mathbf{w}} = y(y\mathbf{w} - [\{1 + \lambda^{-1}\}\mathbf{w}^T\mathbf{w} - \lambda^{-1}]\mathbf{x}) \quad (11)$$

$$\dot{\lambda} = y^2 - \lambda. \quad (12)$$

A similar modification is possible for the DOUG rule. For this system we have $\Xi = y^2\lambda^{-1}(1 - \mathbf{w}^T\mathbf{w})$. In the vicinity of the minor eigenvector and eigenvalue we see that $E\{y^2\lambda^{-1}\} \approx 1$. Fig. 1 shows that in the coupled learning rule the time course of the weight vector length resembles that of OJA+ and CHEN. With respect to the convergence of the weight vector, Fig. 2 shows no difference to the other rules. The eigenvalue estimate converges to the minor eigenvalue of $\mathbf{C}$ (not shown). When applied to the image data (same parameters as before; initial eigenvalue estimate $\lambda_0 = 1.0$), the weight vector length stays close to unity as in OJA+ and CHEN; see Fig. 3. The modification of the penalty term can apparently solve the problem of the residual deviation of the novel rule.

However, the improvement in the stabilization of the weight vector length comes at the price of impaired overall robustness: in one of the 10 runs in figure 4, the coupled learning rule diverged in the initial phase of the learning process (white box in the diagram). In the coupled system (11, 12), the Jacobian has the eigenvalues $\alpha_1 = -2$ and $\alpha_j = -\lambda_j + \lambda_1$, $j > 1$. Thus, as in the CHEN rule, one eigenvector of the Jacobian does not depend on the eigenvalues of $\mathbf{C}$, which appears to have a relation to stability problems far from the fixed point (see Sec. 4.1).

The coupled rule does not lead to an improved overall convergence of the weight vectors as can be seen in Fig. 4.

5.4. Convergence Speed

A major problem that should affect all MCA learning rules in Table 1 is the convergence speed. It depends on the eigenvalues of the covariance matrix α_j which in MCA rules necessarily contain the factor $-\lambda_j + \lambda_1$ for $j > 1$. This factor guarantees that only minor eigenvectors are stable fixed points, while all other fixed points are repellers or saddles. The speed of convergence in the different eigendirections of the system therefore varies with the (relative) range of all other eigenvalues of the covariance matrix.²⁰ Especially in problems where the range covered by the

eigenvalues of the covariance matrix is large, it may become difficult to select an appropriate step size γ_t that ensures fast convergence but still guarantees the stability of Euler's method. This is one reason that, in our second example (PCA of image windows), the final weight vector still depends on its initial value for all rules tested here, despite the long training time.

The convergence speed in the first eigendirection depends on the eigenvalue α_1 of the Jacobian. For the tested rules we find: OJA+ $\alpha_1 = -2(1 - \lambda_1)$, DOUG $\alpha_1 = -2\lambda_1$, CHEN $\alpha_1 = -2$, novel rule (10) $\alpha_1 = -2\lambda_1$, coupled rule (11, 12) $\alpha_1 = -2$. As mentioned before, both CHEN and the coupled rule have a stability problem far from the fixed point, and in both rules we find that α_1 does not depend on the eigenvalues of $\mathbf{C}$. However, a local analysis as applied throughout this paper will not be sufficient to study this problem.

6. Conclusion

The novel minor component learning rule introduced in this paper was shown to self-stabilize the weight vector length. The rule is related to the rules CHEN and DOUG and can be derived from the unified stabilization approach suggested by Chen and Amari. In contrast to the OJA+ rule, the novel rule is stable independently of the minor eigenvalue. In its discrete form, the novel rule shows a small residual deviation of the weight vector length from unity. In a coupled rule obtained from a modification of the novel rule, the residual deviation disappears, however at the price of impaired stability.

Acknowledgments

The author would like to thank the anonymous reviewer for pointing out the relation of the novel learning rule to the "unified stabilization approach", and A. Könies for his help and comments on the first version of this paper.

References

1. S. Ouyang, Z. Bao, G.-S. Liao, and P. C. Ching, "Adaptive minor component extraction with modular structure," *IEEE Trans. Signal Processing*, **49**, 2127–2137 (2001).
2. G. Cirrincione, M. Cirrincione, J. Hérault, and S. van Huffel, "The MCA EXIN neuron for the minor com-

- ponent analysis," *IEEE Trans. Neural Networks*, **13**, 160–186 (2002).
3. L. Xu, E. Oja, and C. Y. Suen, "Modified hebbian learning for curve and surface fitting," *Neural Networks*, **5**, 441–457 (1992).
4. F. Luo, R. Unbehauen, and A. Cichocki, "A minor component analysis algorithm," *Neural Networks*, **10**, 291–297 (1997).
5. T. Chen, S. Amari, and Q. Lin, "A unified algorithm for principal and minor components extraction," *Neural Networks*, **11**, 385–390 (1998).
6. F.-L. Luo and R. Unbehauen, "Comments on: A unified algorithm for principal and minor components extraction," *Neural Networks*, **12**, 393 (1999).
7. A. Taleb and G. Cirrincione, "Against the convergence of the minor component analysis neurons," *IEEE Trans. Neural Networks*, **10**, 207–210 (1999).
8. E. Oja, "Principal components, minor components, and linear neural networks," *Neural Networks*, **5**, 927–935 (1992).
9. D. Z. Feng, Z. Bao, and L. C. Jiao, "Total least mean squares algorithm," *IEEE Trans. Signal Processing*, **46**, 2122–2130 (1998).
10. S. C. Douglas, S.-Y. Kung, and S. Amari, "A self-stabilized minor subspace rule," *IEEE Signal Processing Letters*, **5**, 328–330 (1998).
11. T. Chen and S. Amari, "Unified stabilization approach to principal and minor components extraction," *Neural Networks*, **14**, 1377–1387 (2001).
12. S. Ouyang, Z. Bao, and G. Liao, "Adaptive step-size minor component extraction algorithm," *Electronics Letters*, **35**, 443–444 (1999).
13. Q. Zhang and Y.-W. Leung, "A class of learning algorithms for principal component analysis and minor component analysis," *IEEE Trans. Neural Networks*, **11**, 200–204 (2000).
14. L. Wang and J. Karhunen, "A unified neural bigradient algorithm for robust PCA and MCA," *Int. J. Neural Systems*, **7**, 53–67 (1996).
15. E. Oja, "A simplified neuron model as principal component analyzer," *J. Math. Biol.*, **15**, 267–273 (1982).
16. L. Xu, "Least mean square error reconstruction principle for self-organizing neural-nets," *Neural Networks*, **6**, 627–648 (1993).
17. T. Chen, "Modified Oja's algorithms for principal subspace and minor subspace extraction," *Neural Processing Letters*, **5**, 105–110 (1997).
18. K. I. Diamantaras and S. Y. Kung, *Principal Component Neural Networks. Theory and Applications*. John Wiley & Sons (1996).
19. A. Hyvärinen, J. Karhunen, and E. Oja, *Independent Component Analysis*. John Wiley & Sons (2001).
20. R. Möller, A. Köries, "Coupled Principal Component Analysis," *IEEE Trans. Neural Networks*, **14** (2003), in press