Available online at www.sciencedirect.com
SCIENCE DIRECT®
Neurocomputing 69 (2006) 1582–1590
NEUROCOMPUTING
www.elsevier.com/locate/neucom

First-order approximation of Gram–Schmidt orthonormalization beats deflation in coupled PCA learning rules

Ralf Möller

Computer Engineering Group, Faculty of Technology, Bielefeld University, D-33954 Bielefeld, Germany

Received 8 February 2005; received in revised form 24 May 2005; accepted 14 June 2005
Available online 27 October 2005
Communicated by B. Hammer

Abstract

In coupled learning rules for principal component analysis, eigenvectors and eigenvalues are simultaneously estimated in a coupled system of equations. Coupled single-neuron rules have favorable convergence properties. For the estimation of multiple eigenvectors, orthonormalization methods have to be applied, either full Gram–Schmidt orthonormalization, its first-order approximation as used in Oja's stochastic gradient ascent algorithm, or deflation as in Sanger's generalized Hebbian algorithm. This paper reports the observation that a first-order approximation of Gram–Schmidt orthonormalization is superior to the standard deflation procedure in coupled learning rules. The first-order approximation exhibits a smaller orthonormality error and produces eigenvectors and eigenvalues of better quality. This improvement is essential for applications where multiple principal eigenvectors have to be estimated simultaneously rather than sequentially. Moreover, loss of orthonormality may have an harmful effect on subsequent processing stages, like the computation of distance measures for competition in local PCA methods.

Keywords: Principal component analysis; Coupled learning rules; Orthonormalization; Gram–Schmidt method; Deflation

1. Introduction

In the wake of the initial contribution by Oja [9], a plethora of learning rules for principal component analysis (PCA) have been suggested. Several authors introduced modifications of the original Oja rule where the learning rate for the weight update is adjusted according to an eigenvalue estimate, e.g. projection approximation subspace tracking [16], learning rules based on recursive least square approaches [1,5,13,3], and the adaptive learning algorithm [2]. A number of these "coupled" learning rules, which simultaneously estimate eigenvectors and eigenvalues in a coupled system of equations, can be derived from a common framework by applying Newton's method to an information criterion [8]. Coupled learning rules are known to exhibit improved convergence, leading to better eigenvector estimates or requiring less training steps.

If more than one principal eigenvector has to be estimated, some decorrelation method has to be applied. The statement of Hornik and Kuan that hierarchical decorrelation methods perform better than symmetrical methods is still valid [4]; it is an additional advantage of hierarchical methods that the resulting eigenvectors are sorted with respect to the corresponding eigenvalues. A number of hierarchical methods are descendants of the Gram–Schmidt orthonormalization procedure. In these methods, neurons are arranged in a chain, and the weight modification of each chain element depends on the previous stages in the chain. Unfortunately, the Gram–Schmidt method has a complexity in the order of nm^2 if m principal eigenvectors of dimension n have to be determined. Therefore, approximations of the Gram–Schmidt procedure have been introduced which reduce the complexity to an order of nm. A first-order approximation of the Gram–Schmidt method for small learning rates is the basis of the stochastic gradient ascent algorithm suggested by Oja and Karhunen [11,12]. The deflation principle introduced by Sanger for his generalized Hebbian learning

E-mail address: ftupindr@ti.uni-bielefeld.de.
URL: http://www.ti.uni-bielefeld.de.

0925-2312/$ - see front matter © 2005 Elsevier B.V. All rights reserved.
doi:10.1016/j.neucom.2005.06.016

method can be interpreted as further step of approximation [14,4,10]. Other attempts have been made to stick to full Gram–Schmidt orthonormalization, but reduce the effort by a factor of 2 by interlocking learning rule and orthonormalization [6].

Approximations of the Gram–Schmidt procedure can entail a severe loss of orthonormality in the later stages of the chain, specifically when all neurons in the chain are trained simultaneously. This affects the convergence speed of the chain as a whole [6], and has a negative impact on subsequent processing stages which rely on orthonormality. In local PCA approaches, for example, several PCA networks may compete through a distance measure like the Mahalanobis metric which is sensitive to deviations from orthonormality in the weight vector set [7].

A dependency of the quality of eigenvector and eigenvalue estimates on the specific deflation procedure was noticed before in coupled learning rules [8]. In this paper it is shown that the first-order approximation of the Gram–Schmidt method is superior to the standard deflation method in coupled learning rules, with respect to both convergence and orthonormality error. A coupled multi-neuron learning rule is derived from a general equation for the first-order approximation. Moreover, the effect of additional explicit weight vector normalization to unit length is investigated. Plain Hebbian learning rules and coupled Hebbian learning rules are briefly recapitulated in Section 2. Section 3 describes Gram–Schmidt orthonormalization, its first-order approximation, and deflation, and derives specific equations for Hebbian and coupled learning. The results on experiments with medium-dimensional ($n = 64$) and with high-dimensional image data ($n = 16\,384$) are reported in Section 4. Results and implementation issues are discussed in Section 5.

2. Plain and coupled Hebbian learning rules

To simplify the notation, only a single learning step is considered. The m weight vectors before learning are denoted as $\mathbf{v}_k$ with $k = 1, \ldots, m$, the weight modification as $\mathbf{m}_k$, and the modified weights after the learning step (but before orthonormalization) as $\mathbf{w}_k$, thus $\mathbf{w}_k = \mathbf{v}_k + \mathbf{m}_k$. Note that for the approximations described below, $\mathbf{m}_k$ is supposed to be small (as an effect of small learning rates). In *plain Hebbian rules*, the weight change is defined as

$$\mathbf{m}_k = \gamma \mathbf{x}\mathbf{x}^{\mathrm{T}}\mathbf{v}_k = \gamma y_k \mathbf{x}, \tag{1}$$

where $\mathbf{x}$ is the input vector, $y_k = \mathbf{v}_k^{\mathrm{T}}\mathbf{x}$ is the output value of neuron k in the chain, and γ is the small learning rate. Thus, the modified weight vector is obtained from

$$\mathbf{w}_k = \mathbf{v}_k + \gamma y_k \mathbf{x}. \tag{2}$$

In *coupled Hebbian rules*, a modification of Hebbian rules, the learning rate γ is divided by the current eigenvalue estimate λ_k

$$\mathbf{m}_k = \gamma \lambda_k^{-1} \mathbf{x}\mathbf{x}^{\mathrm{T}}\mathbf{v}_k = \gamma \lambda_k^{-1} y_k \mathbf{x} \tag{3}$$

leading to the update equation

$$\mathbf{w}_k = \mathbf{v}_k + \gamma \lambda_k^{-1} y_k \mathbf{x}. \tag{4}$$

Here, the eigenvalue estimate λ_k is a temporal average of y_k^2 obtained by a low-pass filter [8]; the eigenvalue estimate of the next time step λ_k' is computed from

$$\lambda_k' = \lambda_k + \gamma(y_k^2 - \lambda_k). \tag{5}$$

3. Orthonormalization methods

From the modified weight vectors $\mathbf{w}_k$, exact Gram–Schmidt orthonormalization produces an orthonormal version $\mathbf{u}_k$, while in the two approximations, we use $\hat{\mathbf{u}}_k$ instead to indicate that the vectors are only approximately orthonormal. The weight vectors of the next learning step $\mathbf{v}_k'$ are either obtained by directly assigning the orthonormalized vector

$$\mathbf{v}_k' = \mathbf{u}_k \quad \text{or} \quad \mathbf{v}_k' = \hat{\mathbf{u}}_k \tag{6}$$

or by assigning a vector normalized to unit length

$$\mathbf{v}_k' = \hat{\mathbf{u}}_k / \|\hat{\mathbf{u}}_k\|. \tag{7}$$

Note that all weight vectors are updated simultaneously.

3.1. Gram–Schmidt orthonormalization

In this notation, Gram–Schmidt orthonormalization can be written as

$$\mathbf{u}_k^* = \mathbf{w}_k - \sum_{j=1}^{k-1} (\mathbf{u}_j^{\mathrm{T}}\mathbf{w}_k)\mathbf{u}_j, \quad \mathbf{u}_k = \frac{\mathbf{u}_k^*}{\|\mathbf{u}_k^*\|}. \tag{8}$$

The computational complexity is proportional to nm^2, where n is the complexity of all vector operations (scalar product, sum, normalization) while the factor m^2 results from the sum in (8) which has to be recomputed for each weight vector index k.

3.2. First-order approximation

For arbitrary learning rules, the first-order approximation of Eq. (8) derived by Oja and Karhunen [11,12] is given by

$$\hat{\mathbf{u}}_k = \mathbf{v}_k + \mathbf{m}_k - \sum_{j=1}^{k-1} (\mathbf{v}_j^{\mathrm{T}}\mathbf{m}_k + \mathbf{v}_k^{\mathrm{T}}\mathbf{m}_j)\mathbf{v}_j - \mathbf{v}_k^{\mathrm{T}}\mathbf{m}_k\mathbf{v}_k. \tag{9}$$

The validity of this approximation can be proven by induction under the assumption that quadratic products of components of vectors $\mathbf{m}_i$ and $\mathbf{m}_j$ almost vanish. Furthermore, it is assumed that the previous weight vectors $\mathbf{v}_k$ are close to orthonormality, thus $\mathbf{v}_i^{\mathrm{T}}\mathbf{v}_j \approx \delta_{ij}$ (with δ_{ij} denoting Kronecker's delta). Also the weight vector normalization in Eq. (8) is approximated to the first order.

It is not visible how Eq. (9) would in general reduce the effort compared to a Gram–Schmidt orthonormalization since still there is a sum which has to be determined anew