Neural Approaches to Image Compression/ Decompression Using PCA based Learning Algorithms

Luminita State[1], Catalina Cocianu[2], Panayiotis Vlamos[3] and Doru Constantin[1]

[1]Department of Computer Science, University of Pitesti, Pitesti, Romania

[2]Department of Computer Science, Academy of Economic Studies, Bucharest, Romania

[3]Department of Computer Science, Ionian University, Corfu, Greece

Abstract. Principal Component Analysis is a well-known statistical method for feature extraction, data compression and multivariate data projection. Aiming to obtain a guideline for choosing a proper method for a specific application we developed a series of simulations on some the most currently used PCA algorithms as GHA, Sanger variant of GHA and APEX. The paper reports the conclusions experimentally derived on the convergence rates and their corresponding efficiency for specific image processing tasks.

1 Introduction

Principal component analysis allows the identification of a linear transform such that the axes of the resulted coordinate system correspond to the largest variability of the signal. The signal features corresponding to the new coordinate system are uncorrelated. One of the most frequently used method in the study of convergence properties corresponding to different stochastic learning PCA algorithms basically proceeds by reducing the problem to the analysis of asymptotic stability of the trajectories of a dynamic system whose evolution is described in terms of an ODE [5]. The Generalized Hebbian Algorithm (GHA) extends the Oja's learning rule for learning the first principal components. Aiming to obtain a guideline for choosing a proper method for a specific application we developed a series of simulations on some the most currently used PCA algorithms as GHA, Sanger variant of GHA and APEX.

2 Hebbian Learning in Feed-forward Architectures

The input signal is modeled as a wide-sense-stationary *n*-dimensional process $(X(t), t \geq 0)$ of mean 0 and covariance matrix $E(X(t)X(t)^T) = S$. We denote by $\Phi_1,...,\Phi_n$ a set of unit eigen-vectors of S indexed according to the decreasing order of their corresponding eigen-values $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$. The most informative

State L., Cocianu C., Vlamos P. and Constantin D. (2008).
Neural Approaches to Image Compression/Decompression Using PCA based Learning Algorithms.
In *Proceedings of the 8th International Workshop on Pattern Recognition in Information Systems*, pages 187-192

directions of the process $(X(t), t \geq 0)$ are given by $\Phi_1,...,\Phi_n$ and for any $m, 1 \leq m \leq n$ its LMS-optimal linear features are $\Phi_1,...,\Phi_m$. The architecture of a PCA neural network consists of the *n*-neuron input layer and the *m*-neuron computation layer. The aim is to develop an adaptive learning algorithm to encode asymptotically $\Phi_1,...,\Phi_m$ as values of the synaptic vectors $W_1,...,W_m$ of the neurons in the computation layer. Let $W(t) = (W_1(t),...,W_m(t))$ be the synaptic memory at the moment *t*, and let $Y(t) = (Y_1(t),...,Y_m(t))^T$ be the output of the computation layer, $1 \leq j \leq m$, $Y_j(t) = W_j^T(t)X(t)$. The Hebbian rule for learning the first principal component is, $W_1(k+1) = W_1(k) + \eta(k)X(k)Y_1(k)$, where the sequence of learning rates $(\eta(k))$ are taken such that the conditions of the Kushner theorem hold [5], $\sum_{k=1}^{\infty} \eta(k) = \infty$, $\lim_{k\to\infty} \eta(k) = 0$, there exists $p > 1$ such that $\sum_{k=1}^{\infty} (\eta(k))^p < \infty$. The normalized version of the Hebbian learning rule is,

$$W_1(k+1) = \frac{W_1(k) + \eta(k)X(k)Y_1(k)}{\|W_1(k) + \eta(k)X(k)Y_1(k)\|} \tag{1}$$

In order to get a local learning scheme a linearized version of (1) using first order approximation was proposed in [7] yielding to the cellebrated Oja's learning algorithm,

$$W_1(k+1) = W_1(k) + \eta(k)\left(X(k)Y_1(k) - Y_1^2(k)W_1(k)\right) \tag{2}$$

The Generalized Hebbian Algorithm (GHA) [3] is one of the first neural models for extracting multiple PCs. At any moment *t*, each neuron *j*, $j \geq 1$, receives two inputs, the original signal $X(t)$ and the deflated signal $X_j(t)$ and computes $Y_j(t) = W_j^T(t)X(t)$ and $\tilde{Y}_j(t) = W_j^T(t)\tilde{X}_j(t)$, $\tilde{X}_j(t) = \tilde{X}_{j-1}(t) - Y_{j-1}(t)W_{j-1}(t)$, $j \geq 2$. The GHA learning scheme is, for $2 \leq j \leq m$,

$$W_1(k+1) = W_1(k) + \eta(k)\left(X(k)Y_1(k) - Y_1^2(k)W_1(k)\right) \tag{3}$$

$$W_j(k+1) = W_j(k) + \eta(k)\left(\tilde{X}_j(k)\tilde{Y}_j(k) - \tilde{Y}_j^2(k)W_j(k)\right) \tag{4}$$

where $Y_j(k) = W_j^T(k)X(k)$, $\tilde{X}_j(k) = \tilde{X}_{j-1}(k) - Y_{j-1}(k)W_{j-1}(k) = \sum_{i=1}^{j-1} Y_i(k)W_i(k)$, and $\tilde{Y}_j(k) = W_j^T(k)\tilde{X}_j(k)$.

The variant proposed by Sanger [7] simplifies the learning process by using only output of each neuron in both, the synaptic learning scheme and the input deflation. The Sanger variant of GHA is, for $2 \leq j \leq m$,

$$W_1(k+1) = W_1(k) + \eta(k)\left(X(k)Y_1(k) - Y_1^2(k)W_1(k)\right) \tag{5}$$

$$W_j(k+1) = W_j(k) + \eta(k)\left(\tilde{X}_j(k)Y_j(k) - Y_j^2(k)W_j(k)\right) \tag{6}$$

where $Y_j(k) = W_j^T(k)X(k)$ and $\tilde{X}_j(k) = \tilde{X}_{j-1}(k) - Y_{j-1}(k)W_{j-1}(k)$ is the input deflated at the level of the *j*th neuron.

The APEX learning algorithm proposed in [2] generalizes the idea of lateral influences by imposing a certain learning process to the weights of lateral connections. The output of each neuron *j*, is computed from its own output and the effects of the outputs corresponding to all neurons *i*, $1 \le i \le j-1$, weighted by the coefficients $a_{ij}(t)$,

$$Y_j(t) = W_j^T(t)X(t) - \sum_{i=1}^{j-1} a_{ij}(t)Y_i(t) \quad j \ge 2 \tag{7}$$

The learning scheme for the local memories is essentially the Oja's learning rule taken for the transformed outputs Y_j,

$$W_j(t+1) = W_j(t) + \eta(t)\left[Y_j(t)X(t) - Y_j^2(t)W_j(t)\right] \tag{8}$$

The learning scheme for the weights of lateral connections is given by,

$$a_{ij}(t+1) = a_{ij}(t) + \eta(t)\left[Y_i(t)Y_j(t) - Y_j^2(t)a_{ij}(t)\right] \tag{9}$$

Note that the theoretical analysis [1], [2], [3], establishes the almost sure convergence to the principal components of the sequences of weight vectors generated by the above mentioned algorithms.

3 Recursive Least Square Learning Algorithm of the Principal Directions

Let $W_1(t\text{-}1)$ be the synaptic vector at the moment *t* and assume that the inputs are applied at the moments $t=0,1,2,\ldots$. If we denote by $X(k)$ the input at the moment *k*, then the output is $Y(k) = W_1(k\text{-}1)h_1(k) = W_1(k\text{-}1)W^T_1(k\text{-}1)\,X(k)$, where $h_1(k)=W^T_1(k\text{-}1)\,X(k)$ is the neural activation induced by the input. The mean error at the moment *t* is $J_1(t) = \sum_{k=1}^{t} \varepsilon^2(k)$, where $\varepsilon^2(k) = \|X(k) - Y(k)\|^2$. The aim is to determine $\hat{W}_1(t)$ minimizing $J_1(W_1(t))$ the overall error, when at each moment of time *k*, $1 \le k \le t$, the decompression is assumed as being performed using the filter $W_1(t)$, that is,

$$J_1(W_1(t)) = \frac{1}{t}\sum_{k=1}^{t}\left(X(k) - W_1(t)h_1(k)\right)^T\left(X(k) - W_1(t)h_1(k)\right) \tag{10}$$

$$\hat{W}_1(t) = \arg\left(\inf_{W_1(t)\in R^n} J_1(W_1(t))\right),\ \hat{W}_1(t) = \frac{\overline{X}^T(t)h^T{}_1(t)}{\left\|\overline{h}_1^T(t)\overline{h}_1(t)\right\|^2} \tag{11}$$

Denoting by $P_1(t) = \left(\sum_{k=1}^{t} h_1^2(k)\right)^{-1}$ and $k_1(t) = h_1(t)P_1(t)$, we get the RLS algorithm ,

$W_1(0)$ randomly selected; $h_1(t) = W^T_1(t\text{-}1)X(t)$; $\quad k_1(t) = \dfrac{P_1(t-1)h_1(t)}{1 + h_1^2(t)P_1(t-1)}$

$$P_1(t)=[1-k_1(t)h_1(t)]P_1(t-1);\quad \hat{W}_1(t)=\hat{W}_1(t-1)+k_1(t)\left[X(t)-h_1(t)\hat{W}_1(t-1)\right]$$

In case that the largest eigen value λ_1 of the covariance matrix Σ is of multiplicity order 1 and let ϕ_1 be its corresponding unit eigen vector. The theoretical analysis concerning the behavior of the sequence $\left(\hat{W}_1(t)\right)_{t\in N}$ establishes that, almost surely,

If $(W_1(0))^T\phi_1>0$, then $\lim_{t\to\infty}\hat{W}_1(t)=\phi_1$. If $(W_1(0))^T\phi_1<0$, then $\lim_{t\to\infty}\hat{W}_1(t)=-\phi_1$.

Let $X(t)=\sum_{i=1}^{n}\alpha_i(t)\phi_i$ be the expansion of the input signal in terms of the $\{\phi_1,...,\phi_n\}$, an orthogonal basis of Σ eigen vectors, where the corresponding eigen values are taken in the decreasing order. Let $d_p(t)=X(t)-\sum_{i=1}^{p-1}\alpha_i(t)\phi_i$ be the deflated signal at the level *p*, $2\le p\le n$. The extended RLS algorithm for learning the first *m* principal components is given by the following learning equations.

$$h_p(t)=W^T(t-1)X(t);\quad k_p(t)=\frac{P_p(t-1)h_p(t)}{1+h_p^2(t)P_p(t-1)}$$

$$P_p(t)=[1-k_p(t)h_p(t)]P_p(t-1);\ \hat{W}_p(t)=\hat{W}_p(t-1)+k_p(t)\left[X(t)-h_p(t)\hat{W}_p(t-1)\right]$$

Theoretical analysis establishes that, if $\lambda_1>\lambda_2>\cdots>\lambda_p>\lambda_{p+1}\ge\cdots\ge\lambda_n$, then for each $p\ge1$, the sequence $\left(\hat{W}_p(t)\right)_{t\in N}$ generated by the extended RLS converges to either ϕ_p or $-\phi_p$.

4 Experimental Analysis and Concluding Remarks

In the following we present the use of above mentioned learning schemes for image compression/decompression purposes. Let $I(t)$ be a wide-sense- stationary *N*-dimensional process of mean $E(I(t))$ resulted by sampling a given image I; $\bar{I}(t)=I(t)-E(I(t))$. Each sampled matrix $\bar{I}(t)$ is processed row by row, each row being split in lists of 15 consecutive components. We denote by $X(t)$ such a sub-list and we assume that $X(t)\sim N(0,\Sigma)$. We denote by $n=15$ the dimension of the input data, $m=3$ the number of desired principal components, $t_{max}\in\{10,20,50,75\}$ the number of the variants of the image I, $\eta(t)=\frac{1}{t\ln(t)}$ the sequence of learning rates taken to satisfy the constrains considered in the Kushner theorem, $W_0\in M_{15x3}(R)$ the initial synaptic memories whose entries are randomly generated; each column vector of W_0 is of norm 1. In case of the APEX algorithm, the initial values of the lateral connection weights are $\forall i\ge j, a_{ij}=0$ and for all $1\le i<j\le3$, a_{ij} are randomly generated according to the uniform distribution on $[0,1)$. The reported results are

obtained with respect to the following examples, $\sum_i = diag\left(\sigma_1^{(i)},...,\sigma_{15}^{(i)}\right),\ \ i=1,2,3,$ $\sigma_k^{(1)}=0.1,\ \ \sigma_2^{(2)}=6,\ \ \sigma_1^{(1)}=15, \sigma_2^{(1)}=\sigma_1^{(2)}=10, \sigma_3^{(1)}=5,\ \ \ \sigma_3^{(2)}=2, \sigma_k^{(2)}=0.04$ $\sigma_1^{(3)}=4, \sigma_2^{(3)}=\sigma_3^{(3)}=1,\ \ \sigma_k^{(3)}=0.01,\ \ 4\le k\le 15$. The empirical mean variation of the synaptic vectors on the final iteration and the mean error with respect to the eigen vectors are given by,

$$V=\frac{1}{m}\sum_{i=1}^{m} D\left(W_i\left(t_{max}\right),W_i\left(t_{max}-1\right)\right),$$

$$D\left(W_i\left(t_{\max}\right),W_i\left(t_{\max}-1\right)\right)=\sum_{k=1}^{n}\left|W_i\left(t_{\max}\right)(k)-W_i\left(t_{\max}-1\right)(k)\right|$$

$$Er=\frac{1}{m}\sum_{i=1}^{m} E\left(W_i\left(t_{max}\right),\Phi_i\right),\ E\left(W_i\left(t_{max}\right),\Phi_i\right)=\frac{1}{n}D\sum_{k=1}^{n}\left|W_i\left(t_{max}\right)(k)-\Phi_i(k)\right|.$$

The obtained results are shown in Table 1and Table 2.

Table 1.

V - GHA	Er- GHA	V - Sanger	Er- Sanger	V- APEX	Er- APEX	$\sum$	t_{max}
0.0377	0.0339	0.0371	0.0423	0.0491	0.0499	$\sum_1$	75
0.0186	0.0295	0.0186	0.0403	0.0243	0.0426	$\sum_2$	
0.0064	0.0379	0.0054	0.0532	0.0074	0.0417	$\sum_3$	
0.0387	0.0334	0.0393	0.0429	0.0542	0.0485	$\sum_1$	50
0.0276	0.0331	0.0273	0.0450	0.0348	0.0453	$\sum_2$	
0.0090	0.0414	0.0070	0.0561	0.0102	0.0442	$\sum_3$	
0.0896	0.0417	0.0851	0.0551	0.1085	0.0543	$\sum_1$	20
0.0572	0.0434	0.0499	0.0572	0.0662	0.0505	$\sum_2$	
0.0160	0.0484	0.0114	0.0614	0.0172	0.0493	$\sum_3$	
0.1569	0.0555	0.1751	0.0612	0.1759	0.0626	$\sum_1$	10
0.0774	0.0629	0.0600	0.0629	0.0858	0.0531	$\sum_2$	
0.0202	0.0637	0.0135	0.0637	0.0211	0.0520	$\sum_3$	

The entries of $W_0 \in M_{15x3}(R)$ are randomly generated, but each column of W_0 is of norm 1. The ratio $\frac{V}{Er}$ of the stabilization coefficient *V* and the error *Er*, is fast decreasing in case of the APEX and GHA algorithms as compared to its variation in case of Sanger variant. The APEX and GHA lead to smaller errors versus the stabilization index V. The stabilization of the Sanger variant is installed faster than in case of GHA and APEX. The errors are significantly influenced by the variation of the eigen values and they are less influenced by their actual magnitude.

According to the results obtained by our tests we conclude that there are no significant differences from the point of view of the corresponding convergence rates between the GHA and the Sanger variant, but the APEX algorithm proves to be slower than them, most probably because it the convergence rate is more influenced by the initial values. Also, the performance is strongly dependent on the magnitude of the noise variances. The tests on the efficiency of the RLS algorithm were performed on the 10×10 matrix representations of the Latin letters. The experiments pointed out that the good quality can be maintained when the compression/decompression process involved at least the first 15 components. Only 5 line features assure enough accuracy in the compression/decompression process.

Table 2.

$\frac{V}{Er}$ - GHA	$\frac{V}{Er}$ - Sanger	$\frac{V}{Er}$ - APEX	Σ	t_{max}
1.1120	0.8770	0.9839	Σ_1	75
1.1586	0.9160	1.2634	Σ_2	
2.1486	1.5446	1.9981	Σ_3	
2.8270	2.1320	2.8099	Σ_1	50
0.6305	0.4615	0.6029	Σ_2	
0.8338	0.6066	0.7682	Σ_3	
1.3179	0.8723	1.3108	Σ_1	20
1.5357	0.9538	1.6158	Σ_2	
0.1688	0.1015	0.1774	Σ_3	
0.2173	0.1247	0.2307	Σ_1	10
0.3305	0.1856	0.3488	Σ_2	
0.3899	0.2119	0.4057	Σ_3	

References

1. Chatterjee, C., Roychowdhury, V.P., Chong, E.K.P.: On Relative Convergence Properties of PCA Algorithms, IEEE Trans. on Neural Networks, vol.9,no.2 (1998)
2. Diamantaras, K.I., Kung, S.Y.: Principal Component Neural Networks: theory and applications, John Wiley &Sons, (1996)
3. Haykin, S., Neural Networks A Comprehensive Foundation, Prentice Hall,Inc. (1999)
4. Hastie, T., Tibshirani, R., Friedman,J.: The Elements of Statistical Learning Data Mining, Inference, and Prediction, Springer (2001)
5. Kushner, H.J., Clark, D.S.: Stochastic Approximation Methods for Constrained and Unconstrained Systems, Springer Verlag (1978)
6. J. Karhunen, E. Oja: New Methods for Stochastic Approximations of Truncated Karhunen-Loeve Expansions, Proc. 6th Intl. Conf. on Pattern Recognition, Springer Verlag (1982)
7. Sanger, T.D.: An Optimality Principle for Unsupervised Learning, Advances in Neural Information Systems, ed. D.S. Touretzky, Morgan Kaufmann (1989)