Professor Catalina COCIANU, PhD
The Bucharest Academy of Economic Studies
Professor Luminita STATE, PhD
University of Pitesti
Lecturer Panayiotis VLAMOS PhD
Ionian University, Corfu, Greece

NEURAL IMPLEMENTATION OF A CLASS OF PCA LEARNING ALGORITHMS

Abstract. *Principal component analysis allows the identification of a linear transformation such that the axes of the resulted coordinate system correspond to the largest variability of the investigated signal. The advantages of using principal components reside from the fact that bands are uncorrelated and no information contained in one band can be predicted by the knowledge of the other bands, therefore the information contained by each band is maximum for the whole set of bits. The paper reports a series of conclusions concerning the performance and efficiency of some of the most frequently used PCA algorithms implemented on neural architectures.*

Keywords*: image processing, principal component analysis, neural network, learning algorithms, image compression/decompression.*

JEL Classification : C45, C46, C63, CO2.

1 INTRODUCTION

Feature selection referrers to a process whereby a data space is transformed into a feature space that, in theory, has precisely the same dimension as the original data space. However, the transformation is designed in such a way that a data set may be represented by a reduced number of effective features and yet retains most of the intrinsic information content of the data, that is the data set undergoes a dimensionality reduction. In other words, feature extraction is expected to allow the avoiding of the "curse of dimensionality", to improve the generalization ability of classifiers and also to reduce the computational requirements of pattern classification.

Principal Component Analysis, also called Karhunen-Loeve transform is a well-known statistical method for feature extraction, data compression and multivariate data projection and so far it has been broadly used in a large series of signal and image processing, pattern recognition and data analysis applications.

Principal component analysis allows the identification of a linear transformation such that the axes of the resulted coordinate system correspond to the largest variability of the investigated signal. The signal features corresponding to the new coordinate system are uncorrelated, that is, in case of normal models these components are independent. In case of pictorial signals, this means that, if we apply this transform to a grey scale image by using the values of the first coordinate of each pixel, it will contain totally uncorrelated information from the information that will be contained in the grey scale image formed by the second coordinate of each pixel and the information contained in the image formed by the third coordinate of each pixel.

The advantages of using principal components reside from the fact that bands are uncorrelated and no information contained in one band can be predicted by the knowledge of the other bands, therefore the information contained by each band is maximum for the whole set of bits (Diamantaras, 1996).

The grey values in the bands created from principal component analysis have no physical meaning, as they do not correspond to any physical colors. As a result, the grey value of a pixel cannot be used directly for the classification of a pixel. This is particularly relevant to remote sensing applications, where pixels are classified according to their grey values. In a principal component band, pixels that represent water, for example, may appear darker or brighter then other pixels in the image depending on the image content, while the degree of grayness of water pixels in the various spectral bands is always consistent, well understood by remote sensing scientists, and often used to identify them.

Self-organization is one of the most important learning paradigms of neural systems. The purpose of an algorithm for self-organizing learning is to discover significant patterns or features in the input data without the help provided by an external teacher. The ability to adapt to the environment without the provision of an external teacher is encountered in nature in most intelligent organisms. In this paradigm, the lack of teaching signals is compensated for by an inner purpose, i.e., some built-in criterion or objective function that the system seeks to optimize.

A large number of specialized neural networks and learning algorithms have been proposed to perform principal component analysis (PCA) tasks. One of the most frequently used method in the study of the convergence properties corresponding to different stochastic learning PCA algorithms, is derived from (Kushner and Clark, 1978) developments and 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. The Generalized Hebbian Algorithm (GHA) extends the Oja's learning rule for learning the first principal components, the extension being essentially based on the Hotteling deflation technique.

In the following we restrict the development to the problem of feature extracting unsupervised neural networks derived on the base of the biologically motivated Hebbian self-organizing principle which is conjectured to govern the natural neural assemblies and the classical principal component analysis (PCA) method used by statisticians for almost a century for multivariate data analysis and feature

extraction.

Both ends of the connection have appealing properties resulting from residing from the simplicity and locality of the Hebbian type learning and the optimality of the PCA method in dimensionality reduction. Classical PCA is based on the second-order statistics of the data and, in particular, on the eigen-structure of the data covariance matrix and accordingly, the PCA neural models incorporate only cells with linear activation functions. More recently, several generalizations of the classical PCA models to non-Gaussian models, the Independent Component Analysis (ICA) and the Blind Source Separation techniques (BSS) have become a very attractive and promising framework in developing more efficient image restoration algorithms.

The paper reports a series of conclusions concerning the performance and efficiency of some of the most frequently used PCA algorithms implemented on neural architectures and their corresponding efficiency for specific image processing tasks.

2 HEBBIAN LEARNING IN FEEDFORWARD 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 *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 $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$. From the point of view of the classical LMS criterion, the most informative directions of the process $(X(t), t \geq 0)$ are given by $\Phi_1,...,\Phi_n$ and accordingly, 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 a 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 us denote by $W(t) = (W_1(t),...,W_m(t))$ 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 where $Y_j(t) = W_j^T(t)X(t), 1 \leq j \leq m$.

The Hebbian learning rule for learning the first principal component is:

$$W_1(k+1) = W_1(k) + \eta(k)X(k)Y_1(k) \qquad (1)$$

where the sequence of learning rates $(\eta(k))$ are taken such that the conditions of the Kushner theorem hold, $\sum_{k=1}^{\infty} \eta(k) = \infty$, $\lim_{k\to\infty} \eta(k) = 0$, there exists $p > 1$ such that $\sum_{k=1}^{\infty} (\eta(k))^p < \infty$.

The Equation (1) is not interesting because the resulting algorithm is unstable that is $\lim_{k\to\infty} \|W_1(k)\| = \infty$ but asymptotically $W_1(k)$ is directed toward $L(\Phi_1)$ the linear

subspace generated by Φ_1, hence the instability is the only obstacle inhibiting Equation (1) from being a real principal component analyzer. Consequently, the normalized version of the Hebbian learning rule:

$$W_1(k+1)=\frac{W_1(k)+\eta(k)X(k)Y_1(k)}{\|W_1(k)+\eta(k)X(k)Y_1(k)\|} \qquad (2)$$

is a possible learning scheme for the first principal component but it is not suitable enough because it is not a local rule.

In order to get a local learning scheme a linearized version of (2) using first order approximation was proposed in (Karhunen, Oja, 1982), yielding to the celebrated Oja's learning algorithm:

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

The Generalized Hebbian Algorithm (GHA) (Haykin, 1999) is one of the first neural models for extracting multiple PCs. The idea of GHA is to use the Hotelling deflation technique and the Oja's algorithm for learning the required number of principal components.

Each neuron *j* is influenced by all neurons *i*, *i*<*j* and its input is the deflated signal at the level of the *j*th principal component. 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 two outputs,

$$Y_j(t)=W_j^T(t)X(t)$$
$$\tilde{Y}_j(t)=W_j^T(t)\tilde{X}_j(t)$$

where $\tilde{X}_j(t)=\tilde{X}_{j-1}(t)-Y_{j-1}(t)W_{j-1}(t)$, $j \geq 2$, is the deflated signal at the level of the *j*th principal component, $\tilde{X}_1(t)=X(t)$.

The GHA learning scheme is,

$$W_1(k+1)=W_1(k)+\eta(k)\left(X(k)Y_1(k)-Y_1^2(k)W_1(k)\right) \qquad (4)$$
$$W_j(k+1)=W_j(k)+\eta(k)\left(\tilde{X}_j(k)\tilde{Y}_j(k)-\tilde{Y}_j^2(k)W_j(k)\right) \qquad (5)$$

for $2 \leq j \leq m$,

where:

$$Y_j(k)=W_j^T(k)X(k),$$

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

The variant proposed by Sanger (Sanger, 1989) 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,

$$W_1(k+1)=W_1(k)+\eta(k)\left(X(k)Y_1(k)-Y_1^2(k)W_1(k)\right) \qquad (6)$$

$$W_j(k+1) = W_j(k) + \eta(k)\left(\tilde{X}_j(k)Y_j(k) - Y_j^{\,2}(k)W_j(k)\right) \tag{7}$$

for $2 \le j \le m$,

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.

Obviously, the synaptic learning algorithm can be also written as,

$$W_j(k+1) = W_j(k) + \eta(k)\left(X(k)Y_j(k) - Y_j(k)\sum_{i=1}^{j} Y_i(k)W_i(k)\right)$$

$$1 \le j \le m \tag{8}$$

The ODE assigned by the Kushner theorem is,

$$\frac{dW_1(t)}{dt} = SW_1(t) - \left(W_1^T(t)SW_1(t)\right)W_1(t), \tag{9}$$

$$\frac{dW_j(t)}{dt} = SW_j(t) - \left(W_j^T(t)SW_j(t)\right)W_j(t) - \sum_{i=1}^{j-1}\left(W_i^T(t)SW_j(t)\right)W_i(t),$$

$$2 \le j \le m. \tag{10}$$

The APEX learning algorithm proposed in (Diamantaras, Kung, 1996) generalizes the idea of lateral influences by imposing a certain learning process to the weights of lateral connections. The output of each neuron *j*, $j \ge 2$, 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 $a_{ij}(t)$,

$$Y_j(t) = W_j^T(t)X(t) - \sum_{i=1}^{j-1} a_{ij}(t)Y_i(t) \tag{11}$$

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

$$W_j(t+1) = W_j(t) + \eta(t)\left[Y_j(t)X(t) - Y_j^2(t)W_j(t)\right] \tag{12}$$

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

$$a_{ij}(t+1) = a_{ij}(t) + \eta(t)\left[Y_i(t)Y_j(t) - Y_j^2(t)a_{ij}(t)\right] \tag{13}$$

The ODE obtained according to the Kushner construction is,

$$\frac{dW_1(t)}{dt} = SW_1(t) - \left(W_1^{\,T}(t)SW_1(t)\right)W_1(t) \tag{14}$$

$$\frac{dW_j(t)}{dt} = SW_j(t) - \sum_{i=1}^{j-1}\lambda_i\phi_i a_{ij}(t) - \sigma_j(t)W_j(t) \tag{15}$$

for $j \ge 2$, where:

$$\begin{cases} \sigma_j(t) = q_j^T(t) S q_j(t) \\ q_j(t) = W_j(t) - \Phi a_j(t) \\ a_j(t) = (a_{1j}(t),..., a_{j-1,j}(t), 0,...,0) \end{cases}$$

and

$\Phi = (\phi_1,...,\phi_m)$ is the matrix whose columns are unit eigen vectors corresponding to the largest *m* eigen values $\lambda_1,...,\lambda_m$ of *S*.

Note that the theoretical analysis (Diamantaras, Kung, 1996), (Haykin, 1999), (Chatterjee, Roychowdhury, 1998) establishes the almost sure convergence to the principal components of the sequences of weight vectors generated by the above mentioned algorithms. The convergence is assured in case the *m* largest eigen values of *S* are such that $\lambda_1 > \lambda_2 > \cdots > \lambda_m > \lambda_{m+1} \geq \cdots \geq \lambda_n$.

3 RECURSIVE LEAST SQUARE LEARNING ALGORITHM OF THE PRINCIPAL DIRECTIONS

The adaptive extraction of the first principal component can be performed using the recursive least square (RLS) algorithm on a neural architecture consisting of an unique hidden processing unit. The RLS algorithm is briefly presented in the following. 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,…. The neural architecture is depicted in Figure 1.

Each of the input and respectively output layers F_X, F_Y consists of *n* processing units and the hidden layer F_H contains one neuron. 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. In other words, at each moment *t*, the compression of the input signal is performed by the linear filter $(W_1(t\text{-}1))^T$ and the decompression is performed also linearly using the filter $W_1(t\text{-}1)$. Consequently, the mean error at the moment *t* is $J_1(t) = \sum_{k=1}^{t} \varepsilon^2(k)$, where $\varepsilon^2(k) = \|X(k) - Y(k)\|^2$.

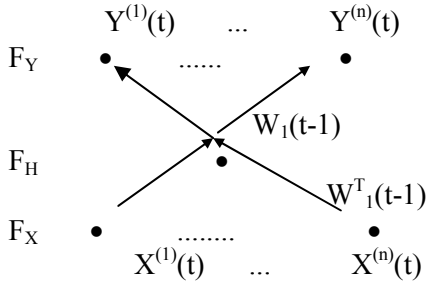

Figure 1: The neural architecture of RLS

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,

$$J_1(W_1(t)) = = \frac{1}{t}\sum_{k=1}^{t}(X(k)-W_1(t)h_1(k))^T(X(k)-W_1(t)h_1(k)) \tag{16}$$

and

$$\hat{W}_1(t) = \arg\left(\inf_{W_1(t)\in R^n} J_1(W_1(t))\right) \tag{17}$$

Using straightforward computation, we get

$$\hat{W}_1(t) = \frac{\overline{X}^T(t)h^T{}_1(t)}{\left\|\overline{h}_1^T(t)\overline{h}_1(t)\right\|^2} \tag{18}$$

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 adaptive version of the RLS learning algorithm ,

$$\begin{cases} W_1(0) \text{ randomly selected} \\ h_1(t) = W^T{}_1(t\text{-}1)X(t) \\ k_1(t) = \dfrac{P_1(t-1)h_1(t)}{1+h_1^2(t)P_1(t-1)} \\ P_1(t)=[1\text{-}k_1(t)h_1(t)]P_1(t\text{-}1) \\ \hat{W}_1(t) = \hat{W}_1(t-1) + k_1(t)\left[X(t) - h_1(t)\hat{W}_1(t-1)\right] \end{cases}$$

We assume that the first component corresponding to the input is unambiguous, that is 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. Note that if $W_1(0) \notin L^{\perp}(\phi_1)$ then, $W_1(t) \notin L^{\perp}(\phi_1)$ a.s. for any t, where $L(\phi_1) = span\{\phi_1\}$.

The theoretical analysis concerning the behavior of the sequence $\left(\hat{W}_1(t)\right)_{t\in N}$ generated by the RLS algorithm establishes that:

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 $\lim_{t\to\infty}\hat{W}_1(t) = -\phi_1$.

Note that the RLS is a stable learning scheme almost surely for the first principal component of the input distribution.

The extension of the RLS for learning the first m principal components can be obtained using the Hotelling deflation technique.

Let $X(t) = \sum_{i=1}^{n}\alpha_i(t)\phi_i$ be the expansion of the input signal in terms of the $\{\phi_1,...,\phi_n\}$, the orthogonal basis of the Σ eigen vectors, where the corresponding eigen values are sorted 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, $2 \le p \le n$. Then the spectral decomposition of the covariance matrix Σ_p corresponding to $d_p(t)$ is $\sum_{i=p}^{n} \lambda_i \phi_i \phi_i^T$, therefore, λ_p is the largest eigen value of Σ_p and ϕ_p is its first component.

Taking into account these arguments, we arrive to a sequential process of learning any number of principal components.

The extended RLS algorithm is given by the following learning equations.

$$\begin{cases} h_p(t) = W^T(t-1)X(t) \\ k_p(t) = \dfrac{P_p(t-1)h_p(t)}{1 + h_p^2(t)P_p(t-1)} \\ P_p(t) = [1 - k_p(t)h_p(t)]P_p(t-1) \\ \hat{W}_p(t) = \hat{W}_p(t-1) + k_p(t)\left[X(t) - h_p(t)\hat{W}_p(t-1)\right] \end{cases}$$

The implementation of the extended RLS algorithm can be performed on the simple feed forward architecture depicted in Figure 2.

Theoretical analysis establishes that, if $\lambda_1 > \lambda_2 > \cdots > \lambda_p > \lambda_{p+1} \ge \cdots \ge \lambda_n$, then the sequence $\left(\hat{W}_p(t)\right)_{t \in N}$ generated by the extended RLS fulfils almost surly,

If $\left(W_p(0)\right)^T \phi_p > 0$, then $\lim_{t \to \infty} \hat{W}_p(t) = \phi_p$

If $\left(W_p(0)\right)^T \phi_p < 0$, then $\lim_{t \to \infty} \hat{W}_p(t) = -\phi_p$

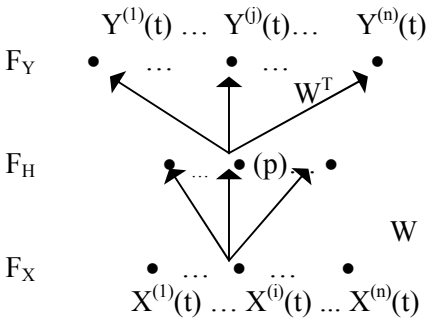

Figure 2: The neural architecture of extended RLS

4 EXPERIMANTAL ANALYSIS AND CONCLUSIVE 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; we denote by $\bar{I}(t) = I(t) - E(I(t))$. Each sampled matrix $\bar{I}(t)$ is processed row by row, each row being split in lists of

15 consecutive components. In the following, we denote by *n* the dimension of the input data. In this case, n=15. Let $X(t)$ be a sub-list. We assume that $X(t) \sim N(0, \Sigma)$. We denote by m the number of desired principal components and by $t_{\max}$ the number of the distorted variants of the image *I*. The sequence of learning rates is $\eta(t) = \frac{1}{t \ln(t)}$ and it is taken such that the constrains considered in the Kushner theorem are satisfied. Let W_0 be 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 $\forall i \geq j, a_{ij} = 0$ and for all $1 \leq i < j \leq 3$, a_{ij} are randomly generated according to the uniform distribution on $[0,1)$.

The experiments were performed for several covariance matrices, the reported results being obtained with respect to the following examples,

m=3, $t_{\max} = 10,20,50,75$

$\Sigma_i = diag\left(\sigma_1^{(i)},...,\sigma_{15}^{(i)}\right), \quad i = 1,2,3$

where:

$\sigma_1^{(1)} = 15, \sigma_2^{(1)} = 10, \sigma_3^{(1)} = 5, \sigma_k^{(1)} = 0.1$

$\sigma_1^{(2)} = 10, \sigma_2^{(2)} = 6, \sigma_3^{(2)} = 2, \sigma_k^{(2)} = 0.04$

$\sigma_1^{(3)} = 4, \sigma_2^{(3)} = 1, \sigma_3^{(3)} = 1, \sigma_k^{(3)} = 0.01$

$4 \leq k \leq 15$

In order to compare the convergence rates of the learning algorithms of Hebbian type, for each sample, we computed,

- the empirical mean variation of the synaptic vectors on the final iteration,

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

$$D\left(W_i(t_{\max}), W_i(t_{\max}-1)\right) =$$

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

- the mean error with respect to the eigen vectors $Er = \frac{1}{m}\sum_{i=1}^{m} E\left(W_i(t_{max}), \Phi_i\right)$,

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

The obtained results are reported in (Rosca, State, Cocianu, 2008)

In case the entries of $W_0 \in M_{15x3}(R)$ are randomly generated, but each column of W_0 is of norm 1, the following conclusions can be derived. 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 the Sanger variant.

Also, in case of the APEX and GHA algorithms the values of the ration $\frac{V}{Er}$ remain larger then in case of Sanger algorithm, that is the APEX and GHA lead to smaller errors versus the stabilization index V. The stabilization of the Sanger variant is installed faster then in case of GHA and APEX. In other words, approximates of the principal components are learned faster by the Sanger algorithm as compared to the GHA and APEX, but the performance expressed in terms of the error is lower. The errors are significantly influenced by the variation of the eigen values and they are less influenced by their actual magnitude.

In case we use randomly generated vectors as initial synaptic memories, the tests showed that good results can be achieved when the entries are non zero small values.

The results reported in the following were obtained for the settings,

- the entries of the initial synaptic memories $W_0 \in M_{15x3}(R)$ uniformly generated in $[0.05,0.15) \cup (-0.15,-0.05]$;
- $n = 15, m = 2$, $t_{max} \in \{75,150,200\}$;
- the data generated according to the covariance matrices $\Sigma_i = diag\left(\sigma_1^{(i)},...,\sigma_{15}^{(i)}\right), \quad i = 1,2,3$

The obtained results are shown in table 1, table 2, and table 3.

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.

Table 1

V - GHA	V - Sanger	V- APEX	Σ	t_{max}
0.0161	0.0162	0.0189	Σ_1	200
0.0087	0.0087	0.0097	Σ_2	
0.0025	0.0029	0.0030	Σ_3	
0.0227	0.0225	0.0264	Σ_1	150
0.0120	0.0121	0.0134	Σ_2	
0.0036	0.0041	0.0041	Σ_3	
0.0495	0.0491	0.0553	Σ_1	75
0.0241	0.0264	0.0279	Σ_2	
0.0059	0.0064	0.0065	Σ_3	

Table 2

Er - GHA	Er - Sanger	Er- APEX	Σ	t_{max}
0.0342	0.0343	0.0494	Σ_1	200
0.0452	0.0465	0.0675	Σ_2	
0.1009	0.1011	0.1037	Σ_3	
0.0395	0.0388	0.0567	Σ_1	150
0.0519	0.0529	0.0721	Σ_2	
0.1034	0.1035	0.1052	Σ_3	
0.0589	0.0587	0.0758	Σ_1	75
0.0722	0.0715	0.0841	Σ_2	
0.1078	0.1080	0.1083	Σ_3	

Table 3

$\frac{V}{Er}$ - GHA	$\frac{V}{Er}$ - Sanger	$\frac{V}{Er}$ - APEX	Σ	t_{max}
0.4707	0.4723	0.3825	Σ_1	200
0.1924	0.1870	0.1437	Σ_2	
0.0247	0.0286	0.0289	Σ_3	
0.5746	0.5798	0.4656	Σ_1	150
0.2312	0.2287	0.1858	Σ_2	
0.0348	0.0396	0.0389	Σ_3	
0.8404	0.8364	0.7295	Σ_1	75
0.3337	0.3692	0.3317	Σ_2	
0.0547	0.0592	0.0600	Σ_3	

The tests on the efficiency of the RLS algorithm were performed on the 10×10 matrix representations of the Latin letters. For each letter we considered 11

samples and the Hamming distance was taken as a criterion in evaluating the quality of the compression/decompression process. The experiments pointed out that the good quality can be maintained when the compression/decompression process involved at least the first 15 components. The error measured in terms of the Hamming distance stabilizes for $m \geq 5$ at the values 3, in case of Karhunen-Loeve compression and 0.5 in case of RLS compression respectively. Consequently only 5 line features assure enough accuracy in the compression/decompression process.

Several tests were also performed on gray level images and the conclusions can be summarized as follows. Good quality can be assured when higher compression rates are considered; in our tests, only about 7% of the signal characteristics were needed for decompression purposes. Relative low restoration errors when at least 7% of the first components were involved in the compression/decompression process. Relative noise robustness.

REFERENCES

[1]**Chatterjee, C., Roychowdhury, V.P., Chong, E.K.P.(1998),** ***On Relative Convergence Properties of Principal Component Analysis Algorithms*****,** IEEE Transaction on Neural Networks, vol.9,no.2;

[2]**Chellappa, R. , Jinchi, H. (1985),** ***A Nonrecursive Filter for Edge Preserving Image Restoration***, Proceedings, International Conference on Acoustic, Speech and Signal Processing, Tampa;

[3]**Chellappa, R., Kashyap,R.L. (1982),** ***Digital Image Restoration Using Spatial Interaction Models***. In Proc. Intl.Conf.on Acoustic, Speech and Signal Processing, vol. ASSP-30;

[4]**Cocianu, C., State, L., Stefanescu, V., Vlamos, P. (2006),** ***PCA-Based Data Mining Probabilistic and Fuzzy Approaches with Applications in Pattern Recognition***, Proceedings of ICSOFT 2006, Portugal, pp. 55-60;

[5]**Cocianu, C., State, L., Stefanescu, V., Vlamos, P.(2004),** ***On the Efficiency of a Certain Class of Noise Removal Algorithms in Solving Image Processing Tasks*****.** In: Proceedings of the ICINCO, Setubal, Portugal;

[6]**Cocianu, C., State, L., Rosca, I., Vlamos, P. (2007),** ***A New Adaptive Classification Scheme Based on Skeleton Information,*** Proceedings of SIGMAP;

[7]**Diamantaras, K.I., Kung, S.Y. (1996),** ***Principal Component Neural Networks: Theory and Applications***, John Wiley &Sons;

[8]**Elad, M., Aharon, M. (2004),** ***Image Denoising via Sparse and Redundant Representations over Learned Dictionaries***, IEEE transactions on image processing, vol. 15, n°12, pp. 3736-3745;

[9]**Gonzales, R., Woods, R. (2002),** ***Digital Image Processing***, Prentice Hall;

[10]**Haykin, S. (1999),** ***Neural Networks A Comprehensive Foundation***, Prentice Hall,Inc. ;

[11]**Hastie, T., Tibshirani, R., Friedman,J. (2001),** ***The Elements of Statistical Learning Data Mining,*** Inference, and Prediction, Springer;

[12]**Hyvarinen, A., Karhunen, J., Oja,E. (2001),** ***Independent Component Analysis,*** John Wiley &Sons;

[13]**Hyvarinen, A., Hoyer, P., Oja, P.(1999),** ***Image Denoising by Sparse Code Shrinkage***, www.cis.hut.fi/projects/ica,
[14]**Jain, A. K., Kasturi, R., Schnuck, B. G. (1995),** ***Machine Vision,*** McGraw Hill;
[15]**Kushner, H.J., Clark, D.S. (1978),** ***Stochastic Approximation Methods for Constrained and Unconstrained Systems,*** Springer Verlag;
[16] **Karhunen, J., Oja, E. (1982),** ***New Methods for Stochastic Approximations of Truncated Karhunen-Loeve Expansions***, Proceedings of the 6th International Conference on Pattern Recognition, Springer Verlag;
[17]**Larose, D.T. (2006),** ***Data Mining. Methods and Models*****,** Wiley-Interscience, John Wiley and Sons, Inc Publication, Hoboken, New Jersey;
[18]**Liu, J., and Chen, S.(2006.),** ***Discriminant Common Vectors versus Neighborhood Components Analysis and Laplacianfaces: A Comparative Study in Small Sample Size Problem***. Image and Vision Computing 24 ;
[19]**Pitas, I. (1993),** ***Digital Image Processing Algorithms***, Prentice Hall;
[20]**Ruxanda Gh., (2009),** ***Supervised Pattern Recognition with Potential Functions Method*****,** Economic Computation and Economic Cybernetics Studies and Research, issue 2, ASE Publishing House, Bucharest;
[21]**Sanger, T.D.(1989),** ***An Optimality Principle for Unsupervised Learning, Advances in Neural Information Systems*****;** Ed. D.S. Touretzky, Morgan Kaufmann;
[22]**Sonka, M., Hlavac, V.(1997),** ***Image Processing, Analyses and Machine Vision,*** Chapman & Hall Computing;
[23]**Stark, J.L., Murtagh, F., Bijaoui, A.(1995),** ***Multiresolution Support Applied to Image Filtering and Restoration,*** Technical Report;
[24]**State, L, Cocianu, C, Vlamos, P. (2001),** ***Attempts in Using Statistical Tools for Image Restoration Purposes,*** In Proceedings of SCI2001, Orlando, USA, July 22-25;
[25]**State, L., Cocianu, C., Rosca, I., Vlamos, P.(2008),** ***A New Learning Algorithm for Classification in the Reduced Space,*** Proceedings of ICEIS;
[26]**Roşca, I., State, L., Cocianu, C. (2008),** ***Learning Schemes in Using PCA Neural Networks for Image Restoration Purposes***, WSEAS Transactions on Information Science and Applications, Vol. 5, July;
[27]**State, L., Cocianu, C., Vlamos, P., Stefanescu, V. (2008),** ***A New Unsupervised Learning Scheme to Classify Data of Relative Small Volume*****,** Economic Computation and Economic Cybernetics Studies and Research, issue 1-2, ASE Publishing House, Bucharest;
[28]**Umbaugh, S.(1998),** ***Computer Vision and Image Processing,*** Prentice Hall;
[29]**Woods, J.W., Ingle,V.K.(1981),** ***Kalman Filtering in Two Dimensions: Further Results,*** In IEEE Trans. Acoustics, Speech Processing, vol. ASSP-29.