Neural Network for Principal Component Analysis with Applications in Image Compression

Luminita STATE
Dept. of Computer Science, University of Pitesti, Pitesti, ROMANIA

Catalina Lucia COCIANU
Dept. of Computer Science, Academy of Economic Studies, Bucharest, ROMANIA

and

Vlamos PANAYIOTIS
Hellenic Open University, GREECE

ABSTRACT

Classical feature extraction and data projection methods have been extensively investigated in the pattern recognition and exploratory data analysis literature. Feature extraction and multivariate data projection allow avoiding the "curse of dimensionality", improve the generalization ability of classifiers and significantly reduce the computational requirements of pattern classifiers. During the past decade a large number of artificial neural networks and learning algorithms have been proposed for solving feature extraction problems, most of them being adaptive in nature and well-suited for many real environments where adaptive approach is required. 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.

Keywords: feature extraction, pattern recognition, PCA, RLS algorithm, Karhunen-Loeve transform, image processing, data compression/decompression

1. GENERAL VIEW ON A CERTAIN CLASS OF PCA LEARNING SCHEMES

Clasical feature extraction and data projection methods have been extensively studied in the pattern recognition and exploratory data analysis literature. Feature selection referes to a process whereby a data space is transformed into a feature space that, in theory, has exactely the same dimesion 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 retain 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.

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 [4] 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 the 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. Let us assume that the first m principal components have to be encoded as local memories (W_j) (synaptic weights) of a neural system, where $1 \le m \le n$ and n is the dimension of the input data. The computing layer consists of m units of local memories (W_j), each neuron being interconnected with all neurons of rank greater or equal then its rank.

The input is sampled from a n-dimensional stochastic process $X(t)=[X^{(1)}(t), \ldots, X^{(n)}(t)]$ stationary in the large sense, $E(X(t))=0$ .Each neuron j is influenced by all neurons i, $i<j$ and its input is the deflated signal at the level of the jth principal component. At any moment t, each neuron j, $j \ge 1$, receives two inputs, the original signal $X(t)$ and the deflated signal $X_j(t)$ and computes two outputs,

$$Y_j(t) = W_j^T(t) X(t) \quad (1)$$

$$\tilde{Y}_j(t) = W_j^T(t) \tilde{X}_j(t) \quad (2)$$

where $\tilde{X}_j(t) = \tilde{X}_{j-1}(t) - Y_{j-1}(t) W_{j-1}(t)$, $j \ge 2$, is the deflated signal at the level of the jth principal component, $\tilde{X}_1(t) = X(t)$. The updating of the local memories is given by,

$$W_1(t+1) = W_1(t) + \eta(t)\left[Y_1(t)X(t) - Y_1^2(t)W_1(t)\right] \quad (3)$$

$$W_j(t+1) = W_j(t) + \eta(t)\left[\tilde{Y}_j(t)\tilde{X}_j(t) - \tilde{Y}_j^2(t)W_j(t)\right] \quad (4)$$

for $j \ge 2$.

The variant of the GHA proposed by Sanger [9] simplifies the learning scheme by using only the output $Y_j(t)$ for both, updating the synaptic memories and signal deflation. According to the Sanger PCA learning algorithm, the updating is given by,

$$W_j(t+1) = W_j(t) + \eta(t)\left[Y_j(t)X(t) - Y_j(t)\sum_{i=1}^{j} Y_i(t)W_i(t)\right] \quad (5)$$

for $j \ge 1$.

The learning rates $\eta(t)$ are constrained to fulfill the regularity conditions given by Kushner theorem,

$$\begin{cases}\eta(t)>0\\ \sum_{t\geq 1}\eta(t)=\infty\\ \lim_{t\to\infty}\eta(t)=0\\ \sum_{t\geq 1}\eta^{p}(t)<\infty \text{ for some } p>1\end{cases}$$

According to the Kushner construction, the ODE describing the dynamics is,

$$\frac{dW_1(t)}{dt}=SW_1(t)-\left(W_1{}^T(t)SW_1(t)\right)W_1(t) \quad (6)$$

$$\frac{dW_j(t)}{dt}=SW_j(t)-\left(W_j{}^T(t)SW_j(t)\right)W_j(t)-$$
$$-\sum_{i=1}^{j-1}\left(W_i{}^T(t)SW_j(t)\right)W_i(t) \quad (7)$$

for $j\geq 2$, where $S=E\left(X(t)X^T(t)\right)$.

The APEX learning algorithm proposed by Kung and Diamantaras [3] generalizes the idea of lateral influences by imposing a certain learning process to the weights of lateral connections. The output of each neuron *j*, $j\geq 2$, is computed from its own output and the effects of the outputs corresponding to all neurons *i*, $1\leq i\leq j-1$, weighted by the coefficients $a_{ij}(t)$,

$$Y_j(t)=W_j^T(t)X(t)-\sum_{i=1}^{j-1}a_{ij}(t)Y_i(t) \quad (8)$$

performed using an unique hidden processing unit as follows. Let $X(t)=[X^{(1)}(t),\ ...,\ X^{(n)}(t)]$ be the *n*-dimensional input signal modeled as a stationary stochastic process of mean $m=0$ and covariance matrix S. We denote by $W_1(t\text{-}1)$ the synaptic vector at the moment *t* and assume that the inputs are applied at the moments $t=0,1,2,\ldots$. The neural architecture $F_X \xrightarrow{W_1^T} F_H \xrightarrow{W_1} F_Y$ 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 $\varepsilon^2(k)=\|X(k)-Y(k)\|^2$.

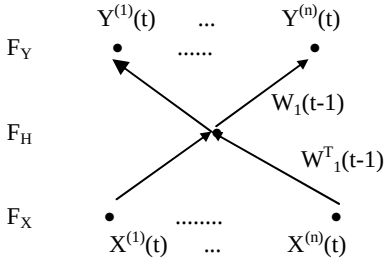

Figure 1

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

$$W_j(t+1)=W_j(t)+\eta(t)\left[Y_j(t)X(t)-Y_j^2(t)W_j(t)\right] \quad (9)$$

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

$$a_{ij}(t+1)=a_{ij}(t)+\eta(t)\left[Y_i(t)Y_j(t)-Y_j^2(t)a_{ij}(t)\right] \quad (10)$$

The ODE obtained according to the Kushner construction is,

$$\frac{dW_1(t)}{dt}=SW_1(t)-\left(W_1{}^T(t)SW_1(t)\right)W_1(t) \quad (11)$$

$$\frac{dW_j(t)}{dt}=SW_j(t)-\sum_{i=1}^{j-1}\lambda_i\phi_i a_{ij}(t)-\sigma_j(t)W_j(t) \quad (12)$$

for $j\geq 2$, where

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

and

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

2. EXTENDED RLS ALGORITHM FOR PRINCIPAL COMPONENT ANALYSIS

The adaptive extraction of the first principal component can be

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*, $1 £ k £ t$, the decompression is assumed as being performed using the filter $W_1(t)$, that is,

$$J_1(W_1(t))=$$
$$=\frac{1}{t}\sum_{k=1}^{t}\left(X(k)-W_1(t)h_1(k)\right)^T\left(X(k)-W_1(t)h_1(k)\right) \quad (13)$$

and

$$\hat{W}_1(t)=\arg\left(\inf_{W_1(t)\in R^n}J_1(W_1(t))\right). \quad (14)$$

Using straightforward computation, we get

$$.\hat{W}_1(t)=\frac{\overline{X}^T(t)h^T{}_1(t)}{\left\|\overline{h}_1^T(t)\overline{h}_1(t)\right\|^2} \quad (15)$$

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 (1),

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

We assume that the first component corresponding to the input is unambiguous, that is the largest eigen value l_1 of the covariance matrix *S* is of multiplicity order 1 and let f_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 $L(\phi_1) = span\{\phi_1\}$.

Theorem 1. Let $\left(\hat{W}_1(t)\right)_{t\in N}$ be the sequence generated by the stochastic algorithm (1).

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 (1) is a stable scheme for a.s. learning the first component of the input distribution.

The extension of the adaptive scheme (1) 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 (2) can be expressed as follows:

Input*: X*(*t*) stationary stochastic process of mean 0 and covariance matrix Σ

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

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

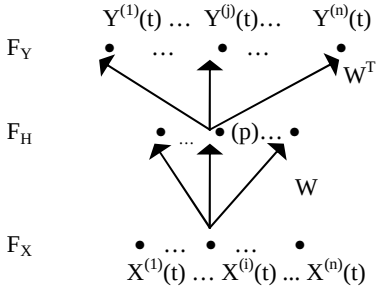

Figure 2.

Theorem 2. Let $\left(\hat{W}_p(t)\right)_{t\in N}$ be the sequence generated by the stochastic algorithm (2).

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

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

3. EXPERIMENTAL REPORT ON USING THE RLS ALGORITHM IN COMPRESSING BINARY IMAGES AND GRAY LEVEL IMAGES

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. Some of the results are depicted in Figure 6 and Figure 7 (see Appendix).

Several tests were also performed on gray level images and the conclusions can be summarised 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.

Some of the results are presented in the pictures 3, 4 and 5 (see Appendix) . For each example, there are supplied the input prototype and the decompressed image respectively when,

- The compression/decompression involved the first 3 principal components of 320 for 8 input samples (Figure 3);
- The compression/decompression involved the first 25 principal components of 320 for 35 input samples (Figure 4);
- The compression/decompression involved the first 2 principal components of 250 for 23 input samples (Figure 5).

4. REFERENCES

[1] S. Bannour, M.R. Azimi-Sadjadi, "Principal Component Extraction Using Recursive Least Squares Learning" **IEEE Transaction on Neural Networks**, vol.6,no.2, 1995

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

[3] K.I. Diamantaras, S.Y. Kung, **Principal Component Neural Networks: theory and applications** , John Wiley &Sons,1996

[4] H.J. Kushner., D.S. Clark, **Stochastic Approximation Methods for Constrained and Unconstrained Systems**, Springer Verlag, 1978

[5] S. Haykin, **Neural Networks A Comprehensive Foundation**, Prentice Hall,Inc. 1999

[6] J. Mao, A.K. Jain, "Artificial Neural Networks for Feature Extraction and Multivariate Data Projection", **IEEE Transaction on Neural Networks**, vol.6,no.2, 1995

[7] K. Matsuoka, M. Kawamoto, " A Neural Network that Self-Organizes to Perform Three Operations Related to Principal Component Analysis", **Neural Networks**, vol.7,no.5, 1994

[8] M.D. Plumbley, “Lyapunov Functions for Convergence of Principal Component Algorithms”, **Neural Networks**, vol.8,no.1, 1995
[9] T.D. Sanger, “An Optimality Principle for Unsupervised Learning”, **Advances in Neural Information Systems**, ed. D.S. Touretzky, Morgan Kaufmann, 1989
[10] L. State, C. Cocianu, P. Vlamos, "Attempts in Using Statistical Tools for Image Restoration Purposes", **Proceedings of SCI2001**, Orlando, USA, July 22-25, 2001.
[11] L. State, C. Cocianu, P. Vlamos, "A Regressive Technique of Image Restorations", **Proceedings of the 29th ICC&IE**, Nov. 1-3, 2001, Montreal, Canada
[12] L. State, C. Cocianu, P. Vlamos, "A Multirezolution Based Approach of Data Compression/ Decompression", **Proceedings of the 29th ICC&IE**, Nov. 1-3, 2001, Montreal, Canada

5. APPENDIX

The input prototype The decompressed image

Figure 3

The input prototype The decompressed image

Figure 4

The input prototype The decompressed image

Figure 5

The Karhunen-Loève compression

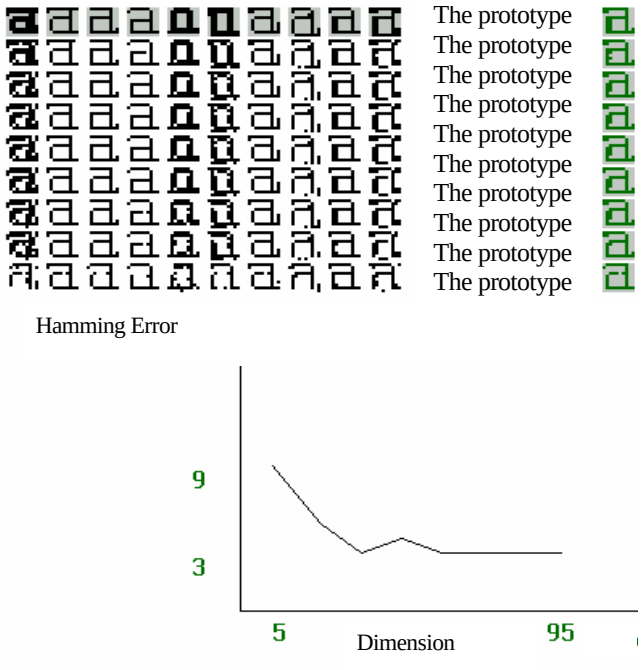

Figure 6

The RLS compression

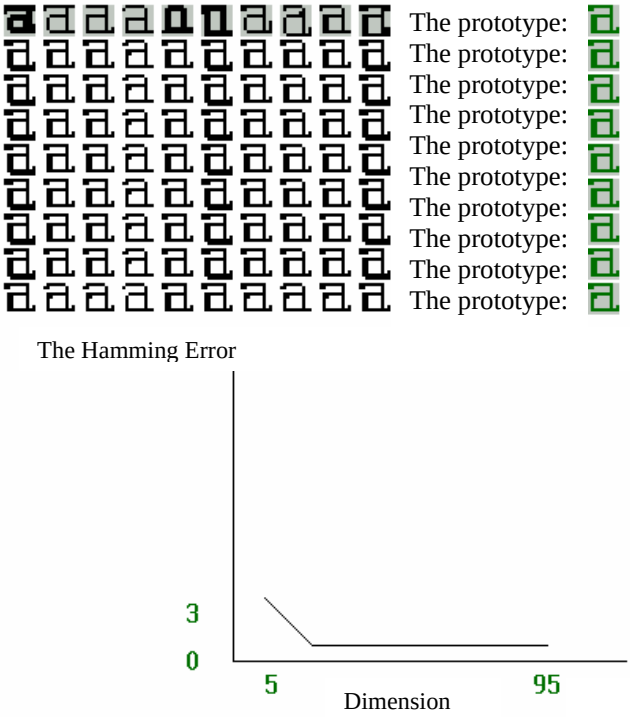

Figure 7