A NEW CLASS OF APEX–LIKE PCA ALGORITHMS

Simone Fiori, Aurelio Uncini and Francesco Piazza

Dept. Electronics and Automatics – University of Ancona (Italy)
E-mail:simone@eealab.unian.it

ABSTRACT

One of the most commonly known algorithm to perform neural Principal Component Analysis of real-valued random signals is the Kung-Diamantaras' Adaptive Principal component EXtractor (APEX) for a laterally-connected neural architecture. In this paper we present a new approach to obtain an APEX-like PCA procedure as a special case of a more general class of learning rules, by means of an optimization theory specialized for the laterally-connected topology. Through simulations we show the new algorithms can be faster than the original one.

1. INTRODUCTION

Principal Component Analysis (PCA) of multivariate random signals is a well-known statistical data analysis technique [1, 7]. It is possible to show that a linear transformation $\mathbf{z} = \mathbf{W}^t\mathbf{x}$ of a given multiple random signal $\mathbf{x}$ into a new random signal $\mathbf{z}$ with less components than $\mathbf{x}$, such that:

- the transformed signal power is maximized under suitable constraints [4];
- the transformed scalar signals are statistically decorrelated [4];
- the signal $\mathbf{x}$ is optimally represented by $\mathbf{z}$ (in the mean squared reconstruction error sense) [4];
- a proper measure of the uncertainty of $\mathbf{z}$ is maximized [9],

can be obtained by assuming $\mathbf{W} = \mathbf{F}$, where $(\mathbf{F}, \cdot)$ is a PCA of $\mathbf{x}$. (The formal definition of PCA in terms of matrix pairs $(\mathbf{F}, \mathbf{D})$ can be found in [3].) Matrix $\mathbf{F}$ contains eigenvectors (normalized to unitary norms) of the covariance matrix of the analyzed signal, while $\mathbf{D}$ contains the powers of the Principal Components arranged in a descending order.

In the literature several algorithms are known that allow the extraction of the (unique) PCA of a signal from itself. The most commonly used are those by Sanger (Generalized Hebbian Algorithm, GHA, [8]), Kung-Diamantaras (APEX, [5, 6, 7]). All of these methods are characterized by different architectural complexity, convergence speed properties and numerical precision at the equilibrium.

In this paper we deal with one of these: the Adaptive Principal component EXtractor (APEX, [6, 7]) based on a laterally-connected neural architecture. It has a wide relevance in the field of analog implementations because it is characterized by a very low complexity. Here we derive a new class of PCA algorithms based on the laterally-connected neural architecture, arising from a simple optimization theory specialized for this topology. Such a class contains, as a special case, an APEX-like algorithm, but it contains also a subclass of algorithms that show a smaller architectural complexity and interesting convergence features when compared with the original one.

NOTATION. In the following $E[\cdot]$ returns the mathematical expectation of the argument; operator $\mathtt{SUT}[\mathbf{A}]$ returns the *strictly upper triangular* part of the square matrix $\mathbf{A}$; the i:th entry of the generic vector $\mathbf{v}$ is denoted with v_i.

2. THE LATERALLY-CONNECTED NEURAL ARCHITECTURE

Kung and Diamantaras realized a Principal Component analyzer using a linear neural network described by the following neural scheme:

$$\mathbf{y} = \mathbf{W}^t\mathbf{x} + \mathbf{L}^t\mathbf{y} \,, \tag{1}$$

with a proper unsupervised learning rule. The input vector $\mathbf{x} \in \mathcal{R}^p$, the output vector $\mathbf{y} \in \mathcal{R}^m$ (with $m \leq p$, arbitrarily fixed), the direct-connection $p \times m$ weight-matrix $\mathbf{W}$ and the lateral-connection $m \times m$ weight-matrix $\mathbf{L}$ are intended to be evaluated at the same temporal instant. The columns of $\mathbf{W}$ and $\mathbf{L}$ are named in the following way: $\mathbf{W} = [\mathbf{w}_1\ \mathbf{w}_2\ \cdots\ \mathbf{w}_m]$, $\mathbf{L} = [0\ \mathbf{l}_2\ \cdots\ \mathbf{l}_m]$. Notice that, being $\mathbf{L}^t$ a strictly lower-triangular square matrix (i.e. $L_{ik} = 0$ if $i \geq k$), this neural network is hierarchical *not* recurrent. The original learning rule for the weight-matrix $\mathbf{W}$ was:

$$\Delta\mathbf{W} = \eta[\mathbf{X}\tilde{\mathbf{Y}} - \mathbf{W}\tilde{\mathbf{Y}}^2] \,, \tag{2}$$

This research was supported by the Italian MURST.

and the learning rule for the weight-matrix $\mathbf{L}$ was:

$$\Delta\mathbf{L} = -\eta \mathtt{SUT}[\mathbf{Y}\tilde{\mathbf{Y}}] - \eta\mathbf{L}\tilde{\mathbf{Y}}^2 \,, \qquad (3)$$

where η is a positive learning rate, $\mathbf{X}$ is a $p \times m$ matrix, $\mathbf{Y}$ and $\tilde{\mathbf{Y}}$ are $m \times m$ matrices, defined by:

$$\mathbf{X} = [\underbrace{\mathbf{x}\ \mathbf{x}\ \cdots\ \mathbf{x}}_{m}] \,,\ \mathbf{Y} = [\underbrace{\mathbf{y}\ \mathbf{y}\ \cdots\ \mathbf{y}}_{m}] \,,$$
$$\tilde{\mathbf{Y}} = \mathrm{diag}(y_1, y_2, \ldots, y_m) \,.$$

Kung and Diamantaras were able to prove the convergence of the above algorithm under some conditions. In particular, we can restate their result saying that:

(Theorem.) *Let* $\mathbf{x}$ *be a* p*-components real random signal, zero-mean, with a finite covariance endowed with* m *non-null distinct eigenvalues, and let* $(\mathbf{F}, \mathbf{D})$ *the unique PCA of* $\mathbf{x}$*. Let DKN be the neural-net described by (1) trained by means of the learning pair (2)-(3). If the rate* η *is chosen so small that the behavior of the algorithm is asymptotically stable and the initial entries of* $\mathbf{W}$ *are small random numbers and* $\mathbf{L}(0) = 0$*, then in the mean it holds true that:*

$$\lim_{t\to\infty} \mathbf{L}(t) = \mathbf{0} \,,$$
$$\lim_{t\to\infty} \mathbf{W}(t) = \mathbf{F} \,,$$
$$\lim_{t\to\infty} E[\mathbf{y}(t)\mathbf{y}^t(t)] = \mathbf{D} \,.$$

In other words, under the above conditions the DKN asymptotically becomes, in the mean, a principal component analyzer. □

Strictly speaking, they proved that if η is sufficiently small and suitable initial conditions are assumed, then in the mean $(\mathbf{W}, E[\mathbf{y}\mathbf{y}^t]) \to (\mathbf{F}, \mathbf{D})$, where $(\mathbf{F}, \mathbf{D})$ is a PCA of the signal $\mathbf{x}$. We call the above issue the *Kung-Diamantaras Result* (KDR).

3. THE ψ–APEX CLASS

In the following Subsection a new class of APEX–like algorithms is presented. Later, differences and similarities between our new algorithms and other ones will be discussed.

3.1. APEX–like algorithms based on an optimization formulation

A PCA transformation is such that the transformed signals (with the above symbology, $\mathbf{z} = \mathbf{W}^t\mathbf{x}$) are characterized by maximum variance. Furthermore, from the formal definition of PCA it is known that, at the equilibrium, any unique PCA vector $\mathbf{w}_i$ must be orthogonal with respect to each other and with an unitary norm.

These targets can be thought as separated objectives to be attained by means of laterally-connected neural topology. More formally we can state the following:

(Proposition.) *It is possible to define a pair* (J, C) *of objective functions whose extremization process yields a class of PCA algorithms containing, as a special case, an APEX-like.* □

Functions J and C can be properly fixed by examining the structure of a generic output signal y_i from (1) squared. Direct calculations show:

$$y_i^2 = (\mathbf{w}_i^t\mathbf{x})^2 + (\mathbf{l}_i^t\mathbf{y})^2 + 2(\mathbf{w}_i^t\mathbf{x})(\mathbf{l}_i^t\mathbf{y}) \,. \qquad (4)$$

The first term at the right hand contains in the mean the power of the transformed signal $z_i = \mathbf{w}_i^t\mathbf{x}$, while the second term at the right hand of the above equation contains in the mean a linear combination of the output signals cross-correlation, in fact it holds true that $E[(\mathbf{l}_i^t\mathbf{y})^2] = \mathbf{l}_i^t E[\mathbf{y}\mathbf{y}^t]\mathbf{l}_i$. By definition of PCA, the first one has to be maximized under the constraint $\mathbf{w}_i^t\mathbf{w}_i = 1$ ([4]), while the second one must be zeroed.

Here we propose to use the *direct-connection* adaptation to maximize the powers of the transformed signal by maximizing the following objective function:

$$J(\mathbf{W}, \mathbf{L}) \stackrel{\mathrm{def}}{=} \sum_{i=1}^{m} E[y_i^2] + \frac{1}{2}\sum_{i=1}^{m}(\mathbf{w}_i^t\mathbf{w}_i - 1)\mu_i \,, \qquad (5)$$

with respect to $\mathbf{W}$ only. In the above equation μ_i are so-called *Lagrange multipliers* to be determined by imposing the constraints $\mathbf{w}_i^t\mathbf{w}_i = 1$ in the equilibrium conditions $\frac{\partial J}{\partial \mathbf{w}_i} = 0$. It is important to notice that by definition of $\mathbf{L}$, a scalar product $(\mathbf{l}_i^t\mathbf{y})$ *does not depend on* $\mathbf{w}_i$, but only on $\mathbf{w}_j$ for $j < i$, then from equations (4) and (5) we obtain:

$$\frac{\partial J}{\partial \mathbf{w}_i} = 2E[(\mathbf{w}_i^t\mathbf{x})\mathbf{x}] + 2E[(\mathbf{l}_i^t\mathbf{y})\mathbf{x}] + \mu_i\mathbf{w}_i$$
$$= 2E[y_i\mathbf{x}] + \mu_i\mathbf{w}_i \,,$$

therefore the optimum $\mathbf{w}_i$ satisfies:

$$\mathbf{w}_i^t\frac{\partial J}{\partial \mathbf{w}_i} = 2E[y_i(\mathbf{w}_i^t\mathbf{x})] + \mu_i = 0 \,,$$

and optimum μ_i is $\mu_i = -2E[y_i z_i]$.

If the Gradient Steepest Ascent (GSA) method is used to adapt each $\mathbf{w}_i$, that means $\Delta\mathbf{w}_i = +\frac{1}{2}\eta\frac{\partial J}{\partial \mathbf{w}_i}$, the stochastic learning rule for $\mathbf{W}$ reads:

$$\Delta\mathbf{W} = \eta[\mathbf{X}\tilde{\mathbf{Y}} - \mathbf{W}\tilde{\mathbf{Y}}\tilde{\mathbf{Z}}] \,, \qquad (6)$$

where:

$$\tilde{\mathbf{Z}} = \mathrm{diag}(z_1, z_2, \ldots, z_m) \,.$$

and true gradient of J has been replaced by its stochastic instantaneous approximation.

Finally, we choose to adapt the *lateral-connection* weight-matrix $\mathbf{L}$ only, in order to *minimize* a cost function defined as:

$$C(\mathbf{W}, \mathbf{L}) \stackrel{\mathrm{def}}{=} \sum_{i=1}^{m} E[y_i^2] + \sum_{i=1}^{m}(\mathbf{l}_i^t\mathbf{l}_i)\psi_i \,, \qquad (7)$$

where a set of m Lagrange multipliers ψ_i has been introduced for the constraints $||\mathbf{l}_i||^2 = 0$ (that have to be reached at the equilibrium to preserve KDR) and to add to the system a number of degree of freedom. Besides, it is interesting to recognize that those constraints also embed a *regularization* property on the global criterion [2]. As it can be directly proved by using standard Kuhn-Tucker theory [2], under these constraints there are *no theoretical reasons to force functions ψ_i to assume any particular shape.*

This second objective function C can be minimized, with respect to the variable matrix $\mathbf{L}$, by means of a Gradient Steepest Descent (GSD) method $\Delta\mathbf{l}_i = -\frac{1}{2}\eta\frac{\partial C}{\partial \mathbf{l}_i}$. From equations (4) and (7) it follows:

$$\frac{\partial C}{\partial \mathbf{l}_i} = 2E[y_i\mathbf{y}^{[i]}] + 2\psi_i\mathbf{l}_i \ ,$$

where $\mathbf{y}^{[i]} = [y_1\ y_2\ \cdots\ y_{i-1}\ \cdots\ 0\ \cdots\ 0]^t$ for $2 \leq i \leq m$, and $\mathbf{y}^{[1]} = [0\ 0\ \cdots\ 0\ 0]^t$. By rewriting GSD equations in matrix notation, ignoring again expectation operator, the new stochastic learning rule for $\mathbf{L}$ reads:

$$\Delta\mathbf{L} = -\eta\mathtt{SUT}[\mathbf{Y}\tilde{\mathbf{Y}}] - \eta\mathbf{L}\tilde{\mathbf{\Psi}} \ , \qquad (8)$$

with matrix $\tilde{\mathbf{\Psi}}$ being defined as:

$$\tilde{\mathbf{\Psi}} = \mathrm{diag}(\psi_1, \psi_2, \ldots, \psi_m) \ .$$

Rule (8) provides minimization of the cross-correlation between the network's output signals.

Now we have all the elements to propose the following definition, relative to the class of algorithm represented by the above new neural learning rules:

(Definition.) *The family of learning rules described by equations (6) and (8) is called the ψ–*APEX *Principal Component analyzer class. The special element in this family with $\tilde{\mathbf{\Psi}} = \tilde{\mathbf{Y}}^2$ is called y^2–*APEX. □

Notice that y^2–APEX is not the same algorithm as the original APEX, but as $\mathbf{L} \to 0$ also $\tilde{\mathbf{Z}} \to \tilde{\mathbf{Y}}$, thus these algorithms asymptotically behave in the same way, and we call it *APEX-like.*

It is also important to remark that, apart from further stability considerations, the choice of the multiplying functions $\psi_i(t)$ is free. In fact, we can adopt as ψ_i any suitable arbitrarily chosen function that guarantees the asymptotic stability of the global learning process and good performances of the Principal Component analyzing algorithm.

3.2. Discussion

In practice, in our experiments we have examined the following three cases:

1. all the $\psi_i(t)$ are chosen null;
2. $\psi_i(t)$ are arbitrarily chosen non-null constant values $\bar{\psi}_i$;
3. the $\psi_i(t)$ are assumed as particular non-constant functions of the unique variables $y_i(t)$.

Roughly speaking, we can identify the special PC's extractor obtained by vanishing free functions $\psi_i(t)$ as the *0–APEX* algorithm, whose descriptive equations are:

$$\Delta\mathbf{W} = \eta[\mathbf{X}\tilde{\mathbf{Y}} - \mathbf{W}\tilde{\mathbf{Y}}\tilde{\mathbf{Z}}] \ , \qquad (9)$$

$$\Delta\mathbf{L} = -\eta\mathtt{SUT}[\mathbf{Y}\tilde{\mathbf{Y}}] \ . \qquad (10)$$

In a computational-complexity point of view this algorithm is the most interesting one, since it requires a smaller amount of operations than the original APEX, as shown in Table 1. The above rule recalls the *linearized Rubner-Tavan model* that the 0–APEX asymptotically behaves like. (For knowing details about Rubner-Tavan approach readers please refer to [7].)

We observed that the term y_i^2 in each of the (3) is too much large and can also lead the algorithm very far from the right solution. Thus, when non-constant non-null functions ψ_i are used, we found useful they satisfy this constraint: Each $\psi_i(t)$ should be a positive function that grows less than t^2 at least for large $|t|$. For instance, we found good results with $\psi_i = |y_i|$. Other suitable choices are of course possible.

Algorithm	Complexity (Operations)
GHA	$2pm + \frac{1}{2}(m^2+m)(p+1)$
APEX	$3pm + \frac{3}{2}m^2 - \frac{1}{2}m$
0–APEX	$3pm + m^2$

Table 1: Complexity comparison.

Table 1 provides estimates of the architectural complexity of the neural networks in terms of the number of elementary operations required by the corresponding learning rules with respect to the network dimensions. We define an "operation" as a product eventually followed by a sum.

4. EXPERIMENTAL RESULTS

To assess our theoretical analysis and compare algorithms' performances, we performed simulations by using Sanger's GHA, standard APEX and our new algorithms belonging to the ψ–APEX class.

Such PCA algorithms have been run with a network input signal $\mathbf{x} = \mathbf{Q}\mathbf{s}$, where $\mathbf{Q}$ is a $p \times p$ orthonormal matrix ($\mathbf{Q}^t\mathbf{Q} = \mathbf{I}$) randomly generated, and $\mathbf{s}$ contains p mutually uncorrelated zero-mean random signals s_i with different powers $\sigma_i^2 = E[s_i^2]$. Signals s_i are placed in $\mathbf{s}$ so that their powers are decreasingly ordered, i.e. $\sigma_i^2 > \sigma_j^2$ if $i < j$. This implies that the first m Principal Components of $\mathbf{x}$ (with $m < p$) are the first m column-vectors of $\mathbf{Q}$.

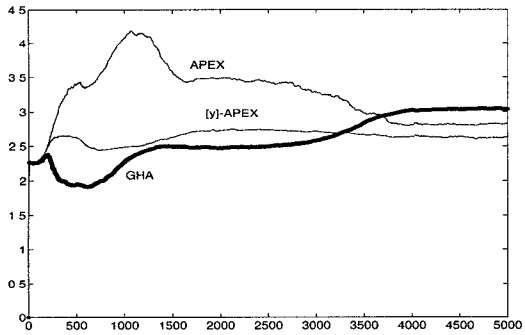

Figure 1: Convergence speed comparison.

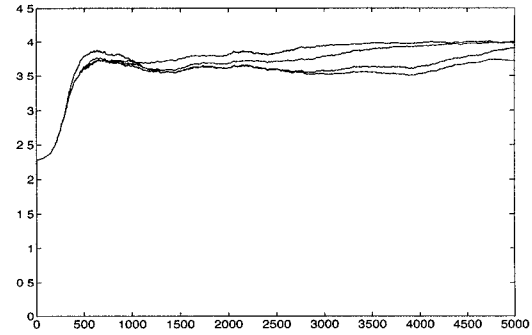

Figure 2: Comparison of 4 ψ–APEX algorithms.

Each algorithm starts from the same initial conditions, that are random for $\mathbf{W}$ and null for $\mathbf{L}$.

In order to compare the convergence speed of the new algorithms with those of the GHA and APEX, a suitable measure of convergence δ is used. This measure is defined as $\delta(\mathbf{W}) = ||\mathbf{W} - \tilde{\mathbf{Q}}||_F$, where $\tilde{\mathbf{Q}}$ is that matrix whose columns are the first m of $\mathbf{Q}$, and $||\cdot||_F$ denotes the Frobenius norm. Note that the quantity δ may converge to different values since recovering the columns of $\mathbf{Q}$ is sign-blind. Simulation presented in Figure 1 concerns GHA, APEX and $|y|$–APEX (that means $\psi_i = |y_i|$) algorithms. The above results are obtained with a learning stepsize $\eta = 0.01$, network's dimension $p = 10$ and $m = 5$. Powers σ_i^2 were assumed from the exponential law $\sigma_i^2 = 2^{2-i}$ (where i ranges from 1 to p) in order to keep a good eigenvalue spread. New $|y|$–APEX performs well: its convergence toward KDR looks faster and its precision seems to be comparable at all with that of the other algorithms.

Figure 2 shows typical courses of APEX, 0–APEX, $|y|$–APEX and y^2–APEX compared together for $\sigma_i^2 = 0.1(p - i + 1)$. Here input signals have small powers one close to another. In this case all algorithms after few steps behave almost identically, therefore the 0–APEX is the most convenient one.

5. CONCLUSION

In [7] a wide generalization of the standard APEX has been presented, but to our knowledge special attention has not been paid to particularization nor tests have been performed in order to discover their features, hence we believe this paper points out some new issue and contains new contributions. Extensions of the new method to the complex-valued case is currently under investigation.

6. REFERENCES

[1] P.F. BALDI AND K. HORNIK, *Learning in linear neural networks: A survey*, IEEE Trans. on Neural Networks, Vol. 6, No. 4, pp. 837 – 858, July 1995

[2] A. CICHOCKI AND R. UNBEHAUEN, *Neural networks for optimization and signal processing*, J. Wiley Ltd., 1993

[3] P. COMON, *Independent Component Analysis, a new concept ?*, Signal Processing, Vol. 36, pp. 287 – 314, 1994

[4] J. KARHUNEN, *Optimization criteria and nonlinear PCA neural networks*, Proc. of International Joint Conference on Neural Networks (IJCNN), pp. 1241 – 1246, 1994

[5] S.Y. KUNG, *Constrained Principal Component Analysis via an orthogonal learning network*, Proc. of International Symposium on Circuits and Systems (ISCAS), pp. 719 – 722, 1990

[6] S.Y. KUNG AND K.I. DIAMANTARAS, *A network learning algorithm for adaptive principal component extraction*, Proc. of International Conference on Acoustics, Speech and Signal Processing (ICASSP), 1990, pp. 861 – 861

[7] S.Y. KUNG AND K.I. DIAMANTARAS, *Principal Component Neural Networks: Theory and Applications*, J. Wiley, 1996

[8] T.D. SANGER, *Optimal unsupervised learning in a single-layer neural network*, Neural Networks, Vol. 2, pp. 459 – 473, 1989

[9] L. XU, *Theories for unsupervised learning: PCA and its nonlinear extension*, Proc. of International Joint Conference on Neural Networks (IJCNN), pp. 1252 – 1257, 1994