Non-linear Complex-Valued Extensions of Hebbian Learning: An Essay

Simone Fiori

Facoltà di Ingegneria dell'Università di Perugia
Polo Scientifico e Didattico del Ternano,
Loc. Pentima bassa, 21, I-05100, Terni (Italy)

E-mail: fiori@unipg.it

Manuscript accepted for publication in:
Neural Computation

Pages: 68, **Figures**: 11, **References**: 97

— August 04, 2004 —

Non-linear Complex-Valued Extensions of Hebbian Learning: An Essay

Simone Fiori

Abstract

The Hebbian paradigm is perhaps the most known unsupervised learning theory in connectionism. It has inspired a wide research activity in the artificial neural network field because it embodies some interesting properties such as locality and the capability of being applicable to the basic weight-and-sum structure of neuron models. The plain Hebbian principle, however, also presents some inherent theoretical limitations that make it unpractical in most cases. Therefore, modifications of the basic Hebbian learning paradigm have been proposed over the last twenty years in order to design profitable signal/data processing algorithms. Such modifications led to the principal-component-analysis-type class of learning rules along with their non-linear extensions. The aim of this essay is primarily to present part of the existing fragmented material in the field of principal component learning within a unified view and contextually to motivate and present extensions of previous works on Hebbian learning to complex-weighted linear neural networks. This work benefits from previous studies on linear signal decomposition by artificial neural networks, non-quadratic component optimization and reconstruction error definition, neural parameters adaptation by constrained optimization of complex-valued learning criteria and orthonormality expression via the insertion of topological elements in the networks or by modifying the network learning criterion. In particular, the considered learning principles and their analysis concern complex-valued principal/minor component/subspace linear/non-linear rules for complex-weighted neural structures, both feedforward and laterally-connected.

1 Introduction

The basic assumptions of connectionism on neural learning are reasonably non-restrictive and rather appealing from an engineering point of view. These may be summarized by:

- The locality principle, which states that the modifications in the connections depend upon the activities of pre- and post-synaptic neurons and do not depend on the activity levels of the other neurons;
- The principle stating that the modification of synapses is slow compared with the characteristic time of neuron dynamics;
- The forgetting principle, stating that if either the pre- or post-synaptic neurons or both are silent then no synaptic changes take place except for exponential decay.

Classical contributions to this analysis are [11, 61].

Perhaps, the most influential work in the history of connectionism is the contribution of the neuro-psychologist Donald Hebb. In the book [47], Hebb presented a theory of behavior based on the physiology of the nervous system. He reduced the types of physiological evidence into two main categories: The existence and properties of continuous cerebral activity, and the nature of synaptic transmission in the central nervous system. Hebb combined these principles in order to develop a theory of how learning occurs within an organism. As a matter of fact, Hebbian learning and cell/synapses creation/death are kinds of learning for which there is neural evidence.

In Hebbian learning, the weights between neurons are adjusted so that each weight better captures the relationship between the units. The Hebbian learning rule specifies that the weight of a connection between two units should be varied in proportion to the product of their activation: It states that the connections between two neurons might be strengthened if the neurons fire simultaneously. As a result, if a unit fires when presented with a pattern, the weights from the active inputs are strengthened, so that the unit will respond to the same pattern even better in the future. In a network of many units, however, a mechanism is necessary to prevent different neurons from encoding the same features. In particular, a way to encourage sparse representations between units is to promote unit decorrelation through lateral inhibition [7, 8, 9, 38]. Lateral inhibition is a distinguishing feature of some artificial unsupervised neural networks. It provides a mechanism that makes a network of neurons hierarchic and with which the neurons compete for the right to generate a response to the incoming

stimuli. This kind of competition enables the abstract neuronal units to form receptive fields that are sensitive to stimuli coming from different regions of the input space [85]. Mutual inhibition between units may be achieved by training the lateral connections via an anti-Hebbian rule: Whenever two units in a layer are active simultaneously, the connection between them becomes more inhibitory, so that their correlation is decreased, and their joint activity will be discouraged in the future [38]. For the standard weight-and-sum neuron model, Hebbian learning makes the synapses be correlators. In some circumstances, the plain Hebbian learning continually strengthens a unit's weights without bound [5, 11, 67]. Within networks endowed with the appropriate structure, however, the mentioned unboundness problem is not present (consider, for instance, the recurrent error-correction model discussed in [44]).

In 1977, Amari [2] investigated a neural theory of association and concept formation and in 1982, Oja [73] studied a stabilized version of classical Hebb rule. They observed that, under fair conditions, the above stabilized learning rule makes the neuron capture the most powerful eigenvector of the input covariance matrix or, in other terms, the rule makes the neuron able to extract the principal component from the input multivariate random signal.

Since Amari-Oja's pioneering work, several new learning algorithms have been proposed for extending the one-unit principal-component neural system to complete neural networks. The classical contribution in the field of principal component networks may be traced back to Sanger in [82], who used an on-line version of the well-known Gram-Schmidt orthogonalization algorithm. Also, Rubner and Tavan [81] and Kung and Diamantaras [25] introduced a linear neural network endowed with lateral inhibitory connections for achieving output decorrelation. Kung-Diamantaras' algorithm is often referred to as adaptive principal component extractor (APEX). Over recent years, several authors introduced different principal component analysis (PCA) rules for generalizing classical ones [1, 6, 22, 27, 29, 30]. A PCA-related argument is principal subspace analysis (PSA), that concerns the computation of the subspace of the input space spanned by the principal eigenvectors [74]. Extended discussions on Hebbian, anti-Hebbian and modified Hebbian learning paradigms and their properties in the context of component/subspace analysis may be found e.g. in

[4, 39, 44, 90] and references therein.

Likely, one of the reasons of the success of stable Hebbian learning theory is its usefulness for solving many signal processing problems, as illustrated for instance in [22, 56, 62, 77] and references therein, where extracting the first principal components is shown to be of prime importance. Nevertheless, it has been clearly demonstrated that computing the *last* principal components of a data-sequence, i.e. those principal components endowed with the smallest (non-zero) power, may be very useful as well [40, 55, 63, 83, 93]. The extraction of the last principal components is commonly referred to as minor component analysis (MCA).

Non-linear versions of standard Hebbian or anti-Hebbian learning rules, namely those rules based on the optimization of non-quadratic cost functions, may make the involved linear neural networks capable of performing the independent component analysis (ICA) of incoming signals (for a review of ICA in the neural-network field see e.g. [12]). An early and pervasive physiological example of such behavior was presented by Barlow and Földiák in [10]: They recalled that the information about the basic tastes (sweet, salty, bitter and sour) is not carried on by separate fibers, instead, each fiber carries a mixed signal with different relative sensitivities to the four basic tastes. The operation of separating the four mixed quantities, performed by the nervous system, could not be explained by simple decorrelation rule. Thus, Barlow and Földiák proposed a modified (anti-Hebbian) rule for synaptic-strength tuning based on the assumption of non-negativity of the taste-variables[1]. Classical contributions on non-linear extensions to PCA are [52, 53, 76].

The aim of the present essay is to describe extensions of previous works to complex-weighted linear neural networks in a formal way. The guideline of the presented research is complex-valued gradient-based optimization of non-quadratic learning criteria. The presented derivations subsume some contributions found in the scientific literature: We recall interesting contributions from advanced statistical theory of learning and try to relate sparsely-published valuable works on these topics in a unifying view. These works span the following topics:

[1] This is what is currently referred to as non-negative independent component analysis [80].

- Linear signal decomposition by artificial neural networks. This widely known theory allows expressing a multivariate signal as a linear combination of features or components which enjoy some special statistical property (like incorrelatedness or independency). The decomposition consists in jointly finding a basis for the signal and the corresponding components. The decomposition is then achieved by components optimization or by reconstruction error minimization.

- Non-quadratic component optimization and reconstruction error definition. Both rely on the proper selection of the involved non-quadratic learning criteria via available suitable interpretations of Hebbian learning such as the ones suggested by Song, Yilong and Feng [86] and by Sudjianto and Hassoun [87].

- Neural parameters adaptation by constrained optimization of complex-valued learning criteria. This is the basis for neural learning with physical/structural constraints when complex-valued input signals or input/output signal pairs are to be dealt with. The constraints considered here are related to the ortho-normality of neural connection matrices.

- Orthonormality expression via the insertion of topological elements in the networks or by modifying the network learning criterion. Topological constraints are given e.g. by lateral connections that force the neurons in a network to encode uncorrelated features. Modified learning criteria are created e.g. by the Lagrange multiplier method for equality constraints.

The development of the theory of linear complex-weighted neural networks is supported by pervasive engineering and physiological motivations. In the supervised-learning field, for instance, it was reported (see [66]) that complex-valued back-propagation models train faster than conventional back-propagation networks, are more resistant to local minima and exhibit better generalization ability. By extending previous works proposed independently by some researchers [13, 41], Nitta reported that complex-valued back-propagation architectures exhibit reduced probability of learning standstill and are able to perform transformations of geometric figures in the complex plane, which their real-valued counterparts are unable to effect [70, 71, 72]. Also, following the pio-

neering work by Widrow (see e.g. [91]) on merging neural networks and adaptive filters, recently Hanna and Mandic [43] presented a complex-valued non-linear gradient-descent learning algorithm for a non-linear neural adaptive filter with adaptive complex-valued activation function; such structure is mentioned to be beneficial when dealing with signals that have rich dynamical behavior. As other applications, Miyauchi *et al.*, proposed an interpretation of optical flow based on complex-weighted neural networks [68] while Müezzinoğlu, Güzeliş and Zurada [69], recently proposed a design method for complex-weighted multistate Hopfield memory, which appears as a generalization of the conventional Hopfield model that can be an efficient tool to process static integral information: The new method was shown to outperform the generalized Hebbian rule, which has yet constituted the only learning rule for this model, in associating phase-modulated integral information. Also, from a biological point of view, the current interest in pulse coded neural networks [59] has created a need for a mathematically compact way of dealing with phase as well as magnitude of neural signals, that may be easily found in the complex-valued representations.

In the unsupervised field, a first extension to classical Sanger's PCA learning theory to the complex plane has been presented by De Castro, De Castro, Amaral and Franco in [23] with application to optimal image compression in the spectral domain. An application of non-linear MCA extended to the complex domain has been presented recently in [33], with application to robust beamforming. Robust beamforming is a well-known signal processing technique allowing for performing spatial filtering of a signal source in presence of spatial noise and other disturbing sources by means of an array of electromagnetic antennas or electro-mechanical microphones, provided that the direction of arrival of the primary source is known. A beamformer may be realized by a complex-weighted neural unit fed with the Fourier transform of the measured signals, hence its complex-valued nature. A bio-engineering application of adaptive beamforming finds e.g. in hearing aid [54]. A way to train the beamforming neuron is to force it to solve a correlation-matrix minimal-eigenvalue extraction problem, which may be formulated as an MCA problem [33]. Further applications of complex-valued principal/minor component analysis have been reviewed and discussed in [62]. The theory of complex-weighted network learning has led to effective

independent component analysis algorithms for complex-valued statistically independent signals (see e.g. [28, 31] and references therein). Typical applications of complex-valued ICA algorithms are to electromagnetic phenomena analysis and modeling [24, 32]. Further examples of applications of complex-valued ICA algorithms are frequency-domain algorithms for separation of sound signals and for biomedical data analysis [21]. Also, in [66], Michaels and Upadhyaya provided an architecture for signal and image processing (focused on capacitance and inductance sensors) and for modeling the timing and pattern of neural signals in biological settings. In particular, they introduced a complex-valued associative memory theory and showed that in its iterative form, its learning procedure is a complex-valued Hebbian adapting algorithm which uses association rather than error correction.

Paper organization. Section 2 is devoted to the complex-valued Hebbian learning theories stemming from the optimization of generalized network output power, while section 3 presents those extended Hebbian learning theories stemming from generalized reconstruction error minimization. In particular, section 2.1 presents a learning theory for a linear feedforward architecture, where the constraints on symmetry/hierarchy are embodied in the learning criterion, while section 2.2 introduces a theory for Rubner-Tavan's [81] laterally-connected network structure. Both sections present a comparison with the real-valued networks/rules counterparts. Section 2.3 presents a possible way of choosing the non-linearity involved in the non-quadratic learning criteria based on Sudjianto-Hassoun fruitful 'maximum-mismatch' principle [87]. Section 2.4 presents some formal results pertaining to the convergence of learning rules in the one-unit case, while section 2.5 deals with the multi-unit case. Section 3.1 illustrates a learning theory stemming from generalized reconstruction error minimization for a symmetric network and section 3.2 discusses some possible ways to select the involved non-quadratic error measures, while section 3.3 considers the hierarchic case and the corresponding choices of involved non-linearities. Section 4 concludes the paper.

2 Complex-Valued Hebbian Learning by Non-Quadratic Output Optimization

The aims of this section are to present a unified view and to present and discuss generalizations of network's architectures and learning rules for principal component/subspace analysis of complex-valued signals. Generalization is achieved by extending the classical architectures to complex-weighted neural networks and by extending the classical learning criteria to non-quadratic optimization objectives by the introduction of non-linear functions. A possible choice of the involved non-quadratic function is also discussed.

In the present section, the Lagrange multiplier method is used extensively in order to construct suitable criteria for learning under orthonormality constraints. It is worth recalling how the Lagrange multiplier method for equality constraints works in practice, leaving to e.g. reference [14] the details of the theory and of specific computations. Suppose that $J(\mathbf{w})$ and $L_i(\mathbf{w})$, $i = 1, \ldots, m$ and $\mathbf{w} \in \mathbb{R}^p$ are differentiable functions that map $\mathbb{R}^p \rightarrow \mathbb{R}$, and it is wanted to solve:

$$\text{opt } J(\mathbf{w}) \text{ such that } L_j(\mathbf{w}) = 0 \ , \ j = 1, \ldots, m \ ,$$

Namely to find the optimum (maximum or minimum) of the function $J(\mathbf{w})$ that satisfy the equality constraints $L_j(\mathbf{w}) = 0$. This is equivalent to solving the following problem:

$$\text{opt } \left[J(\mathbf{w}) - \sum_{j=1}^{m} \lambda_j L_j(\mathbf{w}) \right] \ ,$$

for $\mathbf{w}$ and the λ_j, without restrictions. The variables $\lambda_j \in \mathbb{R}$ are termed Lagrange multipliers. It is not difficult to recast the above optimization problem in terms of e.g. matrix-type variables, if necessary.

In practice, the optimal multipliers λ_j^{opt} may be found analytically through a multiplier elimination method [14]. Then, the optimal parameter-vector $\mathbf{w}$ may be found e.g. by a gradient-based algorithm, namely:

$$\frac{d\mathbf{w}}{dt} = \pm \left(\frac{\partial J(\mathbf{w})}{\partial \mathbf{w}} \right)^{\text{opt}} \ , \tag{1}$$

$$\left(\frac{\partial J(\mathbf{w})}{\partial \mathbf{w}} \right)^{\text{opt}} \stackrel{\text{def}}{=} \frac{\partial}{\partial \mathbf{w}} J(\mathbf{w}) - \sum_{j=1}^{m} \lambda_j^{\text{opt}} \frac{\partial}{\partial \mathbf{w}} L_j(\mathbf{w}) \ , \tag{2}$$

where the '+' sign denotes maximization and the '−' sign denotes minimization.

As a Reader's guidance, the contents of the present section may be anticipated as:

- Learning principal/minor component/subspace by complex-valued gradient-based optimization of non-quadratic criteria.
- Learning in a orthogonally-constrained network by complex-valued gradient-based optimization of non-quadratic criteria and output decorrelation by lateral inhibition.
- Choice of the non-linear criteria via Sudjianto-Hassoun interpretation of Hebbian learning, illustrated by numerical examples.
- Analysis of static and dynamical properties of one-unit complex-valued PCA/MCA neural systems, illustrated by numerical examples.
- Multi-unit complex-valued component/subspace networks: Notes on dynamical properties and on the design of learning algorithms that agree with the orthonormality constrains.

2.1 Complex-valued gradient-based optimization of non-quadratic criteria

As mentioned in the Introduction, the origin of modern Hebbian learning may be traced back to Oja's work on first principal component analysis learning theory.

Principal component analysis is appropriate e.g. when measures on a number of observed variables have been obtained and it is wished to develop a smaller number of artificial variables (termed principal components) that will account for most of the variance in the observed variables. The principal components may then be used as predictor or criterion variables in subsequent analyses. In particular, first principal component analysis deals with the extraction of the principal component that accounts for the largest fraction of total variance.

Oja's first-principal-component learning rule was based on a single neuron, described by $y(t) = \mathbf{w}^T(t)\mathbf{x}(t)$, where $\mathbf{x}(t) \in \mathbb{R}^p$ represents a stationary multivariate zero-mean random process endowed with finite covariance matrix $\mathbf{\Phi}$,

$\mathbf{w}(t) \in \mathbb{R}^p$ is the neuron's weight vector and $y(t) \in \mathbb{R}$ is the neuron's output signal. The learning task of Oja's neuron is to maximize the variance of its response signal, namely of $E[y^2] = \mathbf{w}^T \mathbf{\Phi} \mathbf{w}$ under the constraint $\mathbf{w}^T \mathbf{w} = 1$. This learning rule reads:

$$\frac{d\mathbf{w}(t)}{dt} = E[\mathbf{x}(t)y(t) - \mathbf{w}(t)y^2(t)] \ . \tag{3}$$

In the above relations, t denotes continuous time and $E[\cdot]$ denotes statistical expectation (ensemble average)[2]. This expression clearly reveals the presence of the Hebbian term $\mathbf{x}(t)y(t)$ and of a stabilizing term, thus it is also referred to as stabilized Hebbian learning equation.

It is interesting to recall that the Oja rule was initially formulated as an approximated normalized Hebbian learning algorithm by using a normalization step that bind the connection vector $\mathbf{w}$ to belong to a unit-radius sphere [73]: In the discrete-time counterpart of such Hebbian learning, the learning stepsize is usually supposed to be very small, therefore it is possible to expand by Taylor series the normalized-Hebbian rule with respect to the learning stepsize and to truncate the expansion to the first-order term. This procedure led to the discrete-time version of the learning rule (3).

In first principal component analysis it is known that the extracted weight vector coincides to the eigenvector of the covariance matrix $\mathbf{\Phi}$ corresponding to its largest eigenvalue. In general principal component analysis, more than one unit-norm weight-vectors are sought for. As it is known that the eigenvectors of a covariance (hence symmetric) matrix are orthogonal to each other, it is necessary to enforce the orthogonality of network's weight-vectors.

Related concepts to PCA are minor component analysis, which consists in extracting the weakest principal components, and principal/minor subspace analysis, which consists in finding (whatever basis of) the linear subspaces spanned by the principal/minor vectors.

In order to generalize Oja's work to the complex domain, a linear neural network may be considered, which is formed by linear complex-weighted units.

[2]More formally, the ensemble average should be denoted by the symbol $E_{\mathbf{x}}[f|\mathbf{w}]$, which denotes conditional expectation of function $f(\mathbf{x})$ with respect to the statistics of $\mathbf{x}$ subject to the hypothesis $\mathbf{w}$. For the sake of conciseness, it will hereafter be simply written in short notation as $E[f]$.

Let the following learning objective function for the neural net be defined:

$$J(\mathbf{w}_k) \stackrel{\text{def}}{=} U(\mathbf{w}_k) + L(\mathbf{w}_k) \ , \ k = 1, \ ..., \ m \ . \tag{4}$$

where $\mathbf{w}_k \in \mathbb{C}^p$ represents the weight vector of the k^{th} neuron. The criterion $U(\cdot)$ contains a nonlinear function of the k^{th} neuron's output $y_k \stackrel{\text{def}}{=} \mathbf{w}_k^H \mathbf{x}$ and is defined as follows:

$$U(\mathbf{w}_k) \stackrel{\text{def}}{=} E[f(\mathbf{w}_k^H \mathbf{x})] \ , \tag{5}$$

where $\mathbf{x} \in \mathbb{C}^p$ is the input vector of the network whose values are drawn from a p-dimensional complex-valued input space and the superscript ^{H} denotes Hermitian transpose (i.e. transpose plus complex conjugation). The number of neurons of the network is denoted here with m, where $m \leq p$.

The function $f(\cdot)$ is a real-valued, positive function of complex-valued argument and is supposed of the form:

$$f(\zeta) \stackrel{\text{def}}{=} g(|\zeta|) \ , \ \zeta \in \mathbb{C} \ , \ g : \mathbb{R}_0^+ \rightarrow \mathbb{R}_0^+ \ . \tag{6}$$

The function $g(u)$ is normally required to be differentiable and convex with a minimum in $u = 0$ in a convenient right-sided neighborhood of the origin. The choice of the function $f(\zeta)$ as a real function of $|\zeta|$ seems reasonable for the following reasons:

- In the classical (Oja's) case the objective function contains the neuron's output variance which depends upon the modulus (i.e. the unsigned value) of the output only.

- The leading principle in minor/principal component/subspace analysis is projection power minimization/maximization. In the signal processing literature, the power of a complex-valued signal is defined as the average of its square modulus. This quadratic optimization principle may be extended to robust (non-quadratic) optimization in a natural way, as shown in the present and following sections.

- Ample literature is available on signal processing techniques/algorithms which are based on transformed-output-modulus optimization principles. A recent review of such techniques in the blind signal processing area may be found, for instance, in [3, 20, 78, 88].

- As mentioned in the introduction, some algorithms described in the present contribution have readily found application to engineering-type signal-processing tasks (e.g. adaptive beamforming, blind separation of digital-modulation-type signals and blind localization of electromagnetic sources), which are based on transformed-output-modulus optimization.

The function $L(\cdot)$ is needed for embodying the necessary constraints of orthonormality of the weight-vectors in the criterion (4), namely:

$$\mathbf{w}_j^H \mathbf{w}_k = 0 \text{ if } j \neq k \ , \ \mathbf{w}_k^H \mathbf{w}_k = 1 \ . \tag{7}$$

Note that each orthogonality condition may be rewritten more conveniently by observing that $\mathbf{w}_j^H \mathbf{w}_k = 0$ if and only if $\mathrm{Re}\{\mathbf{w}_j^H \mathbf{w}_k\} = 0$ and $\mathrm{Im}\{\mathbf{w}_j^H \mathbf{w}_k\} = 0$, thus the function $L(\cdot)$ may be expressed as:

$$L(\mathbf{w}_k) \stackrel{\text{def}}{=} \lambda_{kk}(\mathbf{w}_k^H \mathbf{w}_k - 1) + \sum_{j=1, j\neq k}^{K(k)} \mathrm{Re}\{\lambda_{kj} \mathbf{w}_k^H \mathbf{w}_j\} \ . \tag{8}$$

A set of Lagrange multipliers $\{\lambda_{kj}\}$ has been introduced in order to take into account the condition ensuring the normality of the weight vectors, and the pairs of conditions needed for ensuring the orthogonality principle to be met after learning: In fact, the following identity holds:

$$\mathrm{Re}\{\lambda_{kj}(\mathbf{w}_k^H \mathbf{w}_j)\} = \mathrm{Re}\{\lambda_{kj}\}\mathrm{Re}\{\mathbf{w}_k^H \mathbf{w}_j\} - \mathrm{Im}\{\lambda_{kj}\}\mathrm{Im}\{\mathbf{w}_k^H \mathbf{w}_j\} \ ,$$

therefore the expression $\mathrm{Re}\{\lambda_{kj}(\mathbf{w}_k^H \mathbf{w}_j)\}$ is a compact way to express the equality constraints $\mathrm{Re}\{\mathbf{w}_j^H \mathbf{w}_k\} = 0$ and $\mathrm{Im}\{\mathbf{w}_j^H \mathbf{w}_k\} = 0$ by the Lagrange multiplier method (the sign $-$ between the above terms does not care, because the multipliers are variables to be determined). Note that the multipliers are complex-valued, except for the λ_{kk} that are real-valued, therefore the cost function $L(\cdot)$ is by construction real-valued.

The indexing function $K(\cdot)$ has been introduced in order to take into account two distinct cases:

- The *symmetric case*, where $K(k) = m$ for each neuron, that leads to a generalized principal subspace analyzer;
- The *hierarchic case*, where $K(k) = k$, actually leading to a generalized principal component analyzer.

In order to look for optimal weights $\mathbf{w}_k^{\text{opt}}$ maximizing the criterion (4), a gradient steepest ascent learning algorithm is employed, by following the general scheme (1)-(2). By definition, the gradient of a real-valued function $F(\mathbf{w})$ with respect to a complex-valued vector $\mathbf{w}$ is intended as:

$$\frac{\partial F(\mathbf{w})}{\partial \mathbf{w}} \stackrel{\text{def}}{=} \frac{\partial F(\mathbf{u},\mathbf{v})}{\partial \mathbf{u}} + i\frac{\partial F(\mathbf{u},\mathbf{v})}{\partial \mathbf{v}} \ , \tag{9}$$

where $i \stackrel{\text{def}}{=} \sqrt{-1}$ and $\mathbf{u} + i\mathbf{v} = \mathbf{w}$; also, $F(\mathbf{u},\mathbf{v}) : \mathbb{R}^p \times \mathbb{R}^p \rightarrow \mathbb{R}$ denotes the real-valued function of two real-valued arguments derived from $F(\mathbf{u} + i\mathbf{v})$.

First, it is necessary to evaluate the gradient:

$$\frac{\partial U(\mathbf{w}_k)}{\partial \mathbf{w}_k} = E\left[\frac{dg(|y_k|)}{d|y_k|}\frac{\partial |y_k|}{\partial \mathbf{w}_k}\right] = E\left[\frac{g'(|y_k|)}{|y_k|}y_k^\star \mathbf{x}\right] \ . \tag{10}$$

In fact, from definition (9) it follows that $|y_k|\frac{\partial |y_k|}{\partial \mathbf{w}_k} = y_k^\star \mathbf{x}$, where superscript * stands for complex conjugation. In the same way, the gradient of the function $L(\cdot)$ is to be evaluated. The expression of the gradient of $L(\mathbf{w}_k)$ with respect to $\mathbf{w}_k$ is found to be:

$$\frac{\partial L(\mathbf{w}_k)}{\partial \mathbf{w}_k} = 2\lambda_{kk}\mathbf{w}_k + \sum_{j=1, j\neq k}^{K(k)} \lambda_{kj}^\star \mathbf{w}_j \ . \tag{11}$$

By gathering equations (10) and (11) we thus obtain:

$$\frac{\partial J(\mathbf{w}_k)}{\partial \mathbf{w}_k} = E\left[\frac{g'(|y_k|)}{|y_k|}y_k^\star \mathbf{x}\right] + 2\lambda_{kk}\mathbf{w}_k + \sum_{j=1, j\neq k}^{K(k)} \lambda_{kj}^\star \mathbf{w}_j \ . \tag{12}$$

The standard elimination process may be performed in order to get rid of the dependence upon the Lagrange multipliers. It consists in solving the following set of equations for λ_{kj}, after having fixed e.g. the variable k:

$$\mathbf{w}_r^H \left(\frac{\partial J(\mathbf{w}_k)}{\partial \mathbf{w}_k}\right) = 0 \ , \ r = 1, \dots \ , m \ , \tag{13}$$

under the constraint of orthonormality of the vectors $\mathbf{w}_k$ expressed by equations (7). Tedious but straightforward calculations give, after footer renaming, the result:

$$\lambda_{kj}^{\text{opt}} = -E\left[\frac{g'(|y_k|)}{|y_k|}y_k y_j^\star\right]\left(1 - \frac{1}{2}\delta_{kj}\right) \ , \tag{14}$$

where δ_{kh} denotes the Kronecker's delta. It would be interesting to observe that the multipliers do not enjoy any symmetry properties, unless $g'(u) = u$: In this special case, in fact, $(\lambda_{kj}^{\text{opt}})^\star = \lambda_{jk}^{\text{opt}}$ would hold.

By plugging expressions (14) into equation (12), the formula for the optimal gradient of J is readily found:

$$\left(\frac{\partial J}{\partial \mathbf{w}_k}\right)^{\text{opt}} = E\left\{\frac{g'(|y_k|)}{|y_k|} y_k^\star \left[\mathbf{x} - \sum_{j=1}^{K(k)} y_j \mathbf{w}_j\right]\right\} \ . \tag{15}$$

Now the optimal gradient may be used in a steepest ascent algorithm to design a neural optimizing system. In short, by defining the quantities:

$$\mathbf{P}_k \stackrel{\text{def}}{=} \mathbf{I}_p - \sum_{j=1}^{K(k)} \mathbf{w}_j \mathbf{w}_j^H \text{ and } G(u) \stackrel{\text{def}}{=} \frac{dg(u)}{du}\frac{1}{u} \ , \tag{16}$$

with $u \in \mathbb{R}^+$, the new learning rule writes:

$$\frac{d\mathbf{w}_k}{dt} = \mathbf{P}_k E[G(|y_k|) y_k^\star \mathbf{x}] \ , \ k = 1, 2, \ldots, m \ . \tag{17}$$

The factor $E[G(|y_k|)y_k^\star \mathbf{x}]$ may be interpreted as a complex-valued extended Hebbian term common to each neuron, while the projector $\mathbf{P}_k$ is a deflating term which drives each weight-vector $\mathbf{w}_k$ into a different subspace[3].

It would be worth noting that by choosing as $g(\cdot)$ the function $g(u) = \frac{1}{2}u^2$, that means having $g'(u) = u$ and thus $G(u) = 1$, the gradient steepest ascent learning equations for a continuous-time PSA/PCA neural network with m outputs become:

$$\frac{d\mathbf{w}_k}{dt} = E\left[y_k^\star \left(\mathbf{x} - \sum_{j=1}^{K(k)} y_j \mathbf{w}_j\right)\right] \ , \ k = 1, 2, \ldots, m \ . \tag{18}$$

When $K(k) = k$, it coincides with the learning rule proposed by DeCastro, DeCastro, Amaral and Franco [23], that rewrites compactly as:

$$\frac{d}{dt}\mathbf{W} = E\left[\mathbf{x}\mathbf{x}^H - \text{UT}(\mathbf{W}^H \mathbf{x}\mathbf{x}^H \mathbf{W})\right]\mathbf{W} \ ,$$

where $\text{UT}(\cdot)$ returns the upper-triangular part of the matrix contained within. If, furthermore, $\mathbf{x} \in \mathbb{R}^p$, then the DeCastro-DeCastro-Amaral-Franco rule coincides with Sanger's learning paradigm [82].

In the generalized case, the function $g(\cdot)$ is assumed different from quadratic, usually by the help of robust statistics theory [53]. An alternative choice of this function will be discussed in section 2.3 .

[3]The term 'operator' is a little abused here: Unless the vectors $\mathbf{w}_k$ become orthonormal, the operator $\mathbf{P}_k$ does not represent a true projection because it is not idempotent, i.e. $\mathbf{P}_k^2 \neq \mathbf{P}_k$.

2.2 Output decorrelation by lateral inhibition

Kung and Diamantaras realized a Hebbian learning network with the Rubner-Tavan's laterally connected linear neural network [81], endowed with the unsupervised APEX learning rule [25]. The input/output network equations are:

$$\mathbf{y} = \mathbf{z} + \mathbf{H}^H \mathbf{y} \ , \ \mathbf{z} = \mathbf{W}^H \mathbf{x} \ , \tag{19}$$

where $\mathbf{x} \in \mathbb{C}^p$, $\mathbf{y}, \mathbf{z} \in \mathbb{C}^m$, $\mathbf{W} \in \mathbb{C}^{p \times m}$ and $\mathbf{H} \in \mathbb{C}^{m \times m}$. Note that $\mathbf{H}$ is strictly upper-triangular, thus the network is hierarchic (but not recurrent). It is in fact worth noting that the network output signal y_k may be expressed as:

$$y_k = \mathbf{w}_k^H \mathbf{x} + \mathbf{h}_k^H \mathbf{y} \ , \tag{20}$$

for $k = 1, \ldots, m$, with $\mathbf{w}_k \in \mathbb{C}^p$ and $\mathbf{h}_k \in \mathbb{C}^m$. An exemplary laterally-connected network of this kind is depicted in the Figure 1. As the network is hierarchic, the vectors $\mathbf{h}_k$ enjoy the property $h_{kj} = 0$ for $j \geq k$, thus, for instance, $\mathbf{h}_1 = [0 \ 0 \ \cdots \ 0 \ 0]^T$, $\mathbf{h}_2 = [h_{21} \ 0 \ \cdots \ 0 \ 0]^T$, $\mathbf{h}_3 = [h_{31} \ h_{32} \ 0 \ \cdots \ 0 \ 0]^T$ and so forth. This implies that the quantity $\mathbf{h}_k^H \mathbf{y}$ does not depend on $\mathbf{w}_k$, but only on $\mathbf{w}_j$ with $j < k$. Kung and Diamantaras proved that, under reasonable conditions, in the real-valued case the APEX rule make the column-vectors of the direct-connection weight matrix $\mathbf{W}$ asymptotically converge to the first principal eigenvectors of the covariance matrix of the input signal, and the lateral-connection weight-matrix $\mathbf{H}$ asymptotically vanishes to zero.

Later, Chen and Hou [17] extended the APEX algorithm to perform PCA of complex-valued random signals. They proved experimentally that the complex-valued APEX algorithm actually allows to extract a number of principal components from a complex-valued signal.

The aim of the present section is to give an alternative explanation of the complex-valued APEX learning rules, based on gradient optimization, and to extend this result to Hebbian learning. In order to design a PCA-type learning procedure for the considered laterally-connected network, let us define the following criterion function:

$$N_k(\mathbf{W}, \mathbf{H}) \stackrel{\text{def}}{=} E[f(\mathbf{w}_k^H \mathbf{x})] + \mathbf{h}_k^H E[\mathbf{y}\mathbf{y}^H] \mathbf{h}_k \ . \tag{21}$$

The first term in the right-hand side contains the generalized power of the transformed signal $z_k = \mathbf{w}_k^H \mathbf{x}$, where again $f(\zeta) = g(|\zeta|)$, $\zeta \in \mathbb{C}$, while the

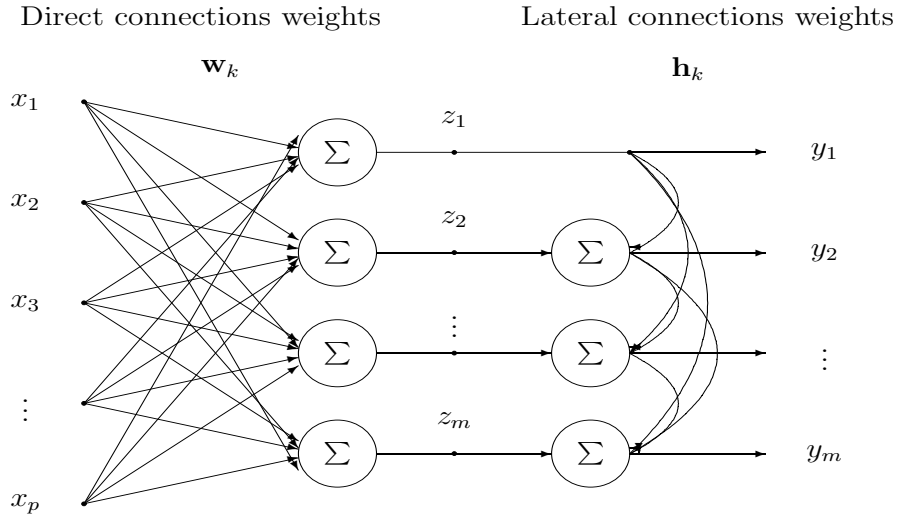

Figure 1: An exemplary network exhibiting laterally connections.

second term contains a linear combination of the cross-correlation values of the outputs. (By 'generalized power' it is meant the expectation of a non-quadratic function of network's output.) By definition of PCA, the first term has to be maximized under the constraint $\mathbf{w}_k^H \mathbf{w}_k = 1$, while the second term must be zeroed. Generalized power maximization and decorrelation of output signals may be thought of as separate objectives. These targets may be attained through the following objective function:

$$P(\mathbf{W}, \mathbf{H}) \stackrel{\text{def}}{=} \sum_{k=1}^{m} \left\{ N_k(\mathbf{W}, \mathbf{H}) + \lambda_k(\mathbf{w}_k^H \mathbf{w}_k - 1) + E[\psi_k](\mathbf{h}_k^H \mathbf{h}_k) \right\} \ , \qquad (22)$$

where the functions λ_k and $E[\psi_k]$ are Lagrange multipliers. The choice of denoting with $E[\psi_k]$ the Lagrange multipliers corresponding to the constraints on the lateral-connection strengths is motivated by the observation that, in the considered learning equations, the optimal values of multipliers appear as the statistical expectation of some functions, thus the adopted notation inherently embodies such features [27]. Also, this choice makes the notation easier.

It deserves to note that the terms $\sum_{k=1}^{m} E[\psi_k](\mathbf{h}_k^H \mathbf{h}_k)$ add degrees of freedom to the learning system: Such term ensures zero lateral forcing at convergence (i.e. $\mathbf{h}_k^H \mathbf{h}_k = 0$) and the presence of the free functions ψ_k affects the learning dynamics and can speed-up the convergence of the network [22, 27] without

changing the set of stationary points of the algorithm.

The direct-connection $\mathbf{W}$ adaptation aims at *maximizing* the power of the transformed signal by maximizing the objective function (22) with respect to $\mathbf{W}$ *only*. In order to adapt each $\mathbf{w}_k$, the gradient steepest ascent rule (1)-(2) may be used. By reasoning as in the previous section for multipliers computation, the optimal gradient of P with respect to $\mathbf{w}_k$ is found to be:

$$\left(\frac{\partial P}{\partial \mathbf{w}_k}\right)^{\text{opt}} = E[G(|y_k|)(\mathbf{x}y_k^\star - z_k y_k^\star \mathbf{w}_k)] \ , \ k = 1, 2, \ldots, m \ , \tag{23}$$

with $G(\cdot)$ being defined as in section 2.1.

Then, the lateral-connection weight-matrix $\mathbf{H}$ only is used in order to *minimize* the cost function defined in (22) in order to decorrelate the network's outputs. By defining the real part and the imaginary part of y_k and $\mathbf{h}_k$ as $y_k \stackrel{\text{def}}{=} y_k^{(R)} + iy_k^{(I)}$ and $\mathbf{h} \stackrel{\text{def}}{=} \mathbf{h}_k^{(R)} + i\mathbf{h}_k^{(I)}$, with some mathematical work we find:

$$\frac{\partial y_k^{(R)}}{\partial \mathbf{h}_k^{(R)}} = \mathbf{y}_{[k]}^{(R)} \ , \frac{\partial y_k^{(I)}}{\partial \mathbf{h}_k^{(R)}} = \mathbf{y}_{[k]}^{(I)} \ ,$$
$$\frac{\partial y_k^{(R)}}{\partial \mathbf{h}_k^{(I)}} = \mathbf{y}_{[k]}^{(R)} \ , \frac{\partial y_k^{(I)}}{\partial \mathbf{h}_k^{(I)}} = -\mathbf{y}_{[k]}^{(I)} \ ,$$

where $\mathbf{y}_{[k]}^{(R)} \stackrel{\text{def}}{=} [y_1^{(R)} \ y_2^{(R)} \ \cdots \ y_{k-1}^{(R)} \ 0 \ \cdots 0]^T$, $\mathbf{y}_{[k]}^{(I)} \stackrel{\text{def}}{=} [y_1^{(I)} \ y_2^{(I)} \ \cdots \ y_{k-1}^{(I)} \ 0 \ \cdots 0]^T$, with $k > 1$ and $\mathbf{y}_{[1]}^{(I)} = \mathbf{y}_{[1]}^{(R)} \stackrel{\text{def}}{=} [0 \ \cdots \ 0]^T$. Hence, we find:

$$\frac{\partial N_k}{\partial \mathbf{h}_k} = 2E\left[\mathbf{y}_{[k]} y_k^\star\right] \ , \tag{24}$$

where $\mathbf{y}_{[k]} \stackrel{\text{def}}{=} [y_1 \ y_2 \ \cdots \ y_{k-1} \ 0 \ \cdots \ 0]^T$ with $k > 1$ and $\mathbf{y}_{[1]} \stackrel{\text{def}}{=} [0 \, 0 \ \cdots \ 0 \, 0]^T$. The function P may be iteratively minimized, with respect to the variable matrix $\mathbf{H}$, by means of a gradient steepest descent rule, where the gradient of P with respect to $\mathbf{h}_k$ assumes the expression:

$$\frac{\partial P}{\partial \mathbf{h}_k} = 2E\left[\mathbf{y}_{[k]} y_k^\star + \psi_k \mathbf{h}_k\right] \ , \ k = 1, 2, \ldots, m \ .$$

It is interesting to note that in view of optimization, there are no theoretical reasons to force functions ψ_k to assume any particular value [22, 27]. In other terms, unless further constraints are established in the optimization of function (22), it is impossible to find optimal values for the multipliers $E[\psi_k]$.

On the basis of the previous calculations, the obtained non-linear complex-valued learning rules for output decorrelation by lateral inhibition reads:

$$\left\{ \begin{array}{rcl} \frac{d\mathbf{w}_k}{dt} & = & E[G(|y_k|)(\mathbf{x}y_k^\star - z_k y_k^\star \mathbf{w}_k)] \ , \ k = 1, 2, \ldots, m \ , \\ \frac{d\mathbf{h}_k}{dt} & = & -E\left[\mathbf{y}_{[k]} y_k^\star - \psi_k \mathbf{h}_k\right] \ , \ k = 1, 2, \ldots, m \ . \end{array} \right. \tag{25}$$

It could be interesting to discuss two possible choices of the functions $\psi(\cdot)$'s and to relate the obtained learning rules to learning paradigms known from the scientific literature:

- The first one consists in assuming null the free functions ψ_k. In order to find a correspondence in the scientific literature with a related learning rule, we consider the classical (quadratic optimization) case, in which the function $g(u)$ is assumed equal to $\frac{1}{2}u^2$ and real-valued signals are dealt with. Then, in the hypothesis that for t large enough the lateral forcing has vanished ($\mathbf{h}_k \approx \mathbf{0}$), because the neurons have learnt to encode uncorrelated features, from the first of equations (19) we see that $z_k \approx y_k$. Consequently, the learning rules (25) can be approximated by:

$$\left\{ \begin{array}{rcl} \frac{d\mathbf{w}_k}{dt} & = & E[y_k(\mathbf{x} - y_k \mathbf{w}_k)] \ , \ k = 1, 2, \ldots, m \ , \\ \frac{d\mathbf{h}_k}{dt} & = & -E[\mathbf{y}_{[k]} y_k] \ , \ k = 1, 2, \ldots, m \ , \end{array} \right. \tag{26}$$

  known as the Rubner-Tavan's learning equations [81]. It is easy to recognize a principal component rule for the direct connections and an anti-Hebbian rule for the lateral connections, in the spirit of Hebbian/anti-Hebbian learning recalled in the Introduction.

- The second choice consists in assuming $\psi_k = |y_k|^2$. Again in the quadratic optimization hypothesis and under the asymptotic assumption $z_k \approx y_k$, the model (25) closely resembles the learning algorithm by Chen-Hou [17] (which, in turn, coincides to the Kung-Diamantaras' learning rule in presence of real-valued signals):

$$\left\{ \begin{array}{rcl} \frac{d\mathbf{w}_k}{dt} & = & E[y_k^\star(\mathbf{x} - y_k \mathbf{w}_k)] \ , \ k = 1, 2, \ldots, m \ , \\ \frac{d\mathbf{h}_k}{dt} & = & -E[y_k^\star(\mathbf{y}_{[k]} + y_k \mathbf{h}_k)] \ , \ k = 1, 2, \ldots, m \ . \end{array} \right. \tag{27}$$

  It deserves to note that the learning equations for $\mathbf{w}_k$ in (25) differ from the corresponding Chen-Hou equations because of the presence of the term $z_k y_k^\star$ instead of $y_k y_k^\star = |y_k|^2$. This difference may be non-negligible: The

quantity $z_k y_k^\star$ is a complex number, thus it embodies a phase factor that $|y_k|^2$ does not possess. Such difference disappears when $m = 1$: In this case the mentioned equations coincide to the Oja's first principal component analyzer in the complex domain.

2.3 The Sudjianto-Hassoun principle and its extension to the complex-valued case

In the sections 2.1 and 2.2, non-quadratic criteria optimization was dealt with in order to define extended Hebbian learning rules. Here we aim at briefly recalling the Sudjianto-Hassoun interpretation of Hebbian learning and to extend this theory to the complex-valued case, because it naturally leads to a possible choice of non-quadratic criterion for learning. Also, an exemplary application to complex-valued independent component analysis will help us clarifying the usefulness of Sudjianto-Hassoun principle in extended Hebbian learning.

Sudjianto and Hassoun [87] considered the problem of maximizing a criterion $J(\mathbf{w}) \stackrel{\text{def}}{=} E[S^2(\mathbf{w}^T\mathbf{x})]$ subject to the restriction $\mathbf{w}^T\mathbf{w} = 1$, where $y = \mathbf{w}^T\mathbf{x}$ is the output of a single-unit neural network and $S(\cdot)$ is a generic saturating sigmoidal function, for instance such that $S(y) \in [-1, +1]$. They noted that maximizing the variance of a saturating function of y leads the model neuron to prefer configurations $\mathbf{w}$ that correspond to having the values of $S(y)$ concentrated around the extremes -1 and $+1$. If the quantity $v \stackrel{\text{def}}{=} S(y)$ is perceived as a new random variable with probability density function $q_V(v|\mathbf{w})$, $q_V(v)$ in short notation, this corresponds to having this distribution U–shaped [87]. The gradient steepest ascent learning rule for this abstract neuronal system is:

$$\frac{d\mathbf{w}}{dt} = (\mathbf{I} - \mathbf{w}\mathbf{w}^T)E[\ell(y)\mathbf{x}] \ , \tag{28}$$

where $\ell(u) \stackrel{\text{def}}{=} 2S'(u)S(u)$. Let us denote now by $q_Y(y|\bar{\mathbf{w}})$ the probability density function of the random variable y due to a configuration $\bar{\mathbf{w}}$ and with $Q_Y(y|\bar{\mathbf{w}})$ its cumulative distribution function, namely:

$$Q_Y(y|\bar{\mathbf{w}}) \stackrel{\text{def}}{=} \int_{-\infty}^{y} q_Y(u|\bar{\mathbf{w}})du \ .$$

Let us also assume $\bar{S}(y) = 2Q_Y(y|\bar{\mathbf{w}}) - 1$. In this case it is well known [87] that the variable v will be uniformly distributed within $[-1, +1]$. The central idea

developed by Sudjianto and Hassoun is that the learning rule (28) will converge to a weight vector surely different from $\bar{\mathbf{w}}$, since the rule seeks a U–shaped distribution of v, that is, a distribution that deviates away from a uniform one.

In order to extend the Sudjianto-Hassoun principle to the complex-valued case, let us consider the cost function:

$$U(\mathbf{w}) \stackrel{\text{def}}{=} E[S^2(|y|)] \ , \tag{29}$$

for a complex-weighted neuron with output $y = \mathbf{w}^H\mathbf{x}$, with $\mathbf{x}$ and $\mathbf{w}$ belonging to $\mathbb{C}^p$. Its gradient steepest ascent maximization under the constraint $\mathbf{w}^H\mathbf{w} = 1$ yields the learning rule:

$$\frac{d\mathbf{w}}{dt} = (\mathbf{I} - \mathbf{w}\mathbf{w}^H)E\left[\ell(|y|)\frac{y^\star}{|y|}\mathbf{x}\right] \ , \tag{30}$$

that closely recalls equation (17) for $m = 1$. The learning rule (30) drives function $S(|y|)$ to take on values around its extremes, for instance $+1$, therefore it seeks the configurations $\mathbf{w}$ making the probability density function $q_V(v)$ different from a uniform one. Now we may consider for $S(|y|)$ a function like:

$$Q_{|Y|}(|y|) \stackrel{\text{def}}{=} \int_0^{|y|} q_{|Y|}(u)du \ .$$

By equating (30) to (17), with $m = 1$, the relationship between $q_{|Y|}$ and g', finds to be:

$$g'(|y|) = 2q_{|Y|}(|y|)\int_0^{|y|} q_{|Y|}(u)du \ . \tag{31}$$

The concept of Sudjianto-Hassoun interpretation of Hebbian learning has been just touched here: An extensive analysis of this interesting and fruitful theory has been recently proposed in [34] in the context of complex-valued independent component analysis.

In summary, independent component analysis allows extracting statistically independent source signals from their linear combinations where the mixing operator is unknown and the temporal dynamic of the source signals is also unknown. Another useful interpretation of ICA techniques concerns their feature extraction ability, which offers a mathematically sounding way of decomposing signals which exhibit involved dynamics into independent basis signals [21, 51].

Formally, the simpler source-observation model is written as $\mathbf{x} = \mathbf{A}\mathbf{s}$, where $\mathbf{s} \in \mathbb{C}^p$ denotes the vector of source signals, $\mathbf{x} \in \mathbb{C}^p$ denotes the vector of observed signals and $\mathbf{A} \in \mathbb{C}^{p\times p}$ denotes the matrix of mixing coefficients. In the

above (noiseless, instantaneous) case, the matrix $\mathbf{A}$ may be assumed orthonormal without loss of generality. In fact, it is known that if $\mathbf{A}$ is not orthonormal, it is possible to pre-process the observed data via a so-termed whitening algorithm that makes it possible to separate out the independent components via a rotation only [21, 51]. In this case, a linear artificial neural network described by $\mathbf{y} = \mathbf{W}^H\mathbf{x}$ with $\mathbf{W} \in \mathbb{C}^{p\times p}$ orthonormal may be effective in separating the independent source signals out from the observed mixtures.

In the case that the sources are typical base-band digital signals used in telecommunications, such as QAM or PSK (QAM stands for *Quadrature Amplitude Modulation* and PSK stand for *Phase Shift Keying* [46]), we may give a quite appealing interpretation of the way the Sudjianto-Hassoun principle works. In fact, in the above mentioned cases, the probability density functions of the generic source signal s_k modulus writes:

$$q_{|s_k|}(|s_k|) = \sum_r q_{|s_k|,r}\delta(|s_k| - \bar{s}_{k,r}) \ , \tag{32}$$

where the constant $\bar{s}_{k,r}$ denotes the r:th possible discrete-value assumed by the quantity $|s_k|$, the symbol $\delta(\cdot)$ denotes the Dirac's delta and the quantities $q_{|s_k|,r}$ denotes the probability that $|s_k| = \bar{s}_{k,r}$. Now, the probability density function of each observation is different from peaked (i.e. more similar to a uniform one), so the aim of the separating network is to make the outputs distributions of the artificial neural network as peaked as possible. This behavior is readily recognized to extend the 'make as U-shaped as possible' the output modulus distribution with 'make as peaked as possible' the same distribution.

These informal considerations offer a further justification for the choice of learning rules based on the optimization of criterion functions that depend on the network output moduli only. The above informal considerations have been given formal investigation in [15, 28, 34].

Some contributions on the use of non-classical principal component techniques to blind source separation have been recently summarized in [34] and the choice of the involved non-linear functions have been discussed for instance in [50, 64]. The main weakness points of these approaches are:

- The non-linear functions are generally chosen on the basis of existing contributions from robust statistics and on heuristic observations/findings;

- Their choice generally rely on the fact that using non-linear functions add to the system high-order statistical features but give little insight into how these features are related to the separation problem;

- Because of the strong non-linearity of the learning equations, it is difficult to give any analytical proof of convergence nor detailed studies about the features of the employed learning criteria.

In order to illustrate the usefulness of Sudjianto-Hassoun principle in extended Hebbian learning for independent component analysis and to clarify the above calculi with an example, let us suppose that input $\mathbf{x}$ contains a complex-valued orthogonal mixture of statistically independent signals [24, 58] and that one of these signals is a Gaussian noise of the form $\nu = \nu^{(R)} + i\nu^{(I)}$, where $\nu^{(R)}$ and $\nu^{(I)}$ are zero-mean Gaussian random variables with variance σ^2. Then it is known that the modulus $|\nu|$ follows the Rayleigh distribution:

$$q_{\mathcal{R}}(u) = \frac{u}{\sigma^2} \exp\left(-\frac{u^2}{2\sigma^2}\right) \Gamma(u) \ ,$$

where $\Gamma(u)$ is the unit-step function. Then by formula (31) we find:

$$g'_{\mathcal{R}}(u) = \frac{2u}{\sigma^2} \left[\exp\left(-\frac{u^2}{2\sigma^2}\right) - \exp\left(-\frac{u^2}{\sigma^2}\right)\right] \ , \ u \geq 0 \ .$$

Figure 2 depicts the Rayleigh warping function $\frac{g'(u)}{u}$ for a unit-power noise. In this case it is possible to express the cumulative distribution function as:

$$Q_{\mathcal{R}}(u) = 1 - \exp\left(-\frac{u^2}{2\sigma^2}\right) \ ,$$

whose shape is illustrated in Figure 2. It is interesting to note that the obtained shape for the neuron model's non-linear function $Q_{\mathcal{R}}(u)$ formally differs from the classical hyperbolic-tangent one, even if its shape resembles the usual saturating sigmoid. Also, in this case, the sigmoid's shape-parameter σ has a clear physical meaning.

In order to illustrate how the above theory works in practice, we may consider a simple 4×4 complex-valued independent component analysis case. The Figure 3 shows the source signals, s_1, s_2, s_3 and s_4, namely, 3 QAM16 signals and a complex Gaussian noise of standard deviation $\sigma = 0.5$ and their four linear superpositions x_1, x_2, x_3 and x_4. The same Figure also shows the histograms of

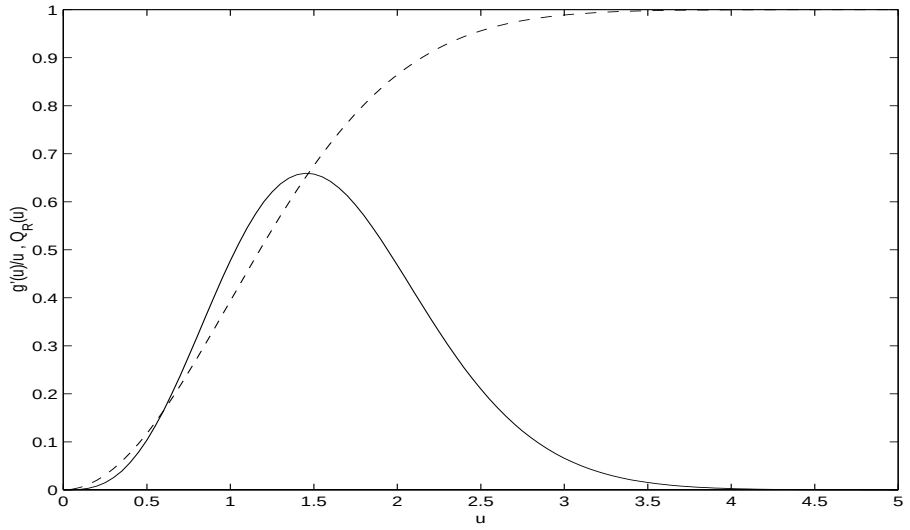

Figure 2: Rayleigh warping function $\frac{g'(u)}{u}$ (solid) and the corresponding cumulative function $Q_{\mathcal{R}}(u)$ (dashed) for $\sigma = 1$.

the moduli $|s_k|$ and $|x_k|$. It is immediate to see that, as the QAM16 symbols are discrete by definition, their histograms are peaked. This simple rule does not apply, of course, to the Gaussian noise. Conversely, the histograms of the observations x_k are distributed. The Figure 4 shows the whitened signals, denoted here by v_1, v_2, v_3 and v_4 and the estimated source signals obtained through the neural learning algorithm obtained by the extended Hebbian learning rule described in this section, denote by y_1, y_2, y_3 and y_4. The histograms of the moduli of the complex-valued signals y_1, y_2, y_3 look close to peaks, as predicted by the theory.

The main contribution of the Sudjianto-Hassoun-principle-based learning theory to ICA is to allow designing the proper structure of non-linear part of neurons *provided that some information is known on the statistical structure of signals that the source signals differ from.* This principle may prove very useful in blind signal processing, where the statistical structure of the involved signals is not known in advance, but it might be easily known the structure of signals different from the useful ones, such as noises.

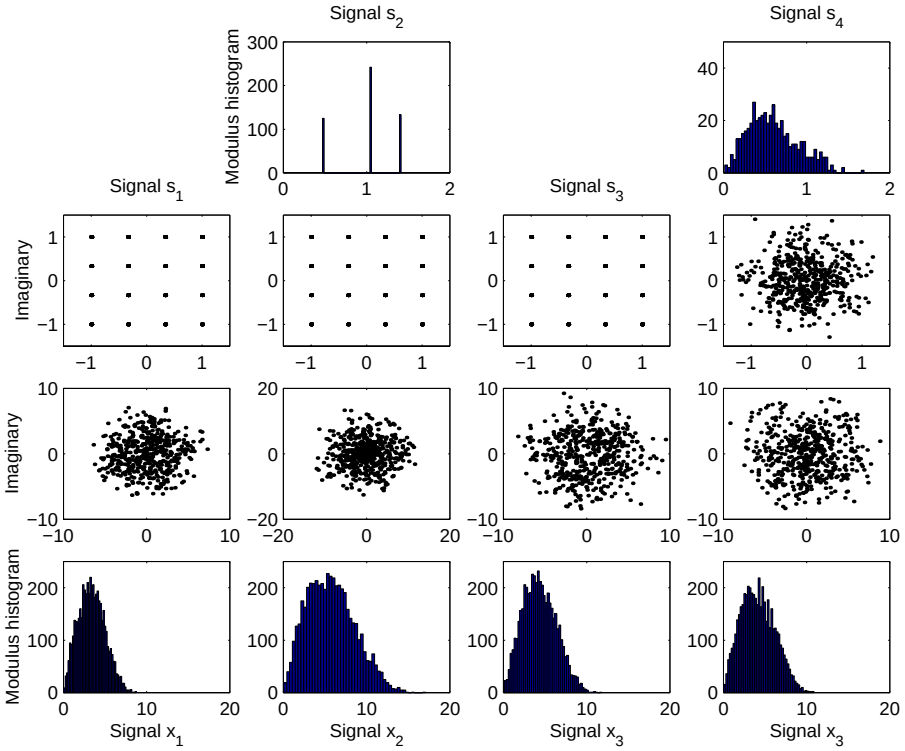

Figure 3: Second row: Source signals s_1, s_2, s_3 and s_4, (3 QAM16 signals and a complex Gaussian noise). Third row: Their four linear superpositions x_1, x_2, x_3 and x_4. First row: Histograms of the moduli $|s_k|$. Fourth row: Histograms of the moduli $|x_k|$.

2.4 Analysis of the one-unit complex-valued PCA/MCA

The learning rule (17) for a single neuron in the quadratic complex-valued case particularizes to the adapting law:

$$\frac{d\mathbf{w}}{dt} = E[y^\star \mathbf{x} - |y|^2 \mathbf{w}] \ , \tag{33}$$

with $\mathbf{x}$ and $\mathbf{w}$ belonging to $\mathbb{C}^p$. By defining the Hermitian covariance matrix $\mathbf{\Phi} \stackrel{\text{def}}{=} E[\mathbf{x}\mathbf{x}^H]$, this differential equation rewrites:

$$\frac{d\mathbf{w}}{dt} = \mathbf{\Phi}\mathbf{w} - (\mathbf{w}^H \mathbf{\Phi} \mathbf{w})\mathbf{w} \ . \tag{34}$$

This system has a set of stationary points defined by:

$$\mathcal{E}_* \stackrel{\text{def}}{=} \{\mathbf{w} \in \mathbb{C}^p : \mathbf{\Phi}\mathbf{w} - \sigma\mathbf{w} = \mathbf{0}\} \ . \tag{35}$$

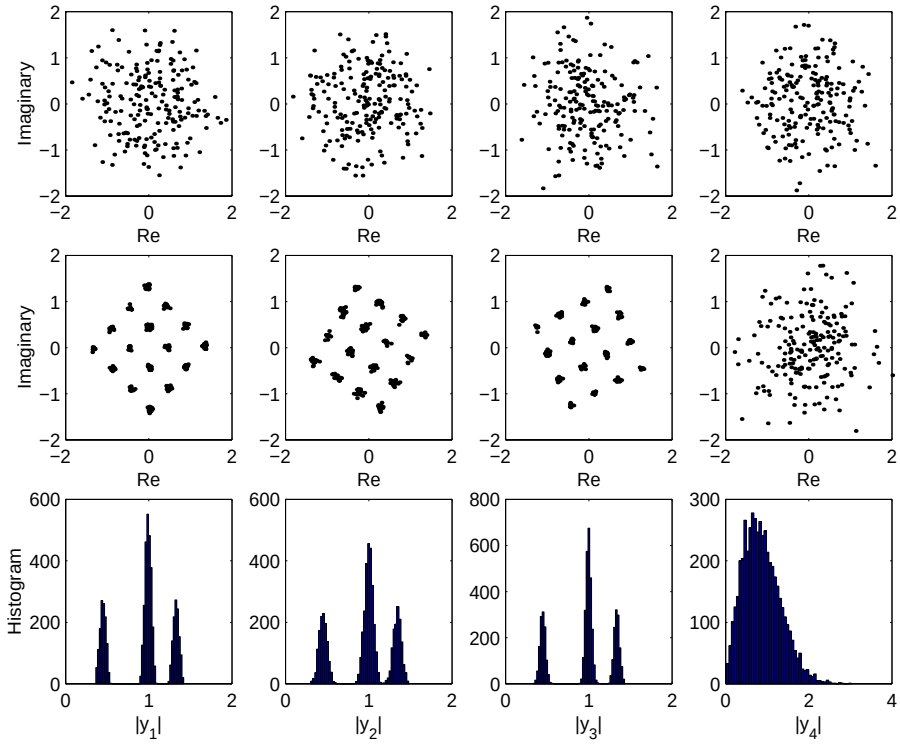

Figure 4: First row: Whitened observations v_1, v_2, v_3 and v_4. Second row: Estimated source signals y_1, y_2, y_3 and y_4 obtained through the extended Hebbian learning rule enriched by Sudjianto-Hassoun principle. Third row: Histograms of the moduli $|y_k|$.

With the exception of the trivial solution $\mathbf{w} = \mathbf{0}$, the elements $\mathbf{w} \in \mathcal{E}_*$ coincide to the eigenvectors of $\mathbf{\Phi}$.

The following proposition states the convergence of the system (34) to the eigenvector in $\mathcal{E}_*$ corresponding to the largest eigenvalue σ. The proof of this Theorem follows from the arguments successfully used in the real-valued case by other Authors [73, 82].

Theorem 1 *Let us suppose* $\mathbf{\Phi}$ *Hermitian in (34) with eigenpairs* $(\sigma_1, \mathbf{q}_1)$, $(\sigma_2, \mathbf{q}_2)$, $\ldots$, $(\sigma_p, \mathbf{q}_p)$. *Suppose then eigenvalues are distinct and arranged in descending order, eigenvectors are normalized so that* $\mathbf{q}_k^H \mathbf{q}_k = 1$ *and* $\mathbf{w}(0)^H \mathbf{q}_1 \neq 0$. *Then there holds:*

$$\lim_{t \to +\infty} \mathbf{w}(t) = \mathbf{q}_1 e^{i\alpha} \ ,$$

where α is arbitrary in $[0, 2\pi]$. □

Proof. Let us expand vector $\mathbf{w}(t)$ by means of the system's eigenbasis [45, 82], that means writing:

$$\mathbf{w}(t) = \theta_1(t)\mathbf{q}_1 + \theta_2(t)\mathbf{q}_2 + \cdots + \theta_p(t)\mathbf{q}_p \ , \tag{36}$$

where scalar functions $\theta_k(t) \in \mathbb{C}$ are termed "principal modes". Plugging equation (36) in equation (34) yields:

$$\sum_{h=1}^{p} \frac{d\theta_h(t)}{dt}\mathbf{q}_h = \sum_{h=1}^{p} \theta_h(t)\mathbf{\Phi}\mathbf{q}_h - \left\{\sum_{k=1}^{p}[\theta_k(t)\mathbf{q}_k]^H \mathbf{\Phi} \sum_{k=1}^{p}[\theta_k(t)\mathbf{q}_k]\right\} \sum_{h=1}^{p} \theta_h(t)\mathbf{q}_h \ .$$

By recalling property $\mathbf{\Phi}\mathbf{q}_k = \sigma_k\mathbf{q}_k$ we have:

$$\sum_{h=1}^{p} \frac{d\theta_h(t)}{dt}\mathbf{q}_h = \sum_{h=1}^{p} \theta_h(t)\mathbf{q}_h\sigma_h - \left[\sum_{k=1}^{p}\sum_{r=1}^{m} \theta_k(t)^\star\theta_r(t)\sigma_r\mathbf{q}_k^H\mathbf{q}_r\right] \sum_{h=1}^{p} \theta_h(t)\mathbf{q}_h \ .$$

Now from the fact that $\mathbf{q}_k^H\mathbf{q}_\ell = \delta_{k\ell}$, it follows that the differential equations for the h^{th} principal modes, with $h \geq 2$, read:

$$\dot{\theta}_h(t) = \theta_h(t)\sigma_h - \sum_{k=1}^{p} |\theta_k(t)|^2\sigma_k\theta_h(t) \ , \ h = 2, \ldots, p \ , \tag{37}$$

while it is particularly useful to write on a separate equation the differential law pertaining to the first principal mode $\theta_1(t)$, that is:

$$\dot{\theta}_1(t) = \theta_1(t)\sigma_1 - |\theta_1(t)|^2\theta_1(t)\sigma_1 - \sum_{k=2}^{p} |\theta_k(t)|^2\sigma_k\theta_1(t) \ . \tag{38}$$

Principal modes $\theta_h(t)$ are complex-valued quantities, and their real and imaginary parts are denoted here as $\theta_h^{(R)}(t)$ and $\theta_h^{(I)}(t)$, respectively. Our aim is now to solve differential system (37)+(38) that is a coupled non-linear system of differential equations. In order to decouple the non-linear differential sub-system (37), let us define the new functions:

$$\phi_h^{(R)}(t) \stackrel{\text{def}}{=} \frac{\theta_h^{(R)}(t)}{\theta_1^{(R)}(t)} \ , \ \phi_h^{(I)}(t) \stackrel{\text{def}}{=} \frac{\theta_h^{(I)}(t)}{\theta_1^{(I)}(t)} \ , \ h = 2, \ldots, p \ .$$

For the real part of the h:th principal mode there holds:

$$\frac{d}{dt}\phi_h^{(R)}(t) = \frac{\dot{\theta}_h^{(R)}(t)\theta_1^{(R)}(t) - \theta_h^{(R)}(t)\dot{\theta}_1^{(R)}(t)}{\theta_1^{(R)}(t)^2} \ , \tag{39}$$

and the system (37)+(38), particularized for the real parts, reads:

$$\dot{\theta}_h^{(R)}(t) = \theta_h^{(R)}(t)\sigma_h - \sum_{k=1}^{p} |\theta_k(t)|^2 \sigma_k \theta_h^{(R)}(t) \ , \ h = 2, \ldots, p \ . \tag{40}$$

$$\dot{\theta}_1^{(R)}(t) = \theta_1^{(R)}(t)\sigma_1 - \sum_{k=1}^{p} |\theta_k(t)|^2 \sigma_k \theta_1^{(R)}(t) \ . \tag{41}$$

By using equations (40) and (41) within equation (39), direct calculations show that the dynamics of the state-variables $\phi_h^{(R)}(t)$ looks:

$$\dot{\phi}_h^{(R)}(t) = (\sigma_h - \sigma_1)\phi_h^{(R)}(t) \ , \ h = 2, \ldots, p \ .$$

A similar formula holds for $\phi_h^{(I)}(t)$. Thus we have proven that there hold:

$$\theta_h^{(R)}(t) = \phi_h^{(R)}(0) e^{(\sigma_h - \sigma_1)t} \theta_1^{(R)}(t) \ , \tag{42}$$

$$\theta_h^{(I)}(t) = \phi_h^{(I)}(0) e^{(\sigma_h - \sigma_1)t} \theta_1^{(I)}(t) \ . \tag{43}$$

The above formulas show that the sub-system (37) has been successfully decoupled, since the dynamics of each principal mode no longer depends upon the other modes, apart from the first one. It is useful to note that from equations (42) and (43) the expression below follows:

$$|\theta_k(t)|^2 = \theta_k^{(R)}(t)^2 + \theta_k^{(I)}(t)^2 = e^{2(\sigma_k - \sigma_1)t} [\phi_k^{(R)}(0)^2 \theta_k^{(R)}(t)^2 + \phi_k^{(I)}(0)^2 \theta_k^{(I)}(t)^2] \ .$$

Now the aim is to solve differential equation (38) for $\theta_1(t)$. It is worth noting that it slightly simplifies if the following variable change is performed:

$$\theta_1(t) = c(t) e^{\sigma_1 t} \ , \ c(t) \in \mathbb{C} \ .$$

After this change we have:

$$|\theta_1(t)|^2 = |c(t)|^2 e^{2\sigma_1 t} \ ,$$

$$|\theta_h(t)|^2 = e^{2\sigma_k t} [\phi_h^{(R)}(0)^2 c^{(R)}(t)^2 + \phi_h^{(I)}(0)^2 c^{(I)}(t)^2] \ ,$$

where $c^{(R)}(t)$ and $c^{(I)}(t)$ stand, respectively, for the real part and the imaginary part of the function $c(t)$. Then, the differential equation (38) becomes:

$$\dot{c}(t) = -|c(t)|^2 e^{2\sigma_1 t} \sigma_1 c(t) - \sum_{k=2}^{p} e^{2\sigma_k t} [\phi_k^{(R)}(0)^2 c^{(R)}(t)^2 + \phi_k^{(I)}(0)^2 c^{(I)}(t)^2] \sigma_k c(t) \ . \tag{44}$$

By defining the auxiliary quantities:

$$\eta^{(R)}(t) \stackrel{\text{def}}{=} -\sum_{k=2}^{p} e^{2\sigma_k t}\sigma_k \phi_k^{(R)}(0)^2 - \sigma_1 e^{2\sigma_1 t} \ , \tag{45}$$

$$\eta^{(I)}(t) \stackrel{\text{def}}{=} -\sum_{k=2}^{p} e^{2\sigma_k t}\sigma_k \phi_k^{(I)}(0)^2 - \sigma_1 e^{2\sigma_1 t} \ , \tag{46}$$

the differential equations for the real and the imaginary parts of $c(t)$ rewrite, compactly:

$$\begin{cases} \dot{c}^{(R)}(t) = \eta^{(R)}(t)c^{(R)}(t)^3 + \eta^{(I)}(t)c^{(I)}(t)^2 c^{(R)}(t) \ , \\ \dot{c}^{(I)}(t) = \eta^{(I)}(t)c^{(I)}(t)^3 + \eta^{(R)}(t)c^{(I)}(t)c^{(R)}(t)^2 \ . \end{cases} \tag{47}$$

Since $\eta^{(R)}(t) < 0$, $\eta^{(I)}(t) < 0$ and at least $c(0) = \mathbf{w}(0)^H \mathbf{q}_1 \neq 0$, two cases should be considered.

CASE I: Only one between $c^{(R)}(0)$ and $c^{(I)}(0)$ differ from zero.

In this case, system (47) simplifies into $\dot{c}(t) = \eta(t)c(t)^3$, where $\eta(t)$ stands for $\eta^{(R)}(t)$ or $\eta^{(I)}(t)$. The above differential equation may be solved by:

$$\int_{c(0)}^{c(t)} \frac{dc}{c^3} = \int_0^t \eta(\tau)d\tau \ \Rightarrow \ \frac{1}{c(t)^2} = \frac{1}{c(0)^2} - 2\int_0^t \eta(\tau)d\tau \ .$$

This readily leads to:

$$\frac{1}{\theta_1(t)^2} = \frac{e^{-2\sigma_1 t}}{\theta_1(0)^2} - 2e^{-2\sigma_1 t}\int_0^t \eta(\tau)d\tau \ .$$

From definitions (45) and (46) it can be seen that under condition $\sigma_1 \notin \{\sigma_2, \ldots, \sigma_m\}$, there holds:

$$\lim_{t\to+\infty} 2e^{-2\sigma_1 t}\int_0^t \eta(\tau)d\tau = -1 \ . \tag{48}$$

This implies the conclusion:

$$\lim_{t\to+\infty} \frac{1}{\theta_1(t)^2} = 1 \ . \tag{49}$$

CASE II: $c^{(R)}(0) \neq 0$ and $c^{(I)}(0) \neq 0$.

The hypotheses imply $c^{(R)}(t) \neq 0$ and $c^{(I)}(t) \neq 0$, thus it is possible to multiply

both sides of equations (47) by a factor $2/c^{(R)}(t)$ and $2/c^{(I)}(t)$, respectively. This way we have:

$$\frac{d\log c^{(R)}(t)^2}{dt} = \frac{d\log c^{(I)}(t)^2}{dt} = 2[\eta^{(R)}(t)c^{(R)}(t)^2 + \eta^{(I)}(t)c^{(I)}(t)^2] \ . \tag{50}$$

The first equality in the chain tells that $\log c^{(R)}(t)^2 = \log c^{(I)}(t)^2 - \log\kappa$, where κ is a positive constant determined by the initial conditions. Equivalently, we have $c^{(I)}(t)^2 = \kappa c^{(R)}(t)^2$. This means that the equation (50) can be easily solved, in that there holds:

$$\frac{d\log c^{(R)}(t)^2}{dt} = 2[\eta^{(R)}(t) + \kappa\eta^{(I)}(t)]c^{(R)}(t)^2 \ .$$

The latter equation is equivalent to $\dot{c}^{(R)}(t) = [\eta^{(R)}(t) + \kappa\eta^{(I)}(t)]c(t)^{(R)}(t)^3$. Again we have:

$$\frac{1}{\theta_1^{(R)}(t)^2} = \frac{e^{-2\sigma_1 t}}{\theta_1^{(R)}(0)^2} - 2e^{-2\sigma_1 t}\int_0^t [\eta^{(R)}(\tau) + \kappa\eta^{(I)}(\tau)]d\tau \ , \tag{51}$$

$$\frac{1}{|\theta_1(t)|^2} = \frac{1}{1+\kappa}\frac{1}{\theta_1^{(R)}(t)^2} \ , \ \kappa = \frac{\theta_1^{(I)}(0)^2}{\theta_1^{(R)}(0)^2} \ . \tag{52}$$

By using twice the result (48), it may be shown that:

$$\lim_{t\to+\infty} 2e^{-2\sigma_1 t}\int_0^t [\eta^{(R)}(\tau) + \kappa\eta^{(I)}(\tau)]d\tau = -(1+\kappa) \ ,$$

thus we arrive at the conclusion:

$$\lim_{t\to+\infty} \frac{1}{|\theta_1(t)|^2} = 1 \ . \tag{53}$$

The results (49) and (53) also imply:

$$\lim_{t\to+\infty} |\theta_k(t)| = 0 \ , \ k = 2, 3, \ldots, p \ .$$

This completes the proof. □

It is worth mentioning that, while reasonable e.g. in signal processing tasks, the assumption that all eigenvalues are distinct is not crucial. The crucial assumption is that the first eigenvalue differs from the others, as suggested e.g. by the fact that the result (49) holds under condition $\sigma_1 \notin \{\sigma_2, \ldots, \sigma_m\}$. In this case, a proof of a similar consistency result may be based on the convergence proof for the real-valued case provided in [48, Theorem 3.3].

When the function $g(\cdot)$ defined in (6) is not quadratic, likely the neural unit just studied behaves in a different way than the Oja's neuron. A study on what happens in the general non-quadratic case for a real-weighted neuron has been carried out by Oja [76]. Such analysis is here extended to the complex case.

To start with, let us define the following quantities:

$$\mathcal{L}(\mathbf{w}) \stackrel{\text{def}}{=} E\left[\frac{g'(|y|)}{|y|} y^{\star}\mathbf{x}\right] = E\left[G(|\mathbf{w}^H\mathbf{x}|)(\mathbf{x}\mathbf{x}^H)\right]\mathbf{w} \ , \tag{54}$$

$$\mathcal{G}(\mathbf{w}) \stackrel{\text{def}}{=} E[g'(|y|)|y|] = E[G(|\mathbf{w}^H\mathbf{x}|)|\mathbf{w}^H\mathbf{x}|^2] \ , \tag{55}$$

with $G(u)$ defined again as in section 2.1. Note that $\mathcal{L}(\mathbf{w}) \in \mathbb{C}^p$, while $\mathcal{G}(\mathbf{w}) \in \mathbb{C}$. Then, the learning rule (17) rewrites:

$$\frac{d\mathbf{w}}{dt} = \mathcal{L}(\mathbf{w}) - \mathbf{w}\mathcal{G}(\mathbf{w}) \ , \tag{56}$$

as in [76]. The stationary points of system (56) are among the elements of the set:

$$\mathcal{W}_* \stackrel{\text{def}}{=} \{\mathbf{w} \in \mathbb{C}^p : \mathcal{L}(\mathbf{w}) - \mathbf{w}\mathcal{G}(\mathbf{w}) = \mathbf{0}\} \ . \tag{57}$$

Let us suppose $g'(u)$ has null even derivatives at the origin, as in [76], thus the MacLaurin expansion of $G(u)$ looks:

$$G(u) = g^{(2)}(0) + \frac{1}{6}u^2 g^{(4)}(0) + \text{higher order terms} \ .$$

Replacing the expanded function $G(u)$ into definitions (54) and (55) gives:

$$\begin{aligned}
\mathcal{L}(\mathbf{w}) &= \mathbf{\Phi}\mathbf{w}g^{(2)}(0) + \frac{1}{6}\mathbf{w}^H E[(\mathbf{x}\mathbf{x}^H)\mathbf{w}(\mathbf{x}\mathbf{x}^H)]\mathbf{w}g^{(4)}(0) + \text{h.o.t.} \ , \\
\mathcal{G}(\mathbf{w}) &= \mathbf{w}^H\mathbf{\Phi}\mathbf{w}g^{(2)}(0) + \frac{1}{6}E[|\mathbf{w}^H\mathbf{x}|^4]g^{(4)}(0) + \text{h.o.t.} \ ,
\end{aligned}$$

where again $\mathbf{\Phi}$ denotes the network input covariance matrix.

Therefore a neuron model with a sigmoidal activation function $G(\cdot)$ instead of a linear one introduces higher order statistics in learning equations. This can be seen by looking at the set of stationary points $\mathcal{W}_*$ that only in a first-order approximation coincides with $\mathcal{E}_*$, provided that the $|g^{(n)}(0)|$ are small enough for $n > 2$, therefore, in general the vectors in $\mathcal{W}_*$ deviate from the principal directions in $\mathcal{E}_*$ and depend thus from high-order statistical structures than covariance.

A widely known problem in the scientific literature is that extending the principal component analysis theory to a minor component analysis theory

is not a straightforward task. This is true even in the complex-domain case, particularly, it is not possible to replace maximization of the criterion (4) $J(\mathbf{w}) = U(\mathbf{w}) + L(\mathbf{w})$ with respect to $\mathbf{w}$, with its minimization, in that the resulting learning rule:

$$\frac{d\mathbf{w}}{dt} = -\frac{\partial J}{\partial \mathbf{w}} = E[-y^\star \mathbf{x} + |y|^2 \mathbf{w}] \ , \tag{58}$$

is unstable. The proof of the following proposition follows from the proof of the corresponding real-valued case presented e.g. in [33, 75].

Theorem 2 *Let $\mathbf{\Phi} \in \mathbb{C}^{p \times p}$ be a Hermitian matrix with eigenpairs $(\sigma_1, \mathbf{q}_1)$, $(\sigma_2, \mathbf{q}_2)$, ..., $(\sigma_p, \mathbf{q}_p)$. Let us suppose eigenvalues are distinct and arranged in descending order, and eigenvectors are normalized so that $\mathbf{q}_k^H \mathbf{q}_k = 1$. Then system $\dot{\mathbf{w}} = -\mathbf{\Phi}\mathbf{w} + (\mathbf{w}^H \mathbf{\Phi} \mathbf{w})\mathbf{w}$ with $\mathbf{w}(0)^H \mathbf{q}_1 \neq 0$ becomes unstable in finite time.* □

Proof. The proof essentially follows that of Theorem 1. Let us consider the principal-modes dynamics:

$$\begin{aligned}
\dot{\theta}_h(t) &= -\theta_h(t)\sigma_h + \sum_{k=1}^{p} |\theta_k(t)|^2 \sigma_k \theta_h(t) \ , \ h = 1, \ldots, p-1 \ , \\
\dot{\theta}_p(t) &= -\theta_p(t)\sigma_p + \sum_{k=1}^{p-1} |\theta_k(t)|^2 \sigma_k \theta_p(t) + |\theta_p(t)|^2 \theta_p(t)\sigma_p \ .
\end{aligned}$$

Let us now define the pseudo-modes:

$$\phi_h^{(R)}(t) \stackrel{\text{def}}{=} \frac{\theta_h^{(R)}(t)}{\theta_p^{(R)}(t)} \ , \ \phi_h^{(I)}(t) \stackrel{\text{def}}{=} \frac{\theta_h^{(I)}(t)}{\theta_p^{(I)}(t)} \ , \ h = 1, \ldots, p-1 \ ,$$

where superscript $^{(R)}$ and $^{(I)}$ denote, as usual, the real part and the imaginary part of a complex-valued variable. By means of the pseudo-modes it is possible to show that:

$$\begin{aligned}
\theta_h^{(R)}(t) &= \phi_h^{(R)}(0) e^{-(\sigma_h - \sigma_p)t} \theta_p^{(R)}(t) \ , \\
\theta_h^{(I)}(t) &= \phi_h^{(I)}(0) e^{-(\sigma_h - \sigma_p)t} \theta_p^{(I)}(t) \ .
\end{aligned}$$

Introducing the complex-valued auxiliary function $c(t)$ so that $\theta_p(t) = c(t)e^{-\sigma_p t}$, it is possible to find the resolving differential equation:

$$\dot{c}(t) = |c(t)|^2 e^{-2\sigma_p t} \sigma_p c(t) - \sum_{k=1}^{p-1} e^{-2\sigma_k t} [\phi_k^{(R)}(0)^2 c^{(R)}(t)^2 + \phi_k^{(I)}(0)^2 c^{(I)}(t)^2] \sigma_k c(t) \ .$$

Again, it is useful to define functions:

$$\eta^{(R)}(t) \stackrel{\text{def}}{=} \sum_{k=1}^{p-1} e^{-2\sigma_k t}\sigma_k \phi_k^{(R)}(0)^2 + \sigma_p e^{-2\sigma_p t} \ ,$$
$$\eta^{(I)}(t) \stackrel{\text{def}}{=} \sum_{k=1}^{p-1} e^{-2\sigma_k t}\sigma_k \phi_k^{(I)}(0)^2 + \sigma_p e^{-2\sigma_p t} \ .$$

By means of these functions it is straightforward to show that the real part and the imaginary part of $c(t)$ satisfy system (47), thus the solution has the form (52). Now the time-integrals of $\eta^{(R)}$ and $\eta^{(I)}$ are explicitly required. The first one finds to be:

$$2e^{2\sigma_p t}\int_0^t \eta^{(R)}(\tau)d\tau = e^{2\sigma_p t}\sum_{k=1}^{p-1}\phi_k^{(R)}(0)^2(e^{-2\sigma_k t}-1) + 1 - e^{2\sigma_p t} \ ;$$

the second integral has a similar form, thus we may ultimately write:

$$\frac{1}{|\theta_p(t)|^2} = B - A(t)e^{2\sigma_p t} \ ,$$

where B is a positive constant and $A(t)$ is an increasing function of the time. It is therefore immediate to note that, depending upon initial conditions, it exists a finite time $\bar{t}$ such that: $\frac{1}{|\theta_p(\bar{t})|^2} = 0$. This completes the proof. □

In [76], Oja proposed an algorithm allowing for the extraction of generalized minor component analysis from real-valued signals that overcomes the mentioned stability problem. The aim of this part is to formalize the stabilization method within the optimization framework and to extend the theory to the complex case.

Let us consider, once again, the problem of minimizing the criterion:

$$C(\mathbf{w}) \stackrel{\text{def}}{=} E[g(|\mathbf{w}^H\mathbf{x}|)] + \lambda(\mathbf{w}^H\mathbf{w} - 1) \ , \tag{59}$$

with respect to the weight-vector $\mathbf{w}$. The expression ofor its gradient is:

$$\frac{\partial C}{\partial \mathbf{w}} = E[G(|y|)y^\star\mathbf{x}] + \lambda\mathbf{w} \ , \tag{60}$$

thus the optimal multiplier may be found by vanishing $\mathbf{w}^H\frac{\partial C}{\partial \mathbf{w}}$ under the constraint $\mathbf{w}^H\mathbf{w} = 1$, that is by solving the equation:

$$\mathbf{w}^H\left(\frac{\partial C}{\partial \mathbf{w}}\right)^{\text{opt}} = E[G(|y|)|y|] + 2\lambda^{\text{opt}}(\mathbf{w}^H\mathbf{w}) = 0 \ .$$

The central point is now that as optimality requires $\mathbf{w}^H\mathbf{w} - 1 = 0$, the latter condition is *equivalent* to:

$$E[G(|y|)|y|] + 2\lambda^{\text{opt}} - \bar{\sigma}(\mathbf{w}^H\mathbf{w} - 1)\mathbf{w} = 0 \ ,$$

with $\bar{\sigma}$ being an arbitrary constant. By solving the previous equation for λ^{opt} and replacing the found function in the expression of $\left(\frac{\partial C}{\partial \mathbf{w}}\right)^{\text{opt}}$, we obtain the stabilizable learning rule:

$$\frac{d\mathbf{w}}{dt} = -E[G(|y|)(y^\star\mathbf{x} - |y|^2\mathbf{w})] - \bar{\sigma}(\mathbf{w}^H\mathbf{w} - 1) \ . \tag{61}$$

When $G(u) = 1$, it is possible to prove that the corresponding minor component analysis learning algorithm converges to the expected solution.

Theorem 3 *Let $\mathbf{\Phi} \stackrel{\text{def}}{=} E[\mathbf{x}\mathbf{x}^H]$ be the covariance matrix of the random process $\mathbf{x}(t)$ with eigenpairs $(\sigma_1, \mathbf{q}_1)$, ..., $(\sigma_p, \mathbf{q}_p)$ and $G(u) = 1$ in (61). Suppose eigenvalues are distinct and arranged in descending order, and eigenvectors are normalized so that $\mathbf{q}_k^H\mathbf{q}_k = 1$. If $\bar{\sigma} > \sigma_1$ then the state-vector $\mathbf{w}$ of system (61) with $\mathbf{w}(0)^H\mathbf{q}_p \neq 0$ asymptotically converges towards $\mathbf{q}_p$ up to a phase factor.* □

Proof. System (61) with $G(u) = 1$ can be rewritten $\dot{\mathbf{w}} = -(\mathbf{\Phi} - \bar{\sigma}\mathbf{I})\mathbf{w} + \mathbf{w}^H(\mathbf{\Phi} - \bar{\sigma}\mathbf{I})\mathbf{w}\mathbf{w}$. Define $\bar{\mathbf{\Phi}} \stackrel{\text{def}}{=} -(\mathbf{\Phi} - \bar{\sigma}\mathbf{I})$. The eigenvalues of $\bar{\mathbf{\Phi}}$ are $\bar{\sigma} - \sigma_p > \bar{\sigma} - \sigma_{p-1} > \cdots > \bar{\sigma} - \sigma_1 > 0$, while its eigenvectors coincide to eigenvectors of $\mathbf{\Phi}$. Thus Theorem 1 applies to system $\dot{\mathbf{w}} = \bar{\mathbf{\Phi}}\mathbf{w} + (\mathbf{w}^H\bar{\mathbf{\Phi}}\mathbf{w})\mathbf{w}$ allowing to conclude that $\mathbf{w}$ asymptotically converges to last eigenvector $\mathbf{q}_p$ up to a phase factor. □

It is important to note that this result gives only a *sufficient* condition for the stability of rule (61). The learning rule (61) embodies a modification that is known as 'origin shift' in linear algebra and that in this context has been shown to arise from the Lagrange multiplier method in a natural way.

As a numerical example, let us consider an input random process $\mathbf{x} \in \mathbb{C}^3$ whose covariance matrix has eigenvalues $\sigma_1 = 3$, $\sigma_2 = 2$, and $\sigma_3 = 1$. Such signal is illustrated in Figure 5. Running the learning rule (33), which allows extracting the first principal component from the data, one expects that the first principal mode (θ_1) tends to one (up to a phase factor), while the second and third principal modes tend to zero, where the principal modes have been defined in the Theorem 2. These results are confirmed by the left-hand panel

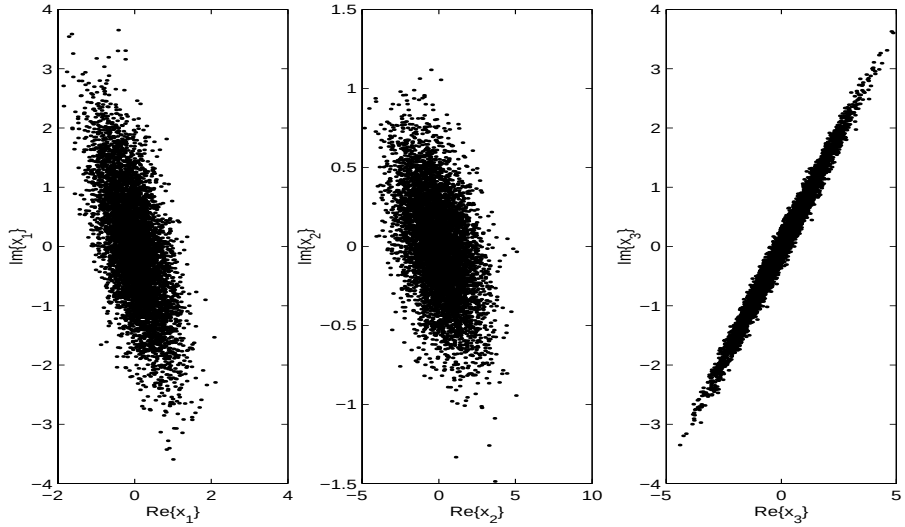

Figure 5: Input data for first principal/minor component analysis experiment.

of Figure 6. Furthermore, we tried to run the learning rule (61) on the same data set in order to extract the first minor component. We tried first with $\bar{\sigma} = 0$, that means using the non-stabilized last minor component analyzer: Simulations show that the rule becomes quickly unstable. Then we tried with $\bar{\sigma} = 1$ as in [76]: Even in this case the rules looks unstable. Finally we tried to use the sufficient condition provided by Theorem 3: Since the largest eigenvalues is 3 we chose $\bar{\sigma} = 3.5$. Simulation results are shown in the right-hand panel of Figure 6. As expected, the first two principal modes converge to zero, while the third mode approaches the value 1.

2.5 Notes on multi-unit complex-valued component/subspace networks

By using the fundamental result found by Oja [73] about the convergence of the neural algorithm allowing to extract the first principal component from a real-valued random process $\mathbf{x}(t)$, Sanger was able to prove the convergence of the generalized Hebbian algorithm (GHA) [82]. Particularly, Sanger proved that the differential system:

$$\frac{d}{dt}\mathbf{W} = \mathbf{\Phi}\mathbf{W} - \mathrm{UT}(\mathbf{W}^T\mathbf{\Phi}\mathbf{W})\mathbf{W} \ ,$$

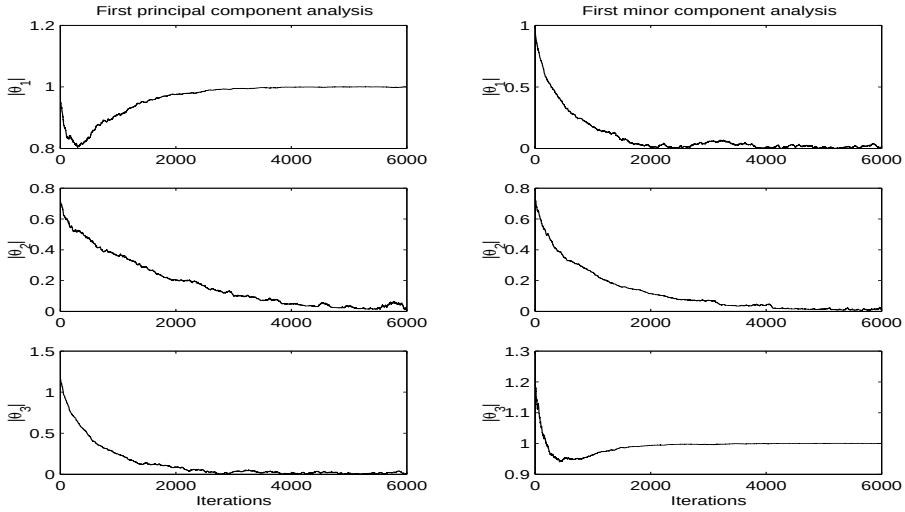

Figure 6: Principal modes modules evolution. Left-hand panel: First principal component analysis case. Right-hand panel: First minor component analysis case.

where $\mathbf{\Phi} \stackrel{\text{def}}{=} E[\mathbf{x}\mathbf{x}^T]$ and $\mathbf{W} \in \mathbb{R}^{p \times m}$, extracts the first m eigenvectors of the covariance matrix $\mathbf{\Phi}$ up to a sign factor.

In the same way, on the basis of the Theorem 1, it would not be difficult to show that the multi-unit learning rule (18), in the hierarchic version, allows for extracting the first m complex eigenvectors up to a phase factor.

The rule (17) was derived via the standard Lagrange multipliers method that is used to enforce the constraints of orthogonality on the weight-vectors. However, the multipliers only guarantee the constraints to be satisfied *at equilibrium*, but – in general – not during system evolution. A detailed analysis of the drawbacks inherent into the use of Lagrange multiplier methods for orthogonality constraints has been proposed recently in [26].

In order to illustrate this effect more clearly, in the case of non-linear complex-valued, component/subspace learning, let us reformulate the orthogonality constraints by borrowing some concepts from differential geometry. (A good reference book is, for instance, [48].) Let us define the complex-valued compact Stiefel manifold as:

$$St(p, m, \mathbb{C}) \stackrel{\text{def}}{=} \{\mathbf{A} \in \mathbb{C}^{p \times m} | \mathbf{A}^H \mathbf{A} = \mathbf{I}_m\} \ .$$

Now it is clear that the requirement that the connection pattern $\mathbf{W}_\star$ at learning Equilibrium is an orthonormal matrix may be equivalently formulated as $\mathbf{W}_\star \in St(p, m, \mathbb{C})$. As the learning rules of interest in the present contribution are expressed in terms of differential equations of the kind $\frac{d\mathbf{w}_k}{dt} = \mathbf{F}_k$, with $\mathbf{F}_k$ being a proper learning vector field for every abstract neuron, it is convenient to express the orthogonality constraints in terms of differential properties. This may be done by invoking the concept of *tangent space* of the Stiefel manifold at a point, namely:

$$T_{\mathbf{W}} St(p, m, \mathbb{C}) = \{\mathbf{V} \in \mathbb{C}^{p \times m} | \mathbf{V}^H \mathbf{W} + \mathbf{W}\mathbf{V}^H = \mathbf{0}_m\} \ .$$

In terms of single weight-vectors, in order for the connection pattern to belong to the Stiefel manifold, they should satisfy:

$$\left(\frac{d\mathbf{w}_k}{dt}\right)^H \mathbf{w}_h + \mathbf{w}_k^H \left(\frac{d\mathbf{w}_h}{dt}\right) = 0 \ , \tag{62}$$

for every pair $(k, h) \in [1, ..., m]^2$. In order to verify if this property is satisfied, it is sufficient to replace the expressions (17) into equations (62). This would lead to the following equations:

$$E[G(|y_k|)y_k \mathbf{x}^H]\mathbf{P}_k^H \mathbf{w}_h + \mathbf{w}_k^H \mathbf{P}_h E[G(|y_h|)y_h^\star \mathbf{x}] = 0 \ .$$

It is now straightforward to find that in the symmetric-network case, the projector operator is such that $\mathbf{P}_k^H \mathbf{w}_h = \mathbf{0}$ for every pair of indices (k, h), while in the hierarchic case it holds $\mathbf{P}_k^H \mathbf{w}_h = \mathbf{0}$ for $k \geq h$ while $\mathbf{P}_k^H \mathbf{w}_h = \mathbf{w}_h$ for $k < h$. As a consequence, when $k < h$, the left-hand side of equation (62) writes:

$$E[G(|y_k|)y_k y_h^\star + G(|y_h|)y_h y_k^\star] \neq 0 \ . \tag{63}$$

From this result it is necessary to conclude that in the symmetric (generalized principal/minor subspace analysis) case, it is true that $\mathbf{W}(t) \in St(p, m, \mathbb{C})$ at any time, while in the hierarchic (generalized principal/minor component analysis) case, in general it happens that $\mathbf{W}(t) \notin St(p, m, \mathbb{C})$ at any time only if the algorithm is out of equilibrium.

It is worth noting, however, that at least in any case it is guaranteed that the optimization of the weight-vectors $\mathbf{w}_k$ happens on a unitary-radius hypersphere. In fact, either in the symmetric and the hierarchic case, it is obtained

from conditions (62) that $\mathbf{w}_k^H \frac{d\mathbf{w}_k}{dt} = 0$, which implies that $\|\mathbf{w}_k\|_2$ is constant (at one, if the weight-vectors are initialized to have unitary norm). Ultimately, this means that vector orthogonality is to be progressively achieved by learning and is not guaranteed at any time. This is a drawback of Lagrange multiplier method.

On the basis of general knowledge and of the results granted from the preceding sections (especially section 2.4), we may summarize these drawbacks as:

- Generally, the method of multiplier consists in embedding the constraints under which the optimization of a criterion function should be performed, in the criterion function itself by adding a properly-constructed function to the original criterion. From a geometrical point of view, it seems unnatural to modifify the objective function instead of restricting it to the set of feasible configurations/variables-values.

- As a consequence of the above-mentioned construction, the constraints should be though to as a sort of sub-target to be achieved in parallel to the optimization of the criterion function, therefore, provided the employed unconstrained optimization algorithm is able to perform the optimization of the augmented criterion, the constraints are fulfilled only when the optimization is completed and not in general at any time.

- The modification of the criterion function may cause the appearance of undesired effects, such as the appearance of local extremes that might make the optimization algorithm unable to solve the learning task or might make the learning progress slower than expected.

- As recalled in the header of section 2, the general idea behind constrained optimization by the Lagrange multiplier method is to replace the constrained optimization flow with an unconstrained optimization flow (that takes places, e.g. in the Euclidean space $\mathbb{R}^p$) by properly embedding the constraints into an augmented learning criterion. It is worth noting that the space that the optimum search takes place in plays an important role in determining the behavior of the gradient-based learning system. As evidenced in the section 2.4, for instance, a learning differential equation on $\mathbb{R}^p$ may be unstable even if it was derived from strong constraints

such that the normality of the weight vectors. Defining learning differential equations on compact manifolds (such as the Stiefel one) instead may help overcoming this serious difficulty.

A detailed treatment of the Lagrange multiplier method may be found e.g. in [14].

In the non-quadratic case, moreover, it is difficult to give general results on learning capabilities. Some recent results for the real-valued case based on the design of appropriate Lyapunov functions for the system equivalent to (56) have been introduced in [89]. These results are valid in the case of homogeneous non-linearities $\mathcal{L}(\mathbf{w})$. It is worth mentioning that for the case that non-linear complex-valued PCA is used for blind source separation of telecommunication-type source signals with a Sudjianto-Hassoun non-linearity, a convergence theorem was given in [34].

Following the line of the preceding notes on multi-unit artificial networks, it also appears appropriate to briefly discuss the important topic concerning the stability of learning equations in view of computer-based implementation. To this aim, it is worth recalling the distinction between the related concepts of *learning differential equation* and *learning algorithm*:

- A learning differential equation arises when a learning trajectory is considered in the weight-space and the learnable variables of an artificial neural networks are parameterized through a free variable (normally the time) that uniquivocally locates a network configuration over a learning trajectory. In the general gradient-based learning setting, for instance, the learning trajectory cannot be written in explicit form and it is specified through a differential equation (often of the first order) whose intuitive meaning is to specify the velocity field associated to learning a specific task. The solution (integration) of such differential equation (provided an initial configuration is specified) gives rise to the network's learning trajectory, formally termed gradient flow.

- A learning differential equation may be solved in closed form very rarely. Normally it is necessary to approximate the gradient flow numerically through an iterative algorithm. This corresponds to sampling the time-

variable and to try to obtain a close approximation of exact learning trajectory. Then, the obtained discrete-time learning equation is what it is normally referred to as learning algorithm.

The passage from a generic learning differential equation to an associated learning algorithm is not unique, as it may be performed in several different ways, and is not consequence-free, because, depending on the chosen time-discretization method, the salient features of the differential equations may be retained or may be lost, as well. In particular, it is worth discussing here the effect of time-discretization on the two salient features of orthogonality-preservation and learning stability:

- **Euler discretization.** The simplest discretization method consists in replacing the derivatives with respect to the time with incremental ratios. This method is widely know as Euler integration technique. For instance, the derivative in equation (3) may be approximated as:

$$\frac{d\mathbf{w}(t)}{dt} \approx \frac{\mathbf{w}(t+T) - \mathbf{w}(t)}{T} \ ,$$

where T denotes the width of the time-interval that the derivative is approximated within. It is clear that the Euler algorithm may be well-suited on a linear space, as the Euclidean manifolds $\mathbb{C}^p$ or $\mathbb{C}^{p \times m}$, while the Stiefel manifold $St(p, m, \mathbb{C})$ is a curved space and it is impossible to move from a network configuration to another configuration through vector/matrix addition. This means that the Euler method does not preserve the orthonormality of a network's connection matrix nor it is guaranteed that the learning system eventually converges to a configuration belonging to $St(p, m, \mathbb{C})$ nor that the network connection keeps in the vicinity of such manifold for sufficiently long time, which may ultimately affect the stability of the learning algorithm. These problems have been widely investigated in the contributions [95, 97]. In the contribution [27] is was shown that by properly selecting the learning stepsize it could be possible to obtain a convergent learning system. In the contributions [18, 19] the problem of the existence of a a Stiefel-manifold attractor for minor component/subspace analysis algorithms was also investigated. It is worth recalling that, as already noted in the section 2.4, the minor compo-

nent/subspace analysis algorithms appear to be the most problematic ones with regard to dynamical stability.

- **Geometric integration.** A good discretization method in the time-domain should allow converting the learning differential equations into discrete-time algorithms so as to retain (up to reasonable precision) the geometric properties that characterize the developed learning rules. From the numerical point of view, the solution of matrix-type differential equations on curved manifolds has been widely investigated over the last years in the context of *geometric integration* (see e.g. [42, 57]), which is a recent branch of numerical analysis. The traditional effort of numerical analysis and computational mathematics has been to render physical phenomena into algorithms that can produce sufficiently affordable, precise and robust numerical approximations. Geometric integration is also concerned with producing numerical approximations preserving the qualitative attributes of the solution to the possible extent. In the context of neural learning under orthogonality constraints, some efforts have been devoted to develop learning algorithm over the group of orthogonal matrices and over the Stiefel manifold. These consist in Riemannian-gradient-based and non-gradient-based learning algorithms which strongly bind the connection patterns to the Stiefel manifold. For a recent review, interested Readers may refer e.g. to [16, 29, 30]. One of the advantages offered by geometric integration over compact manifolds (such as e.g. the orthogonal group and the Stiefel manifold) is the intrinsic stability: Whether the algorithm converge to the expected solution or not, the network connection matrix always keeps of bounded norm.

- **Fixed-point algorithms.** Recently, preliminary results on fast batch-type fixed-point iteration for non-linear Hebbian learning with orthonormality constraints have been illustrated in [35, 36, 37]. The extension of this class of algorithms to the complex domain as well as some fundamental issues such as convergence and computational-complexity reduction are still open problems.

3 Complex-Valued Hebbian Learning by Non-Quadratic Reconstruction Error Minimization

A second philosophy known in the literature allowing formulating Hebbian learning involves the concept of "reconstruction error" minimization.

Let a stationary multivariate random signal $\mathbf{x}(t)$ of size p, whose covariance matrix has distinct eigenvalues, be expanded by means of a basis of size $m < p$ given by $\{\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_m\}$. This means that a new random signal $\hat{\mathbf{x}}(t)$ may be defined so that:

$$\hat{\mathbf{x}} = y_1\mathbf{w}_1 + y_2\mathbf{w}_2 + \cdots + y_m\mathbf{w}_m \ .$$

This is termed the *reconstruction formula*, where the coefficients y_k are defined as the projections $y_k = \mathbf{w}_k^H\mathbf{x}$ in presence of complex-valued signals; projection expressions are termed analysis formulas. It is clear that as $m < p$, in general $\hat{\mathbf{x}}(t) \neq \mathbf{x}(t)$ and the difference between the original and the reconstructed version of $\mathbf{x}$ may be conceived as the *reconstruction error* [53, 92].

In the following sections, we consider non-quadratic complex-valued reconstruction error minimization for a symmetric network, which leads to a generalized principal subspace analysis theory, and the component-wise non-quadratic reconstruction error minimization principle for a hierarchic network leading to a generalized principal component analysis learning theory. We also discuss the problem of non-linearity selection by recalling interesting statistical interpretations of non-quadratic learning criteria.

The rationale for considering non-quadratic reconstruction error optimization was given e.g. in [60]. This contribution was limited to supervised neural learning for modeling purposes, but it keeps valid for the present unsupervised (linear) data modeling. In particular, Liano observed that most neural networks are trained by minimizing the mean squared error pertaining to the given training set. In the presence of outliers, the resulting neural model can differ significantly from the system that generated the data or from the intended abstract model. The purpose of Liano's research was to introduce a robust approach that can minimize the influence of gross errors on the accuracy of the neural model. Two different approaches were used in [60] to study the mechanism by which outliers affect the resulting models: The influence function approach and the

maximum likelihood approach. The same approaches were followed in [86] and are also partially followed here (in fact, the maximum likelihood approach is replaced by the maximum entropy method in the present contribution).

The contents of the present section may be anticipated as:

- Learning principal/minor subspace by minimization of non-quadratic complex-valued reconstruction error.
- Choosing the non-linear criteria via the Song-Yilong-Feng interpretation of Hebbian learning, which is largely based on Liano's research in the context of supervised multilayer perceptron learning.
- Learning principal/minor component by component-wise non-quadratic reconstruction error minimization.
- Illustrative numerical simulation results on the robustness of the considered learning theories.

3.1 Symmetric non-quadratic complex-valued reconstruction error minimization

In the symmetric-network case, the reconstruction error may be formally defined as:

$$\mathbf{e} \stackrel{\text{def}}{=} \mathbf{x} - \hat{\mathbf{x}} = (\mathbf{I} - \mathbf{W}\mathbf{W}^H)\mathbf{x} \ . \tag{64}$$

It may be shown that, in the real-valued case as well as for complex-valued signals, minimizing the mean square reconstruction error leads to principal subspace analysis. The real-valued case has been investigated e.g. by Xu [92]. In the complex-valued case of interest here, this property has been analytically shown by Yang [96] who proved the following two results:

Theorem 4 *([96].)* $\mathbf{W}$ *is a stationary point of* $Y(\mathbf{W}) \stackrel{\text{def}}{=} E[\|\mathbf{x} - \mathbf{W}\mathbf{W}^H\mathbf{x}\|^2]$ *if and only if* $\mathbf{W} = \mathbf{U}_m\mathbf{Q}$ *where* $\mathbf{U}_m \in \mathbb{C}^{p\times m}$ *contains any* m *distinct eigenvectors of* $\mathbf{\Phi} \stackrel{\text{def}}{=} E[\mathbf{x}\mathbf{x}^H]$ *and* $\mathbf{Q} \in \mathbb{C}^{m\times m}$ *is an unitary matrix.* □

Theorem 5 *([96].) All stationary points of* $Y(\mathbf{W})$ *are saddle points except when* $\mathbf{U}_m$ *contains the* m *dominant eigenvectors of* $\mathbf{\Phi}$*. In this case* $Y(\mathbf{W})$ *attains the global minimum.* □

Note the presence of the arbitrary unitary matrix $\mathbf{Q}$ meaning that the network's connection pattern $\mathbf{W}$ that minimizes the reconstruction error corresponding to coefficients $\mathbf{y} = \mathbf{Q}^H \mathbf{U}_m^H \mathbf{x}$ but in general $E[y_k^\star y_j] \neq 0$, that means the network-transformed signals are still correlated.

One of the most interesting features of this approach is that the optimization problem of searching for a weight-matrix $\mathbf{W}$ that minimizes the mean square reconstruction error, is unconstrained, in that no constraints have to be added to the error cost function in order for the minimization problem to be consistent. This appears as a noticeable difference with respect to the material presented in the section 2.

In order to reduce the effects of noise, disturbances and outliers affecting the data-sets, generalized versions of the mean square reconstruction error minimization based PSA algorithms have been developed for analyzing real-valued signals. In what follows we extend this enhanced theory to the complex domain. Instead of minimizing the quantity $E[\|\mathbf{e}\|^2]$, here we derive a generalized PSA algorithm by minimizing, through the use of a gradient-based searching procedure, a more general cost defined as:

$$Z(\mathbf{w}_k) \stackrel{\text{def}}{=} E[\phi(\|\mathbf{e}\|^2)] \ , \tag{65}$$

thought of as a function of any single vector $\mathbf{w}_k$ for easy notation, where $k = 1, 2, \ldots, m$. The function $\phi(u)$ with $u \geq 0$ should be differentiable, convex and should possess a unique minimum in $u = 0$. The problem about its choice is crucial for the behavior of the PSA algorithm and will be discussed later.

In order to minimize function $Z(\cdot)$ with respect to each $\mathbf{w}_k$ by means of a gradient steepest descent recursive algorithm, its gradients are needed, therefore we want to evaluate:

$$\frac{\partial Z(\mathbf{w}_k)}{\partial \mathbf{w}_k} = E\left[\frac{d\phi(\|\mathbf{e}\|^2)}{d(\|\mathbf{e}\|^2)} \frac{\partial \|\mathbf{e}\|^2}{\partial \mathbf{w}_k}\right] \ . \tag{66}$$

Note that from the definition (64) there holds:

$$\|\mathbf{e}\|^2 = \mathbf{x}^H\mathbf{x} - 2\mathbf{x}^H\mathbf{W}\mathbf{W}^H\mathbf{x} + \mathbf{x}^H\mathbf{W}\mathbf{W}^H\mathbf{W}\mathbf{W}^H\mathbf{x} \ .$$

The expression of the gradient of $\|\mathbf{e}\|^2$, is:

$$\frac{\partial \|\mathbf{e}\|^2}{\partial \mathbf{w}_k} = 2(\mathbf{W}\mathbf{y} - 2\mathbf{x})y_k^\star + 2\mathbf{y}^H\mathbf{W}^H\mathbf{w}_k\mathbf{x} \ .$$

By recalling the identity $\mathbf{W}\mathbf{y} = -\mathbf{e} + \mathbf{x}$, the above expression simplifies into:

$$\frac{1}{2}\frac{\partial \|\mathbf{e}\|^2}{\partial \mathbf{w}_k} = -\mathbf{e}y_k^\star - \mathbf{x}\mathbf{e}^H\mathbf{w}_k \ . \tag{67}$$

Let us define $\Phi(u) \stackrel{\text{def}}{=} \frac{d\phi(u)}{du}$. This is what Liano refers to as 'influence' function in [60]. Then, the gradient steepest descent learning equations for generalized principal subspace extraction for complex-valued signals read:

$$\frac{d\mathbf{w}_k}{dt} = -\frac{\partial Z(\mathbf{w}_k)}{\partial \mathbf{w}_k} = E[\Phi(\|\mathbf{e}\|^2)(\mathbf{e}y_k^\star + \mathbf{x}\mathbf{e}^H\mathbf{w}_k)] \ , \ k = 1, 2, \ldots, m \ , \tag{68}$$

with $\mathbf{e}$ being defined by (64).

Choosing the non-linear function $\Phi(\cdot)$ is not generally an easy task. Song, Yilong and Feng proposed a theory allowing to link the choice of $\Phi(\cdot)$ to the statistical features of the reconstruction error for real-valued signals [86]. The aim of the following section is to show how these results may be extended to the complex-valued case.

3.2 The Song-Yilong-Feng principle and its extension to the complex-valued case

The Song-Yilong-Feng theory for real-valued ergodic input-signal $\mathbf{x}$ is based on the definition of the random variable:

$$\varepsilon(\mathbf{x}, \mathbf{W}) \stackrel{\text{def}}{=} \|\mathbf{x} - \mathbf{W}\mathbf{W}^T\mathbf{x}\|^2 \ . \tag{69}$$

Let us suppose that several random input vectors $\{\mathbf{x}_n\}_{n=1,N}$ have been collected. For each input sample we may compute a reconstruction error value $\varepsilon_n = \varepsilon(\mathbf{x}_n, \mathbf{W})$. The likelihood of $\mathbf{W}$ is then defined as the probability:

$$L(\mathbf{E}|\mathbf{W}) \stackrel{\text{def}}{=} q(\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_N|\mathbf{W}) \ , \tag{70}$$

where $\mathbf{E} \stackrel{\text{def}}{=} [\varepsilon_1 \ \varepsilon_2 \ \cdots \ \varepsilon_N]$ and $q(\cdot, \ldots, \cdot|\cdot)$ represents the joint probability density function of the ε_n subjected to the hypothesis $\mathbf{W}$. Under the hypothesis that the $\mathbf{x}_n$ are identically distributed and statistically independent (i.i.d.), the likelihood writes:

$$L(\mathbf{E}|\mathbf{W}) = \prod_{n=1}^{N} q_\varepsilon(\varepsilon_n|\mathbf{W}) = \prod_{n=1}^{N} q(\varepsilon_n) \ , \tag{71}$$

in short notation[4]. Having defined the likelihood of a network connection matrix under the observed reconstruction error values, we may invoke the maximum likelihood estimation theory as a learning tool for the neural network. In order to make this learning principle be profitable, we should hypothesize a distribution for the reconstruction error values.

In the following, we formulate the maximum-likelihood theory in a version suitable for treating an infinite number of available samples and discuss some possible choices of the error distributions. We also relate the Song-Yilong-Feng theory to the conceptualization of robust classification proposed by Xu and Yuille.

Let us suppose that the probability density function of the square reconstruction error ε has the *negative exponential* (one-side Laplacean) form:

$$q_{\mathrm{E}}(u) \stackrel{\mathrm{def}}{=} \kappa e^{-\kappa u}\Gamma(u) \ , \tag{72}$$

where $\kappa > 0$ denotes the Laplacean dispersion parameter and $\Gamma(u)$ denotes again the unit-step function. Then, the log-likelihood takes on the expression:

$$\log L(\mathbf{E}|\mathbf{W}) = N \log \kappa - \kappa \sum_{n=1}^{N} \|\mathbf{x}_n - \mathbf{W}\mathbf{W}^T\mathbf{x}_n\|^2 \ .$$

The meaning of this expression is that the maximum likelihood principle and the minimum average square error principle coincide if the conditioned probability density function $q_\varepsilon(\varepsilon|\mathbf{W})$ is assumed as in (72). Note that the above statement is based on the hypothesis that N is *finite*, otherwise the log-likelihood could be unbounded.

On the basis of this observation, Song, Yilong and Feng proposed in [86] to extend the idea to a generic probability density function, replacing the average-square-error principle with the maximum-likelihood estimation of the weight-matrix $\mathbf{W}$. It consists in looking for a weight-matrix maximizing the quantity:

$$\log L(\mathbf{E}|\mathbf{W}) = \sum_{n=1}^{N} \log q(\varepsilon_n) \ . \tag{73}$$

Now, we aim at extending this optimization criterion in two ways:

[4]It might be worth recalling that successive $\mathbf{x}_n$ are supposed to be i.i.d. but not their components.

- First, the maximum-likelihood theory may be made rigorous for an infinite-size data-set (i.e. for density-function-based descriptions instead of sample-based). The key-step for this extension relies on the use of differential entropy associated to a probability distribution.
- Second, we want to extend the Song-Yilong-Feng principle to complex domain. As it shall be shortly clear, this extension leads us to face again the problem of nonlinear function selection. This problem may be solved directly through the Song-Yilong-Feng principle (an especially interesting formulation comes by choosing the Cauchy-Lorenz distribution for the reconstruction error model) and also from a classification theory by Yuille and Xu in [94] based on statistical physics.

In presence of infinitely many observation, instead of using the likelihood, the differential conditional entropy may be adopted. To this aim, let us define the function:

$$H(\mathbf{W}) \stackrel{\text{def}}{=} -E_{\mathbf{x}}[\log q_{\varepsilon}(\varepsilon|\mathbf{W})|\mathbf{W}] = -E[\log q(\varepsilon)] \tag{74}$$

in short. The quantity $H(\mathbf{W})$ plays the role of a 'minus-mean-log-likelihood'. If we assume again the distribution (72) for the square error ε, we find:

$$H(\mathbf{W}) = -\log \kappa + \kappa E[\|\mathbf{e}\|^2] \ .$$

If the reconstruction error is defined for complex-valued signals as in (64), minimizing the entropy H with respect to $\mathbf{W}$ also means minimizing the mean square error, in symbols:

$$\min_{\mathbf{W}\in\mathbb{C}^{p\times m}}\{H(\mathbf{W})\} \iff \min_{\mathbf{W}\in\mathbb{C}^{p\times m}}\{E[\|\mathbf{e}\|^2]\} \ .$$

Following the idea in [86], the minimum-entropy principle may be extended by assuming different probability density functions $q(\cdot)$. In order to find the matrix $\mathbf{W}$ minimizing the function (74), the gradient steepest descent approach may be considered. Its use requires the evaluation of the gradient:

$$\frac{\partial H(\mathbf{w}_k)}{\partial \mathbf{w}_k} = -E\left[\frac{q'(\|\mathbf{e}\|^2)}{q(\|\mathbf{e}\|^2)}\frac{\partial \|\mathbf{e}\|^2}{\partial \mathbf{w}_k}\right] \ .$$

Defining the statistical 'score function' $\Psi(u) \stackrel{\text{def}}{=} -\frac{q'(u)}{q(u)}$, the gradient steepest descent algorithm pertaining to the entropy (74) minimization reads:

$$\frac{d\mathbf{w}_k}{dt} = E[\Psi(\|\mathbf{e}\|^2)(\mathbf{e}y_k^{\star} + \mathbf{x}\mathbf{e}^H\mathbf{w}_k)] \ . \tag{75}$$

Note that the equations (68) and (75) are identical except for the form of functions $\Phi(\cdot)$ and $\Psi(\cdot)$. The choice of $\Psi(\cdot)$ is driven by the hypotheses about the statistics of the scalar quadratic error $\|\mathbf{e}\|^2$.

In order to illustrate the usefulness of the differential-entropy-based approach, let us consider in details the probability density functions q_{E} defined by (72) and the modified (one-sided) Cauchy-Lorentz distribution recommended in [60, 86]:

$$q_{\mathrm{C}}(u) \stackrel{\mathrm{def}}{=} \frac{2}{\pi}\frac{\theta}{\theta^2+u^2}\Gamma(u) \ , \ \theta > 0 \ .$$

The most interesting difference between the Laplacean and Cauchy probability density functions is that the first distribution is characterized by a mean value and a variance, while the second distribution has *not* a mean value nor a variance[5]. This special feature of Cauchy distribution deserves some comments:

- If we assume a Cauchy distribution for the quadratic error $\|\mathbf{e}\|^2$ instead of the Laplacean one, we do not need to embody any *a-priori* hypotheses about the variability of the error in the algorithm. In other words, the risk of biasing the error distribution with inaccurate information is less serious than with the Laplacean distribution.

- Another interesting implication of the inexistence of moments of the Cauchy distribution is that the effects of Cauchy-based modeling are less tied to the number of available samples. In fact, the practical meaning of the inexistence of moments is that collecting several data points gives no more accurate an estimate of the moments than a single data point does.

- However, it is important to remark that the Cauchy density does impose scale limitation, i.e. the choice of the width parameter θ does affect the width of the receptive fields of the neurons in the component/subspace analysis network.

The score function corresponding to the Cauchy distribution is:

$$\Psi_{\mathrm{C}}(u) = \frac{2u}{\theta^2+u^2}\Gamma(u) \ .$$

[5]In fact, the Cauchy distribution does not admit any statistical moment because its characteristic function is not differentiable at the origin.

Several expressions for the function $\phi(\cdot)$ in (65) have been proposed in the literature [52, 53, 94], usually justified by invoking the theory of robust statistics. These different choices may be discussed under the light of the Song-Yilong-Feng interpretation. In order to accomplish this, it is useful to equate functions $\Phi(u)$ and $\Psi(u)$. It is worth recalling that:

$$\Phi(u) = \frac{d\phi(u)}{du} \ , \ \Psi(u) = -\frac{1}{q(u)}\frac{dq(u)}{du} \ ,$$

where function $\phi(u)$ was introduced in formula (65) and the probability density function $q(u)$ takes part e.g. in formula (71). From these relationships, the following formula is readily found:

$$\phi(u) = -\log q(u) + c \ , \tag{76}$$

where $c \in \mathbb{R}$ is a constant. The above formula may be used in two directions:

- If the probability distribution model for the reconstruction noise $q(\cdot)$ has been selected, then the corresponding warping function $\phi(\cdot)$ may be computed by formula (76). Care should be taken that the resulting function $\phi(\cdot)$ must fulfill regularity as well as convexity requirements, by definition, otherwise the pair (q, ϕ) must be rejected as it is not valid in the present framework. If the pair (q, ϕ) is acceptable, the constant c, whose value does not influence the shape of the warping function $\phi(\cdot)$ but only its vertical placement, may be chosen arbitrarily (e.g. in order to simplify the expression of function $\phi(\cdot)$).

- Alternatively, formula (76) may help discussing some choices of $\phi(\cdot)$ in terms of probability distributions for the square reconstruction error. In this case, care should be taken that the resulting function $q(\cdot)$ be a probability density function, i.e. it is a positive integrable function with unitary area. In this case, the constant c may be determined by formula $c = -\log \int_0^{+\infty} e^{-\phi(\tau)} d\tau$.

As two examples of the first case, we may find the functions $\phi(\cdot)$ corresponding to the Laplacean and Cauchy-Lorentz distributions. Straightforward computations show that they write:

$$\begin{aligned} \phi_{\mathrm{E}}(u) &= -\log \kappa + c + \kappa u \ , \\ \phi_{\mathrm{C}}(u) &= -\log\left(\frac{2\theta}{\pi}\right) + c + \log(\theta^2 + u^2) \ , \end{aligned}$$

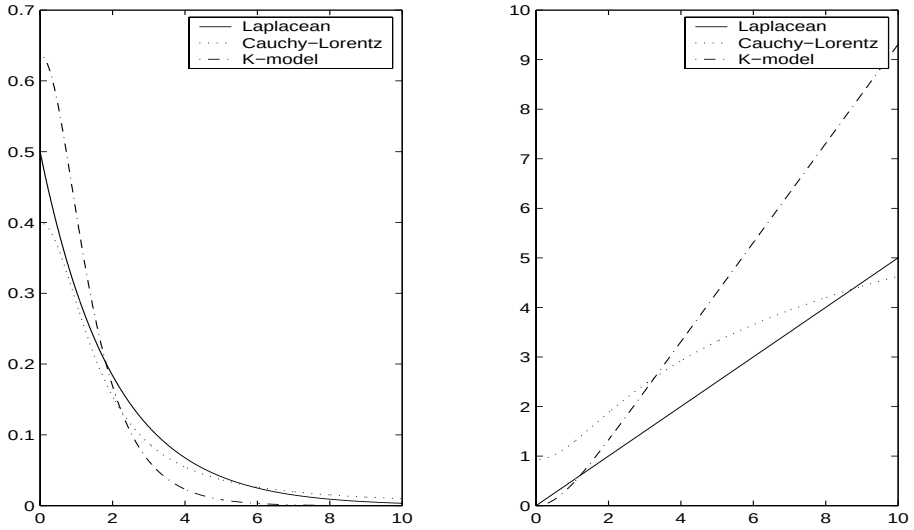

Figure 7: Left panel: Reconstruction error distribution functions q_{E} ($\kappa = 0.5$, [86]), q_{C} ($\theta = 5/\pi$, [86]) and q_{K} ($\beta = 1$, [53]). Right panel: Warping functions ϕ_{E}, ϕ_{C} and ϕ_{K}.

respectively. In these cases, the constant c may be chosen to be zero or e.g. for the Cauchy distribution case as $c = \log\left(\frac{2\theta}{\pi}\right)$, that simplifies the expression of $\phi_{\mathrm{C}}(u)$. The obtained functions may be easily verified to fulfill the required restrictions such as regularity and convexity, at least in a properly chosen right-sided neighborhood of the origin. As an example of the second case, let us consider the warping function $\phi_{\mathrm{K}}(u) \stackrel{\mathrm{def}}{=} \log\cosh(\beta u)$ found in [52, 53]. The corresponding $q(\cdot)$ function is computed to be:

$$q_{\mathrm{K}}(u) = \frac{2}{\pi}\frac{2\beta}{e^{-\beta u} + e^{\beta u}}\Gamma(u) \ ,$$

which may be easily proven to be a density distribution and shall hereafter be referred to as 'K' density model. The shape of the above distributions q_{E}, q_{C} and q_{K}, along with the corresponding warping functions ϕ_{E}, ϕ_{C} and ϕ_{K} are depicted in the Figure 7 for arbitrary parameters values. Both in the negative exponential distribution model and in the 'K' model, as the square reconstruction error ε increases, the density functions value decreases rapidly. This means that in those distributions a large reconstruction error (usually associated to disturbances) has a small probability to occur, then the model does not take into account properly the presence of outliers. Conversely, the use of the modified Cauchy

distribution improves the robustness of the PSA algorithm since it decreases slower than the former two.

To end with, it is interesting to discuss the relationships among the Song-Yilong-Feng theory and the conceptualization proposed by Xu and Yuille in [94] based on statistical physics. Let us consider the reconstruction error $\varepsilon(\mathbf{x}, \mathbf{w})$ due to a one-unit artificial neural network:

$$\varepsilon(\mathbf{x}, \mathbf{w}) \stackrel{\text{def}}{=} \|\mathbf{x} - (\mathbf{w}^H \mathbf{x})\mathbf{w}\|^2 \ , \ \mathbf{x} \in \{\mathbf{x}_n\}_{n=1,N} \ .$$

Xu and Youille [94] considered a neuron's energy function to be minimized for statistical learning:

$$K(\mathbf{f}, \mathbf{w}) \stackrel{\text{def}}{=} \sum_{n=1}^{N} f_n \varepsilon(\mathbf{x}_n, \mathbf{w}) + \eta \sum_{n=1}^{N} (1 - f_n) \ , \tag{77}$$

where η is a constant and the f_k's are random binary variables acting as decision flags indicating whether $\mathbf{x}_k$ is an outlier ($f_k = 0$) or not ($f_k = 1$) and $\mathbf{f} \stackrel{\text{def}}{=} [f_1 \ f_2 \ \cdots \ f_N]^T$. The aim is to minimize $K(\mathbf{f}, \mathbf{w})$ with respect to $\mathbf{f}$ and $\mathbf{w}$ simultaneously. To solve this difficult problem, Xu and Yuille defined a Gibbs distribution:

$$\rho(\mathbf{f}, \mathbf{w}) \stackrel{\text{def}}{=} \frac{1}{\Theta} e^{-\beta K(\mathbf{f}, \mathbf{w})} \ ,$$

where Θ is a normalization constant and $1/\beta$ stands for a 'temperature' (for a discussion on the meaning of parameters η and β see [94]). Then they proposed to approximate $\rho(\mathbf{f}, \mathbf{w})$ by computing the marginal probability distribution $\rho_{\text{m}}(\mathbf{w})$ obtained averaging $\rho(\mathbf{f}, \mathbf{w})$ over all possible configurations $\mathbf{f}$ (which may be though to as a kind of mean-field approximation). This gives [94] $\rho_{\text{m}}(\mathbf{w}) = \frac{1}{\Theta_{\text{m}}} e^{-\beta K_{\text{eff}}(\mathbf{w})}$, with $\Theta_{\text{m}} \stackrel{\text{def}}{=} \Theta e^{N\beta\eta}$ and:

$$-K_{\text{eff}}(\mathbf{w}) \stackrel{\text{def}}{=} \frac{1}{\beta} \sum_{n=1}^{N} \log\{1 + e^{-\beta[\varepsilon(\mathbf{x}_n, \mathbf{w}) - \eta]}\} \ . \tag{78}$$

Then, the optimal connection configuration $\mathbf{w}^{\text{opt}}$ is found by maximizing $-K_{\text{eff}}(\mathbf{w})$. It is now interesting to note that $-K_{\text{eff}}(\mathbf{w})$ is very similar to $\log L(\mathbf{E}|\mathbf{w})$ in (73), thus we may define a probability density function:

$$q_{\text{X}}(u) \stackrel{\text{def}}{=} A[1 + e^{-\beta(u-\eta)}]^{\frac{1}{\beta}} (1 - \Gamma(u - \Xi)) \ , \tag{79}$$

where A is a normalization constant and the factor $1 - \Gamma(u - \Xi)$ makes $q_{\text{X}}(u)$ vanish outside a compact domain $[0, \Xi]$. Then the normalization constant can be

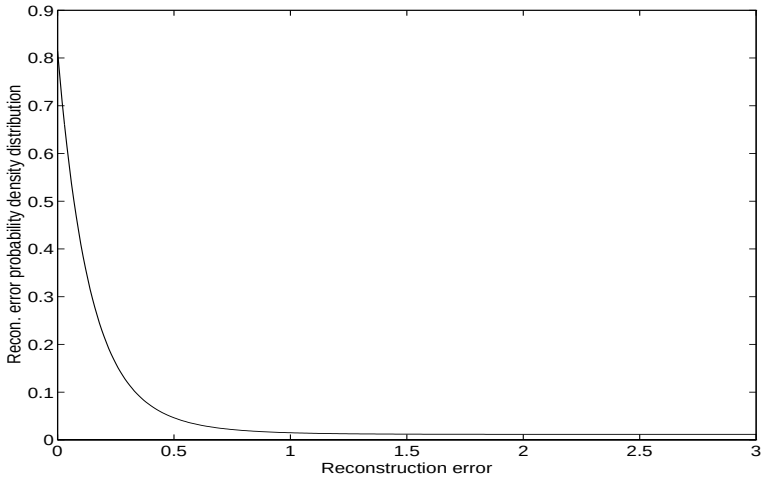

Figure 8: Reconstruction error distribution functions q_{X} ($\beta = 0.5$, $\eta = 4/\pi$ as in [94] and $\Xi = 3$).

found by $\frac{1}{A} = \frac{1}{\beta}\int_D^1 u^{-1}(1+Cu)^{\frac{1}{\beta}}du$, where $D \stackrel{\mathrm{def}}{=} \exp(-\beta\Xi)$ and $C \stackrel{\mathrm{def}}{=} \exp(\beta\eta)$. Note that Ξ may be arbitrarily large, albeit bounded. A distribution q_{X} is shown if Figure 8. The distribution $q_{\mathrm{X}}(u)$ represents an equivalent reconstruction error distribution that would make it possible to select a proper non-linear warping function in reconstruction-error-based learning via the Xu-Yuille statistical-physics-based theory.

3.3 Component-wise non-quadratic reconstruction error minimization

The learning theory discussed in the section 3.1 only allows extracting a principal subspace from a set of complex-valued random signals. A neural network trained by means of that algorithm is not able to effect PCA substantially because of its inherent symmetry.

With the aim to 'break' the symmetry and make the network hierarchic, it is possible to define a reconstruction error vector for each neuron [53] and to design a learning rule for each neuron so that it tries to minimize a local error. Of course, again only linear reconstruction is dealt with. A possible definition

is:

$$\mathbf{e}_k = \mathbf{x} - \sum_{j=1}^{K(k)} y_j \mathbf{w}_j \ , \tag{80}$$

where k ranges from 1 to m, the number of the neurons. The indexing function $K(k)$ determines whether the network is symmetric or not. If $K(k) = m$ we come back to the symmetric case, i.e. all the neurons learn to correct the total error $\mathbf{e} = \mathbf{x} - \sum_{j=1}^{m} y_j \mathbf{w}_j$ while $K(k) = k$ makes the network hierarchic[6].

Following [53], here we discuss the optimization of the cost function:

$$R(\mathbf{e}_k) \stackrel{\text{def}}{=} \sum_{r=1}^{p} E[f(e_{kr})] \ , \tag{81}$$

where, contrary to the case discussed in section 3.1, the r^{th} entry of the k^{th} error vector $\mathbf{e}_k$, namely e_{kr}, warps by the function $f(\cdot)$. Again, $f(\zeta) = g(|\zeta|)$ for $\zeta \in \mathbb{C}$, with $g(u)$ being differentiable, convex, with a unique minimum in $u = 0$ in a convenient right-sided neighborhood of the origin. Evaluating the gradient of $R(\mathbf{e}_k)$ with respect to the weight-vector $\mathbf{w}_k$ involves rather difficult operations that may be considered interesting from a methodological point of view, hence some details about its computation are given in what follows. Particularly, we need the *Hadamard product* operator, defined as:

$$\mathbf{a} \circ \mathbf{c} = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_p \end{bmatrix} \circ \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_p \end{bmatrix} \stackrel{\text{def}}{=} \begin{bmatrix} a_1 c_1 \\ a_2 c_2 \\ \vdots \\ a_p c_p \end{bmatrix} \ ,$$

with $\mathbf{a}, \mathbf{c}$ being p-dimensional real- or complex-valued vectors. Note that the Hadamard product is commutative and associative. Furthermore, we need a set of vectors $\mathbf{b}_r$: Each $\mathbf{b}_r \in \mathbb{R}^p$ is defined so that its entries are 0 except for the r^{th} one equal to 1 (i.e. the $\{\mathbf{b}_r\}$ is the canonical basis of $\mathbb{R}^p$).

To begin with the calculation of $\frac{\partial R(\mathbf{e}_k)}{d\mathbf{w}_k}$, let us consider the quantity $|e_{kr}|^2$, that has the structure:

$$|e_{kr}|^2 = x_r^\star x_r - 2P(e_{kr}) + Q(e_{kr}) \ , \tag{82}$$

[6]Note that in the hierarchic case the reconstruction error may be defined recursively as: $\mathbf{e}_1 = \mathbf{x} - y_1 \mathbf{w}_1$, while for $k \geq 2$: $\mathbf{e}_k = \mathbf{e}_{k-1} - y_k \mathbf{w}_k$.

where the following quantities have been defined:

$$P(e_{kr}) \stackrel{\text{def}}{=} \sum_{j=1}^{K(k)} \text{Re}\{y_j x_r^\star w_{jr}\} \ ,$$

$$Q(e_{kr}) \stackrel{\text{def}}{=} \sum_{j=1}^{K(k)} \sum_{s=1}^{K(k)} \text{Re}\{y_j^\star y_s w_{jr}^\star w_{sr}\} \ .$$

The gradient of $P(e_{kr})$ computed with respect to the complex-valued weight-vector $\mathbf{w}_k$ is found to be:

$$\frac{\partial P(e_{kr})}{\partial \mathbf{w}_k} = y_k^\star \mathbf{x} \circ \mathbf{b}_r + (x_r^\star w_{kr})\mathbf{x} \ . \tag{83}$$

About the gradient of $Q(e_{kr})$, after some mathematical work we find:

$$\frac{1}{2}\frac{\partial Q(e_{kr})}{\partial \mathbf{w}_k} = \sum_{j=1}^{K(k)} \left(y_k^\star y_h \mathbf{w}_j \circ \mathbf{b}_r + y_j^\star w_{hr}^\star w_{kr} \right) \mathbf{x} \ . \tag{84}$$

Now the effects of the non-linearity $f(\cdot)$ have to be taken into account. For the sake of notational compactness, let us define functions:

$$b(u) \stackrel{\text{def}}{=} \frac{dg(u)}{d(u^2)} \ , \ u \geq 0 \text{ and } B(\zeta) \stackrel{\text{def}}{=} \frac{b(|\zeta|)}{|\zeta|} \ , \ \zeta \in \mathbb{C} \ . \tag{85}$$

Let us evaluate the gradient of $R(\mathbf{e}_k)$ with respect to the weight-vector $\mathbf{w}_k$:

$$\frac{\partial R(\mathbf{e}_k)}{\partial \mathbf{w}_k} = \frac{1}{2} \sum_{r=1}^{p} E\left[\frac{b(|e_{kr}|)}{|e_{kr}|} \frac{\partial |e_{kr}|^2}{\partial \mathbf{w}_k} \right] \ . \tag{86}$$

From equations (82)-(86), rearranging terms, recalling the use of the Hadamard operator '$\circ$' and allowing function $B(\cdot)$ to operate component-wise, we have:

$$\begin{aligned} \frac{1}{2}\frac{\partial R(\mathbf{e}_k)}{\partial \mathbf{w}_k} &= E\left\{ \sum_{j=1}^{K(k)} (y_j^\star \mathbf{w}_j^H)[B(\mathbf{e}_k) \circ \mathbf{w}_k]\mathbf{x} \right\} \\ &+ E\left\{ y_k^\star B(\mathbf{e}_k) \circ \sum_{j=1}^{K(k)} y_j \mathbf{w}_j \right\} \\ &- E\left\{ y_k^\star \mathbf{x} \circ B(\mathbf{e}_k) + \mathbf{x}^H [B(\mathbf{e}_k) \circ \mathbf{w}_k]\mathbf{x} \right\} \ . \end{aligned}$$

By replacing the term $\sum_{j=1}^{K(k)} y_j \mathbf{w}_j$ with $\mathbf{x} - \mathbf{e}_k$, we obtain the gradient steepest descent learning algorithm:

$$\frac{d\mathbf{w}_k}{dt} = -\frac{1}{2}\frac{\partial R(\mathbf{e}_k)}{\partial \mathbf{w}_k} = E\left\{ \mathbf{e}_k^H [B(\mathbf{e}_k) \circ \mathbf{w}_k]\mathbf{x} + y_k^\star [B(\mathbf{e}_k) \circ \mathbf{e}_k] \right\} \ . \tag{87}$$

Let us consider the non-quadratic warping function $g(u) \stackrel{\text{def}}{=} 2u^3$ for $u \in \mathbb{R}_0^+$. This choice yields $B(z) = 1$, hence the learning algorithm:

$$\frac{d\mathbf{w}_k}{dt} = E[(\mathbf{e}_k^H \mathbf{w}_k)\mathbf{x} + y_k^\star \mathbf{e}_k] \ , \ k = 1, 2, \ldots, m \ , \tag{88}$$

that coincides with the learning rule (68) when $\Phi(\cdot) = 1$ and $K(k) = m$. Also, in presence of real-valued input signals, the quantities involved in equation (87) are real-valued, thus it may be rewritten as:

$$-\frac{1}{2}\frac{\partial R(\mathbf{e}_k)}{\partial \mathbf{w}_k} = E\left\{\mathbf{e}_k^T[B(\mathbf{e}_k) \circ \mathbf{w}_k]\mathbf{x} + y_k[B(\mathbf{e}_k) \circ \mathbf{e}_k]\right\} \ . \tag{89}$$

Now, since $B(u) = b(u)/u$ for $u \in \mathbb{R}$, if we allow $b(\cdot)$ to operate component-wise, the following identities hold:

$$B(\mathbf{e}_k) \circ \mathbf{e}_k = b(\mathbf{e}_k) \text{ and } \mathbf{e}_k^T[B(\mathbf{e}_k) \circ \mathbf{w}_k] = \mathbf{w}_k^T b(\mathbf{e}_k) \ ,$$

therefore the learning rule corresponding to the gradient (89) for a real-valued PCA neural network reads:

$$\frac{d\mathbf{w}_k}{dt} = E[\mathbf{w}_k^T b(\mathbf{e}_k)\mathbf{x} + \mathbf{w}_k^T \mathbf{x} b(\mathbf{e}_k)] \ , k = 1, 2, \ldots, m \ . \tag{90}$$

This learning rule coincides with the non-classic PCA extraction rule discussed, for instance, in [53].

With reference to the choice of the non-linear function $g(\cdot)$, it seems reasonable to extend the entropy-version of the Song-Yilong-Feng theory in order to model the probability density function of any single reconstruction error $\mathbf{e}_k$. Formally, we can then particularize the cost function (81) as $-\sum_{j=1}^{p} E[\log q(|e_{kj}|^2)]$, that ultimately means identifying $g(u) = -\log q(u^2)$. By choosing as the conditional probability density function the one-sided Cauchy distribution, plugging this $g(u)$ into the definition of $B(\cdot)$ leads to:

$$B_{\mathrm{C}}(z) = \frac{2|z|}{\theta^2 + |z|^4} \ . \tag{91}$$

This shows that the approach in [86] may be successfully extended to the component-wise reconstruction error minimization method.

The extended Hebbian learning theory described in the present section relates to a series of contributions. One of them is the Xu's interpretation of extended Hebbian learning known as 'maximum uncertainty theory' (a detailed

analysis of which has been recently presented in [29]). Other interesting contributions to this topic are the learning theories by Corchado-Fyfe (see [33, Section 2.5]), Miao-Hua [65], Plumbley [79], Shirazi-Peper-Sawai [84] and Higuchi-Eguchi [49].

3.4 Illustrative numerical experiments

In order to illustrate the concept of principal component/subspace analysis in the complex domain by non-quadratic reconstruction error minimization via examples, the results of some numerical simulations are reported below. The experiments aimed at illustrating the convergence of the learning algorithms (68) and (87), and to compare the performances of the robust and non-robust versions in presence of complex-valued data corrupted by gross outliers. Robusteness should emerge by the proper choice of the non-linear warping functions $\Phi(\cdot)$ and $\Psi(\cdot)$.

The performed experiments re-trace the ones suggested in [86, 94]: A network with two inputs x_1 and x_2, and one output is trained both with a set of uncorrupted samples and with the same data set corrupted by a percentage of 3% of outliers. The two data sets are shown in Figure 9. Note that for a single-unit neural network the distinction between PCA and PSA disappears.

The network weight-vector is denoted here by $\mathbf{w}$. Furthermore, the quantity $\mathbf{q}_1$ denotes the first normalized eigenvector of the covariance matrix of the uncorrupted data (i.e. the *true* eigenvector). As component/subspace analysis performance index, we consider the *principal angle* γ, here defined in the following way:

$$\gamma \stackrel{\text{def}}{=} \cos^{-1} \frac{|\mathbf{w}^H \mathbf{q}_1|}{\mathbf{w}^H \mathbf{w}} \ .$$

The outliers are such that they produce a deviation between the true first eigenvector and the noisy first eigenvector of about 30°.

First, the algorithms (68) and (87) have been trained on the uncorrupted data, and both have been able to detect the principal eigenvector corresponding to a principal angle value close to zero.

Second, the algorithm (68) with $\Psi(\cdot) = 1$ was trained on the noisy data: The results are shown in Figure 10. The same algorithm, trained on the same data set, is made robust by introducing the Cauchy-Lorentz non-linearity $\Psi(\cdot) =$

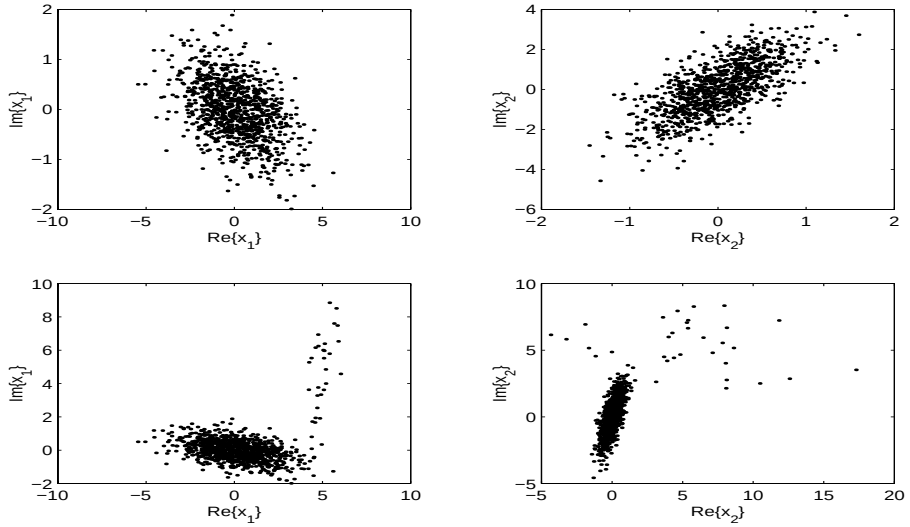

Figure 9: Experiments on reconstruction-error-minimization-based learning: Uncorrupted data set (top panels) and data set with outliers (bottom panels).

$\Psi_{\mathcal{C}}(\cdot)$, and has given instead the result shown in Figure 11, which pertain to the value of the Cauchy parameter $\theta = 5$. The Figures 10 and 11 show the behavior of the principal angle γ as well as the values of the norm of the reconstruction error $\|\mathbf{e}\|^2$ during learning. The net result of the introduction of the Cauchy non-linearity is apparent, as it allows neglecting almost completely the effects of outliers on learning.

The results in both robust/non-robust cases pertaining to the algorithm (87) are completely equivalent and are therefore not shown.

4 Conclusion

The present contribution was born around 1997 and has originated much research work. As the principal component analysis literature is very large and fragmented, this contribution aimed at bringing together a number of mathematical results for principal components analysis and its generalizations such as principal subspace and minor component analysis in the complex domain. The emphasis of the paper was on extensions to complex-valued neural networks and on relating these to a number of previous results known from scientific lit-

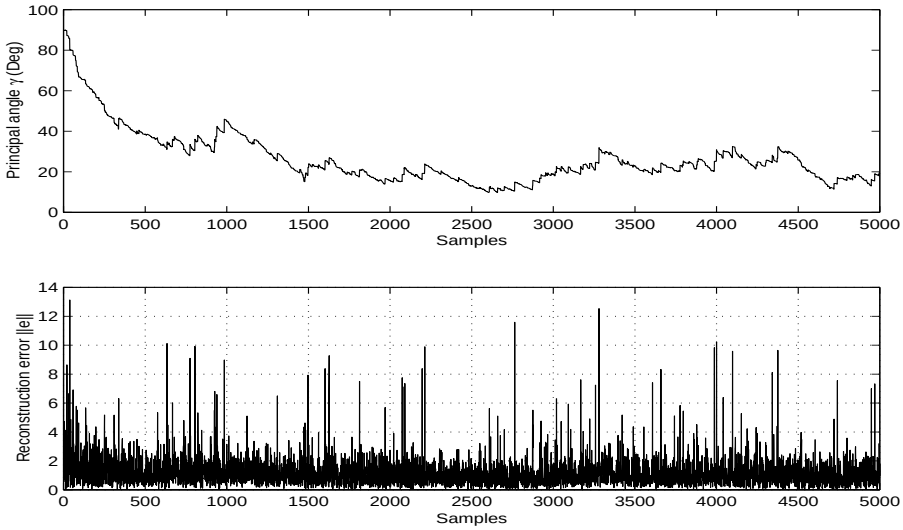

Figure 10: Numerical result obtained with algorithm (68), in the non-robust version ($\Phi(\cdot) = \Psi(\cdot) = 1$).

erature. In particular, extensions of previous research work and relationships were sought on Hebbian learning for linear, complex-weighted, feedforward and laterally-connected neural networks: Learning principles have been presented and analyses have been carried out on complex-valued principal/minor component/subspace quadratic/non-quadratic rules.

The starting point of the analysis was an optimization approach for complex-valued networks which aimed at producing a purposeful contribution. Many existing results could then be subsumed into this framework (although some may not). In summary, the following topics were analyzed throughout the paper:

- Neural networks architectures for principal subspace analysis have been considered in Section 2. After reviewing previous works that considered the complex-valued linear and real-valued non-linear case, the equations for the complex-valued non-linear case were derived using gradient optimization with Lagrange multipliers. Then, neural principal component analyzers for the real-valued linear and complex-valued linear case have been reviewed and extended to the complex-valued non-linear case, again using gradient based optimization. The problem of the choice of the non-linearity is mentioned and an attempt is made to generalize observations

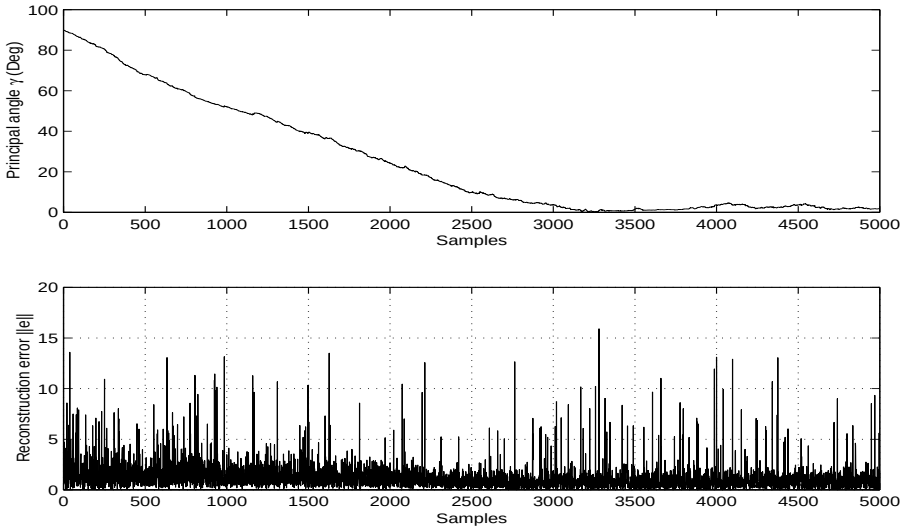

Figure 11: Numerical result obtained with algorithm (68), in the robust version ($\Phi(\cdot) = \Psi(\cdot) = \Psi_{\mathcal{C}}(\cdot)$).

of Sudjianto and Hassoun to the complex case. The analysis proceeded by extending some results from real-valued (linear) minor component analysis algorithms to the complex linear case.

- Minimization of reconstruction error is considered in Section 3. Again, existing principal subspace and principal component approaches for the real-valued non-linear and complex-valued linear cases have been reviewed and gradient equations for the complex-valued non-linear case have been derived. The problem of choosing the non-linearity is discussed based on a maximum likelihood approach by Song, Yilong and Feng for the real-valued non-linear case.

As a crucial part of the presented learning rules is the selection of a cost-function on which the derivation of the involved non-linearity is based, we made some choices concerning this selection. The main results on these topics were:

- We made the choice to consider only cost functions that take into account the magnitude of their complex argument. While this choice could seem natural, it was discussed and motivated with respect to its implicit assumptions through citations of relevant literature and explanation of its usefulness in applied fields related to signal processing.

- The treatment of the shape of the non-linearity seemed to provide insight that would go beyond the results known for non-linearities in the real-valued case. The discussion of the choice of the non-linearity and related developments in independent component analysis is an example of this finding.

We believe the manuscript can be useful for other Researchers because it presents an overview of existing approaches to complex-valued linear and real-valued nonlinear versions of Hebbian learning algorithms and provides derivations of equations that may be useful for the complex-valued non-linear case. It may be of interest for Readers who are interested in the extension of Hebbian learning to the complex and non-linear case.

Acknowledgments

The author wishes to gratefully thank the anonymous Reviewers and the Editor, Prof. Terrence Sejnowski, for the careful and accurate suggestions that helped improving the clarity of the presented material and the overall organization of the manuscript, and Leslie-Anne Chaden for the constant constructive support in handling the manuscript.

The final version of this work was completed when the author was a short-term visitor of the Mathematical Neuroscience Laboratory of the Brain Science Institute (BSI) at the Institute of Physical and Chemical Research RIKEN (Japan), in July-September 2004. The author wishes to express his gratitude to the BSI director, Prof. Shun-ichi Amari, and to the laboratory members for the kindest and warmest hospitality.

References

[1] H.M. Abbas and M.M. Fahmy, *Neural model for Karhunen-Loéve transform with application to adaptive image compression*, IEE Proceeding I, Communications, Speech and Vision, Vol. 140, No. 2, pp. 135 – 143, April 1994

[2] S.-I. Amari, *Neural theory of association and concept formation*, Biological Cybernetics, Vol. 26, pp. 175 – 185, 1977

[3] S.-I. Amari and A. Cichocki, *Adaptive blind signal processing – Neural network approaches*, Proceedings of the IEEE, Vol. 86, No. 10, pp. 2026 – 2048, 1998

[4] J.J. Atick and A.N. Redlich, *Convergent algorithm for sensory receptive field development*, Neural Computation, Vol. 5, No. 1, pp. 45 – 60, 1993

[5] P.F. Baldi and K. Hornik, *Learning in neural networks: A survey*, IEEE Transactions on Neural Networks, Vol. 6, No. 4, pp. 837 – 858, July 1995

[6] S. Bannour and M.R. Azimi-Sadjadi, *Principal component extraction using recursive least squares learning*, IEEE Transactions on Neural Networks, Vol. 6, No. 2, pp. 457 – 469, March 1995

[7] H.B. Barlow, *Unsupervised Learning*, Neural Computation, Vo. 1, pp. 295 – 311, 1989

[8] H.B. Barlow, *Guest editorial*, Perception, Vol. 27, pp. 885 – 888, 1998

[9] H.B. Barlow, *Redundancy Reduction Revisited*, Network: Computation in Neural Systems, Vol. 12, pp. 241 – 253, 2001

[10] H.B. Barlow and P. Földiák, *Adaptation and decorrelation in the cortex*, in *The Computing Neuron*, edited by C. Miall, R.M. Durbin and G.J. Mitchison, Addison-Wesley, Wokingham, England, pp. 54 – 72, 1989

[11] W. Bechtel and A. Abrahamsen, *Connectionism and the mind*, Oxford, UK: Blackwell, 1993

[12] A.J. Bell, T.J. Sejnowski, *An information maximisation approach to blind separation and blind deconvolution*, Neural Computation, Vol. 7, No. 6, pp. 1129 – 1159, 1995

[13] N. Benvenuto and F. Piazza, *On the complex back-propagation algorithm*, IEEE Transactions on Signal processing, Vol. 40, No. 4, pp. 967 – 969, 1992

[14] D.P. Bertsekas, *Constrained Optimization and Lagrange Multiplier Methods*, Athena Scientific (Massachusetts Institute of Technology), 1996

[15] E. Bingham and A. Hyvärinen, *A fast fixed-point algorithm for independent component analysis of complex valued signals*, International Journal of Neural Systems, Vol. 10, No. 1, pp. 1 – 8, February 2000

[16] E. Celledoni and S. Fiori, *Neural learning by geometric integration of reduced 'rigid-body' equations*, Journal of Computational and Applied Mathematics (JCAM). Accepted for publication

[17] Y. Chen and C. Hou, *High resolution adaptive bearing estimation using a complex-weighted neural network*, Proc. of International Conference on Acoustics, Speech and Signal Processing, Vol. II, pp. 317 – 320, 1992

[18] T.-P. Chen, S.-i. Amari and Q. Lin, *A unified algorithm for principal and minor components extraction*, Neural Networks, Vol. 11, No. 3, pp. 385 – 390, 1998

[19] T.-P. Chen and S.-i. Amari, *Unified stabilization approach to principal and minor components extraction algorithms*, Neural Networks, Vol. 14, No. 10, pp. 1377 – 1387, 2001

[20] T.W. Chow and Y. Fang, *Neural blind deconvolution of MIMO noisy channels*, IEEE Trans. on Circuits and Systems – Part I, Vol. 48, No. 1, pp. 116 – 120, 2001

[21] A. Cichocki and S.-i. Amari, *Adaptive Blind Signal and Image Processing*, J. Wiley & Sons, 2002

[22] S. Costa and S. Fiori, *Image compression using principal component neural networks*, Image and Vision Computing Journal (special issue on "Artificial Neural Network for Image Analysis and Computer Vision"), Vol. 19, No. 9-10, pp. 649 – 668, August 2001

[23] M.C.F. De Castro, F.C.C. De Castro, J.N. Amaral and P.R.G. Franco, *A complex valued Hebbian learning algorithm*, Proc. of International Joint Conference on Neural Networks, pp. 1235 – 1238, 1998

[24] G. Desodt and D. Muller, *Complex ICA applied to the separation of radar signals*, Proc. of Signal Processing V: Theories and Applications, Vol. I, pp. 665 – 668, 1990

[25] K.I. Diamantaras and S.Y. Kung, *Principal Component Neural Networks: Theory and Applications*, J. Wiley & Sons, 1996

[26] S.C. Douglas, S.-i. Amari and S.Y. Kung, *On gradient adaptation with unit-norm constraints*, IEEE Transactions on Signal Processing, Vol. 48, No. 6, pp. 1843 – 1847, June 2000

[27] S. Fiori and F. Piazza, *A general class of ψ–APEX PCA neural algorithms*, IEEE Transactions on Circuits and Systems – Part I, Vol. 47, No. 9, pp. 1394 – 1398, September 2000

[28] S. Fiori, *Blind separation of circularly distributed source signals by the neural extended APEX algorithm*, Neurocomputing, Vol. 34, No. 1-4, pp. 239 – 252, August 2000

[29] S. Fiori, *A theory for learning by weight flow on Stiefel-Grassman manifold*, Neural Computation, Vol. 13, No. 7, pp. 1625 – 1647, July 2001

[30] S. Fiori, *A theory for learning based on rigid bodies dynamics*, IEEE Trans. on Neural Networks, Vol. 13, No. 3, pp. 521 – 531, May 2002

[31] S. Fiori, *Extended Hebbian learning for blind separation of complex-valued sources*, IEEE Transactions on Circuits and Systems - Part II, Vol. 50, No. 4, pp. 195 – 202, April 2003

[32] S. Fiori, A. Faba, L. Albini, E. Cardelli and P. Burrascano, *Numerical modeling for the localization and the assessment of electromagnetic field sources*, IEEE Transactions on Magnetics, Vol. 39, No. 3, pp. 1638 – 1641, May 2003

[33] S. Fiori, *A neural minor component analysis approach to robust constrained beamforming*, IEE Proceedings – Vision, Image and Signal Processing, Vol. 150, No. 4, pp. 205 – 218, August 2003

[34] S. Fiori, *Neural ICA by 'maximum-mismatch' learning principle*, Neural Networks, Vol. 16, No. 8, pp. 1201 – 1221, Oct. 2003

[35] S. Fiori, *Fully-multiplicative orthogonal-group ICA neural algorithm*, Electronics Letters, Vol. 39, No. 24, pp. 1737 – 1738, November 2003

[36] S. Fiori, *Fixed-point neural independent component analysis algorithms on the orthogonal group*, Journal of Future Generation Computer Systems (Elsevier). Accepted for publication

[37] S. Fiori, *A fast fixed-point neural blind deconvolution algorithm*, IEEE Transactions on Neural Networks, Vol. 15, No. 2, pp. 455 – 459, March 2004

[38] P. Földiák, *Forming sparse representations by local anti-Hebbian learning*, Biological Cybernetics, Vol. 64, pp. 165 – 170, 1990

[39] C. Fyfe and D. MacDonald, *Epsilon-insensitive Hebbian learning*, Neurocomputing, Vol. 47, pp. 35 – 57, 2002

[40] K. Gao, M.O. Ahmed, and M.N. Swamy, *A constrained anti-Hebbian learning algorithm for total least-squares estimation with applications to adaptive FIR and IIR filtering*, IEEE Transactions on Circuits and Systems – Part II, Vol. 41, No. 11, pp. 718 – 729, November 1994

[41] G.M. Georgiou and C. Koutsougeras, *Complex-domain backpropagation*, IEEE Transactions on Circuits and Systems – Part II, Vol. 39, No. 5, pp. 330 – 334, 1992

[42] E. Hairer, C. Lubich and G. Wanner, *Geometric Numerical Integration*, Springer series in Computational Mathematics, Springer, 2002

[43] A.I. Hanna and D.P. Mandic, *A complex-valued nonlinear neural adaptive filter with a gradient adaptive amplitude of the activation function*, Neural Networks, Vol. 16, No. 2, pp. 155 – 159, March 2003

[44] G.F. Harpur, *Low entropy coding with unsupervised neural networks*, Ph.D. Thesis, Department of Engineering, University of Cambridge, February 1997

[45] S. Haykin, *Neural Networks*, MacMillan College Publishing Company, 1994

[46] S. Haykin, *An Introduction to Analog and Digital Communications*, Wiley Text Books. First edition (January 1989)

[47] D. Hebb, *The Organization of Behaviour*, John Wiley, New York, 1949

[48] U. Helmke and J.B. Moore, *Optimization and Dynamical Systems*, Springer-Verlag, Berlin, 1993

[49] I. Higuchi and S. Eguchi, *Robust principal component analysis with adaptive selection for tuning parameters*, Journal of Machine Learning Research, Vol. 5, pp. 453 – 471, May 2004

[50] A. Hyvärinen, *Fast and robust fixed-point algorithms for independent component analysis*, IEEE Transactions on Neural Networks, Vol. 10, No. 3, pp. 626 – 634, May 1999

[51] A. Hyvärinen, J. Karhunen and E. Oja, *Independent Component Analysis*, John Wiley & Sons, 2001

[52] J. Karhunen and J. Joutsensalo, *Representation and separation of signals using nonlinear PCA type learning*, Neural Networks, Vol. 7, No. 1, pp. 113 – 127, 1994

[53] J. Karhunen and J. Joutsensalo, *Generalizations of PCA, optimization problems, and neural networks*, Neural Networks, Vol. 8, No. 4, pp. 549 – 562, 1995

[54] J.M. Kates, *Superdirective arrays for hearing aids*, Journal of Acoustics Society of America, Vol. 94, No. 4, pp. 1930 – 1933, Oct. 1993

[55] R. Klemm, *Adaptive airborne MTI: An auxiliary channel approach*, IEE Proceedings F, Vol. 134, pp. 269 – 276, 1987

[56] S.Y. Kung, K.I. Diamantaras and J.S. Taur, *Adaptive principal component extraction (APEX) and applications*, IEEE Transactions on Signal Processing, Vol. 42, No. 5, pp. 1202 – 1217, May 1994

[57] A. Iserles, H.Z. Munthe-Kaas, S.P. Nørsett and A. Zanna: *Lie-group methods*, Acta Numerica, Vol. 9, pp. 215 – 365, 2000

[58] B. Laheld and J.F. Cardoso, *Adaptive source separation with uniform performance*, Signal Processing VII: Theories and Applications (EUSIPCO), Vol. 1, pp. 183 – 186, 1994

[59] D.S. Levine, V.R. Brown and T.V. Shirey (Ed.s), *Oscillations in Neural Systems*, The International Neural Networks Society Series, Lawrence Erlbaum Associates, London, 1999

[60] K. Liano, *Robust error measure for supervised neural network learning with outliers*, IEEE Transactions on Neural Networks, Vol. 7, No. 1, pp. 246 – 250, January 1996

[61] R. Linsker, *Local synaptic rules suffice to maximize mutual information in a linear network*, Neural Computation, Vol. 4, pp. 691 – 702, 1992

[62] F.-L. Luo and R. Unbehauen, *Applied Neural Networks for Signal Processing*, Cambridge University Press, Cambridge, 1997

[63] G. Mathew and V. Reddy, *Development and analysis of a neural network approach to Pisarenko's harmonic retrieval method*, IEEE Transactions on Signal Processing, Vol. 42, pp. 663 – 667, 1994

[64] H. Mathis and S.C. Douglas, *On the existence of universal nonlinearities for blind source separation*, IEEE Transactions on Signal Processing, Vol. 50, No. 5, pp. 1007 – 1016, May 2002

[65] Y. Miao and Y. Hua, *Fast subspace tracking and neural network learning by a novel information criterion*, IEEE Transactions on Signal Processing, Vol. 46, pp. 1967 – 1979, 1998

[66] R.B. Michaels and B.R. Upadhyaya, *A complex valued neural network with local learning laws*, Intelligent Engineering Systems through Artificial Neural Networks (C.H. Dagli et al., Ed.s), Vol. 9, pp. 101 – 109, ASME Press, New York, 1999

[67] K.D. Miller and D.J.C. MacKay, *The role of constraints in Hebbian learning*, Neural Computation, Vol. 6, No. 1, pp. 100 – 126, 1994

[68] M. Miyauchi, M. Seki, A. Watanabe and A. Miyauchi, *Interpretation of optical flow through complex neural network*, Proc. of International Conference on Artificial Neural Networks, pp. 645 – 650, 1993

[69] M.K. Müezzinoğlu, C. Güzelış and J.M. Zurada, *A new design method for the complex-valued multistate Hopfield associative memory*, IEEE Transactions on Neural Networks, Vol. 14, No. 4, pp. 891 – 899, July 2003

[70] T. Nitta, *An extension of the back-propagation algorithm to complex numbers*, Neural Networks, Vol. 10, pp. 1391 – 1415, 1997

[71] T. Nitta, *An analysis of the fundamental structure of complex-valued neurons*, Neural Processing Letters, Vol. 12 pp. 239 – 246, 2000

[72] T. Nitta, *Orthogonality of decision boundaries in complex-valued neural networks*, Neural Computation, Vol. 16, pp. 73 – 97, 2004

[73] E. Oja, *A simplified neuron model as a principal component analyzer*, Journal of Mathematics and Biology, Vol. 15, pp. 267 – 273, 1982

[74] E. Oja, *Neural networks, principal components, and subspaces*, International Journal of Neural System, Vol. 1, pp. 61 – 68, 1989

[75] E. Oja, *Principal components, minor components, and linear neural networks*, Neural Networks, Vol. 5, pp. 927 – 935, 1992

[76] E. Oja, *Beyond PCA: Statistical expansions by nonlinear neural networks*, International Conference on Artificial Neural Networks, Vol. 2, pp. 1049 – 1054, 1994

[77] F. Palmieri and J. Zhu, *Self-association and Hebbian learning in linear neural networks*, IEEE Transactions on Neural Networks, Vol. 6, No. 5, pp. 1165 – 1184, 1995

[78] R. Pandey, *Blind equalization and signal separation using neural networks*, Ph.D. Thesis, Indian Institute of Technology (IIT), Roorkee, India, 2001

[79] M.D. Plumbley, *Efficient information transfer and anti-Hebbian neural networks*, Neural Networks, Vol. 6, pp. 823 – 833, 1993

[80] M.D. Plumbley, *Algorithms for non-negative independent component analysis*, IEEE Transactions on Neural Networks, Vol. 14, No. 3, pp. 534 – 543, May 2003

[81] J. Rubner and P. Tavan, *A self-organizing network for principal component analysis*, Europhysics Letters, Vol. 10, No. 7, pp. 693 – 698, 1989

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

[83] R. Schmidt, *Multiple emitter location and signal parameter estimation*, IEEE Transactions on Antennas and Propagation, Vol. 34, pp. 276 – 280, 1986

[84] M.N. Shirazi, F. Peper and H. Sawai, *Principal component analysis by entropy-likelihood optimization*, Trans. of the Information Processing Society of Japan, Vol. 40, No. 10, pp. 3638 – 3644, October 1999

[85] M. W. Spratling and M.H. Johnson, *Preintegration lateral inhibition enhances unsupervised learning*, Neural Computation, Vol. 14, No. 9, pp. 2157 – 2179, 2002

[86] W. Song, L. Yilong and M. Feng, *An adaptive robust PCA neural network*, Proc. of International Joint Conference on Neural Networks, pp. 2288 – 2293, 1998

[87] A. Sudjianto and M.H. Hassoun, *Statistical basis of non-linear Hebbian learning and application to clustering*, Neural Networks, Vol. 8, No. 5, pp. 707 715, 1995

[88] N. Thirion Moreau and E. Moreau, *Generalized criterion for blind multivariate signal equalization*, IEEE Signal Processing Letters, Vol. 9, No. 2, pp. 72 – 74, 2002

[89] J.M. Vegas and P.J. Zufiria, *Generalized neural networks for spectral analysis: Dynamics and Liapunov functions*, Neural Networks, Vol. 17, No. 2, pp. 233 – 245, March 2004

[90] A. Weingessel and K. Hornik, *Local PCA algorithms*, IEEE Transactions on Neural Networks, Vol. 11, No. 6, pp. 1242 – 1250, November 2000

[91] B. Widrow and R. Winter, *Neural nets for adaptive filtering and adaptive pattern recognition*, IEEE Computer, pp. 25 – 39, March 1988

[92] L. Xu, *Least mean square error reconstruction principle for self-organizing neural-nets*, Neural Networks, Vol. 6, pp. 627 – 648, 1993

[93] L. Xu, E. Oja, and C.Y. Suen, *Modified Hebbian learning for curve and surface fitting*, Neural Networks, Vol. 5, pp. 441 – 457, 1992

[94] L. Xu and A.L. Yuille, *Robust PCA by self-organizing rules based on statistical physics approach*, IEEE Transactions Neural Networks, Vol. 6, No. 1, pp. 131 – 143, January 1995

[95] W.-Y. Yan, U. Helmke and J.B. Moore, *Global analysis of Oja's flow for neural networks*, IEEE Transactions on Neural Networks, Vol. 5, No. 5, pp. 674 – 683, September 1994

[96] B. Yang, *Projection approximation subspace tracking*, IEEE Transactions on Signal Processing, Vol. 43, No. 1, pp. 1247 – 1252, January 1995

[97] P.J. Zufiria, *On the discrete-time dynmics of the basic Hebbian neural network node*, IEEE Transactions on Neural Networks, Vol. 13, No. 6, pp. 1342 – 1352, 2002