Complex-Valued Neural Networks for Nonlinear Complex Principal Component Analysis

Sanjay S. P. Rattan and **William W. Hsieh**

Dept. of Earth and Ocean Sciences, University of British Columbia

Vancouver, B.C. V6T 1Z4, Canada

Resubmitted to Neural Networks,

June, 2004

Corresponding author: Prof. William Hsieh

Dept. of Earth and Ocean Sciences, University of British Columbia,

6339 Stores Road, Vancouver, B.C. V6T 1Z4, Canada.

E-mail: whsieh@eos.ubc.ca, tel: (604) 822-2821, fax: (604) 822-6088

Complex-Valued Neural Networks for Nonlinear Complex Principal Component Analysis

S.S.P. Rattan and W.W. Hsieh

Abstract

Principal component analysis (PCA) has been generalized to complex principal component analysis (CPCA), which has been widely applied to complex-valued data, 2-dimensional vector fields, and complexified real data through the Hilbert transform. Nonlinear PCA (NLPCA) can also be performed using auto-associative feed-forward neural network (NN) models, which allows the extraction of nonlinear features in the data set. This paper introduces a nonlinear complex PCA (NLCPCA) method, which allows nonlinear feature extraction and dimension reduction in complex-valued data sets. The NLCPCA uses the architecture of the NLPCA network, but with complex variables (including complex weight and bias parameters). The application of NLCPCA on test problems confirms its ability to extract nonlinear features missed by the CPCA. For similar number of model parameters, the NLCPCA captures more variance of a data set than the alternative real approach (i.e. replacing each complex variable by 2 real variables and applying NLPCA). The NLCPCA is also used to perform nonlinear Hilbert PCA (NLHPCA) on complexified real data. The NLHPCA applied to the tropical Pacific sea surface temperatures extracts the El Niño-Southern Oscillation signal better than the linear Hilbert PCA.

Keywords: Complex Principal Component Analysis, neural networks, Hilbert Transformation, El Niño

1 Introduction

The popular method of principal component analysis (PCA) (Jolliffe, 2002; Preisendorfer, 1988) has been generalized to complex principal component analysis (CPCA) (Horel, 1984). Like PCA, CPCA compresses the information into the fewest possible number of modes, as well as extracting the main features from a set of complex variables. CPCA is also commonly applied to 2-D real vector fields, such as the horizontal wind or ocean currents (Legler, 1983). Spectral PCA of real scalar variables is also performed by complexifying the real data via the Hilbert transform and then applying CPCA (Horel, 1984; Wallace and Dickison, 1972). Sometimes, this approach is used with all frequencies retained. The name Hilbert PCA is commonly used for this type of CPCA (Von Storch and Zwiers, 1999).

Despite the wide applications of the PCA and the CPCA techniques, they are limited to extracting only linear features from data sets. For real variables, there are now a number of ways to nonlinearly generalize PCA (Cherkassky and Mulier, 1998). A common way is to use auto-associative multi-layer perceptron neural network (NN) models (Kramer, 1991). This approach to nonlinear PCA (NLPCA) has been applied to a wide range of data sets including climate data (Monahan, 2001; Hsieh, 2001) with a recent review by Hsieh (2004). However, there does not appear to be any method developed for nonlinearly generalizing CPCA.

Real domain NN models are widely based on multi-layer perceptron networks which rely on backpropagation algorithms for nonlinear optimization (Bishop, 1995). The backpropagation algorithm has been extended to complex variables so that it can be used in a complex-valued NN (Georgiou and Koutsougeras, 1992; Leung and Simon, 1991). The complex-valued NNs have been shown to outperform their real counterparts in many ways, for example, in learning linear geometric transformations (Nitta, 1997).

In this paper, a complex-valued NN model is developed for nonlinear CPCA (NLCPCA). The model and implementation of NLCPCA are given in sections 2 and 3 respectively. The NLCPCA method is then applied to test problems in section 4. The NLCPCA is also extended to perform nonlinear Hilbert PCA in section 5 and applied in section 6 to analyze the tropical Pacific sea surface temperature data.

2 Model and Transfer Functions

2.1 Model

The most common way to perform PCA and CPCA is via singular value decomposition (SVD) (Strang, 1988) of the data matrix. Let the data matrix $\mathbf{Z} = \mathbf{X} + i\mathbf{Y}$ be a complex matrix with dimension $m \times n$ (where m = number of variables, n = number of observations, and the row mean of $\mathbf{Z}$ is zero). Without loss of generality, assume $m \leq n$ with $\mathbf{Z}$ having a rank of r (with $r \leq m$) (if $m > n$, one may apply the arguments below to the complex conjugate transpose of $\mathbf{Z}$). A CPCA model factors the data matrix $\mathbf{Z}$ into a set of orthonormal basis, say, a matrix $\mathbf{U}$ ($m \times r$) in which the columns are eigenvectors of $\mathbf{Z}\mathbf{Z}^{\mathrm{H}}$, a matrix $\mathbf{V}$ ($n \times r$) whose columns are eigenvectors of $\mathbf{Z}^{\mathrm{H}}\mathbf{Z}$ and a matrix $\mathbf{\Lambda}_r$ ($r \times r$) which is a real diagonal matrix containing the singular values $\lambda_1, ..., \lambda_r$, obtained from the square root of the r nonzero eigenvalues of both $\mathbf{Z}\mathbf{Z}^{\mathrm{H}}$ and $\mathbf{Z}^{\mathrm{H}}\mathbf{Z}$ (Strang, 1988). The resultant SVD of the data matrix is

$$\mathbf{Z} = \mathbf{U}\,\mathbf{\Lambda}_r\mathbf{V}^{\mathrm{H}}, \tag{1}$$

where $\mathbf{V}^{\mathrm{H}}$ is the complex conjugate transpose of $\mathbf{V}$. The jth column of $\mathbf{U}$ is also referred as the jth spatial pattern, loading, or empirical orthogonal function. The rank of $\mathbf{Z}$ is $\leq m$ because there are $m - r$ columns of $\mathbf{Z}$ which are linearly dependent. The jth row of $\mathbf{\Lambda}_r\mathbf{V}^{\mathrm{H}}$ gives the complex principal component (CPC) or score of the jth mode.

Since all the features explained by $\mathbf{Z}$ can be described by a subspace spanned by the r linearly independent columns of $\mathbf{V}$, there exists a transformation described by a complex function $\mathbf{G}$ which projects the r coordinates of the row subspace of $\mathbf{Z}$ given by $\mathbf{\Lambda}_r\mathbf{V}^{\mathrm{H}}$ ($r \times n$) back onto a matrix $\mathbf{Z}_{\mathrm{pred}}$ ($m \times n$) of predicted values:

$$\mathbf{Z}_{\mathrm{pred}} = \mathbf{G}(\mathbf{\Lambda}_r\mathbf{V}^{\mathrm{H}}). \tag{2}$$

For the CPCA, the transformation $\mathbf{G}(\mathbf{\Lambda}_r\mathbf{V}^{\mathrm{H}})$ yields r ($m \times n$) matrices of rank one corresponding to each eigenvector $\mathbf{u}_j$ and its associated CPC $\lambda_j\mathbf{v}_j^{\mathrm{H}}$:

$$\mathbf{Z}_{\text{pred}} = \sum_{j=1}^{r} \mathbf{u}_j \, \lambda_j \mathbf{v}_j^{\text{H}}, \tag{3}$$

where $\mathbf{u}_j$ and $\mathbf{v}_j$ are the jth columns of $\mathbf{U}$ and $\mathbf{V}$ respectively and the matrix $\mathbf{u}_j \lambda_j \mathbf{v}_j^{\text{H}}$ is the jth CPCA mode. The first CPCA mode explains the largest portion of variance in the data $\mathbf{Z}$, followed by the second CPCA mode, and eventually to mode r which explains the least. From (1), the mapping $\mathbf{G}$ (in the case of the CPCA) is given simply by the linear transformation $\mathbf{U}$.

The transformation $\mathbf{G}$ is also related to the least squares problem (Malthouse, 1998; Strang, 1988) since the idea is to find a minimum length solution between the predicted value $\mathbf{Z}_{\text{pred}}$ and $\mathbf{Z}$. This is achieved if the column space of the error matrix $\mathbf{Y} = \big(\mathbf{Z} - \mathbf{G}(\mathbf{\Lambda}_r \mathbf{V}^{\text{H}})\big)$ lies perpendicular to the column space of $\mathbf{G}(\mathbf{\Lambda}_r \mathbf{V}^{\text{H}})$. In the least squares sense, this is equivalent to minimizing the sum of the square of the errors via the objective function or cost function J:

$$J = \sum_{i=1}^{m} \sum_{j=1}^{n} \left| (\mathbf{Z})_{ij} - \big(\mathbf{G}(\mathbf{\Lambda}_r \mathbf{V}^{\text{H}})\big)_{ij} \right|^2 . \tag{4}$$

For CPCA, since the function $\mathbf{G}$ is linear, (4) is easily solved by (3) through the SVD technique (Strang, 1988). However, when $\mathbf{G}$ is nonlinear, (4) is used as it can be implemented via a neural network approach.

Kramer's (1991) auto-associative feedforward neural network structure adapted to the complex domain (Fig. 1) can be used to nonlinearly generalize CPCA. There are 3 hidden layers of neurons, with the first layer called the encoding layer, the second, the bottleneck layer (with a single complex neuron), and the third, the decoding layer. The network in Fig. 1 can be regarded as composed of 2 mappings: The first mapping $f : \mathbb{C}^m \rightarrow \mathbb{C}^1$ is represented by the network from the input layer to the bottleneck layer, with the bottleneck neuron giving the nonlinear CPC. The second mapping $\mathbf{g} : \mathbb{C}^1 \rightarrow \mathbb{C}^m$ is represented by the network from the bottleneck neuron to the output layer. This is the inverse mapping from the nonlinear CPC to the original data space. Dimension reduction is achieved by mapping the multi-dimensional input data through the bottleneck with a single complex degree of freedom. It is well known that a feed-forward NN only needs one layer of hidden neurons for it to model any continuous nonlinear function, provided enough hidden neurons are used (Bishop, 1995). For f, this hidden layer is provided by the encoding layer, while for $\mathbf{g}$, it is provided by the decoding layer. For the typical 1-hidden layer feed-forward NN,

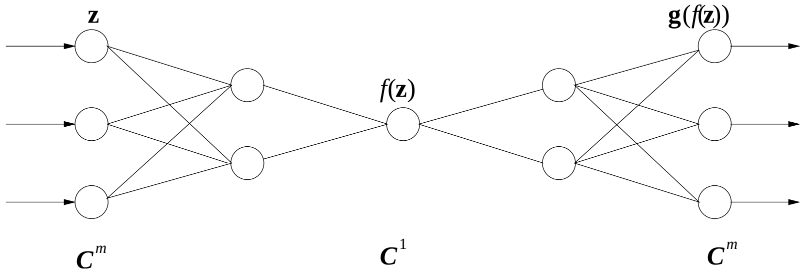

Figure 1: The complex-valued NN model for nonlinear complex PCA (NLCPCA) is an auto-associative feed-forward multi-layer perceptron model. There are m input and output neurons or nodes corresponding to the m variables. Sandwiched between the input and output layers are 3 hidden layers (starting with the encoding layer, then the bottleneck layer and finally the decoding layer) containing q, 1 and q neurons respectively. The network is composed of two parts: The first part from the input to the bottleneck maps the input $\mathbf{z}$ to the single nonlinear complex principal component (NLCPC) $f(\mathbf{z})$. The second part from the bottleneck to the output $\mathbf{z}'$ is the inverse mapping $\mathbf{g}(f(\mathbf{z}))$. For auto-associative networks, the target for the output neurons are simply the input data. Increasing the number of neurons in the encoding and decoding layers increases the nonlinear modelling capability of the network.

the transfer function from the input to the hidden layer is nonlinear, while the transfer function from the hidden layer to the output is usually linear (Bishop, 1995). Hence the 4 transfer functions from the input to the output in Fig. 1 are respectively nonlinear, linear, nonlinear and linear.

The NLCPCA model (Fig. 1) should reduce to CPCA if f and $\mathbf{g}$ are linear. The CPCA can be performed by a simpler neural network containing a single hidden layer (i.e. the bottleneck layer with one neuron) and linear transfer functions. The higher dimensional space (m) of the input is reduced linearly to a one-dimensional space at the bottleneck layer given by $f : \mathbb{C}^m \rightarrow \mathbb{C}^1$ and a linear inverse mapping $\mathbf{g} : \mathbb{C}^1 \rightarrow \mathbb{C}^m$ maps from bottleneck layer to the m-dimensional output $\mathbf{z}'$, such that the least squares error function

$$J = \sum_{j=1}^{n} \left\| \mathbf{z}_j - \mathbf{z}'_j \right\|^2 = \sum_{j=1}^{n} \left\| \mathbf{z}_j - \mathbf{g}\left(f\left(\mathbf{z}_j\right)\right) \right\|^2 \tag{5}$$

is a minimum (with $\mathbf{z}_j$ the jth column of $\mathbf{Z}$). For any input vector $\mathbf{z}$, the bottleneck neuron is given by:

$$f(\mathbf{z}) = \mathbf{w}^{\mathrm{H}}\mathbf{z}\,, \tag{6}$$

where $\mathbf{w}^{\mathrm{H}}$ is the weight vector between the inputs and the bottleneck layer.

In NLCPCA, the additional encoding and decoding layers (Fig. 1) allow the modelling of nonlinear continuous functions f and $\mathbf{g}$. The kth complex neuron t_{ki} at the ith layer is given by the neurons in the previous layer [the $(i-1)$th layer] via the transfer function σ_i with complex weights (w) and biases (b):

$$t_{ki} = \sigma_i \left(\sum_j w_{jki} t_{j(i-1)} + b_{ki} \right) ,$$

with $i = 1\,\text{to}\,4$ denoting, respectively, the encoding, bottleneck, decoding and output layers, (and $i = 0$, the input layer). A nonlinear transfer function (described in detail in the next section) is used at the encoding and decoding layers (σ_1 and σ_3), whereas σ_2 and σ_4 are linear (actually the identity function).

2.2 Transfer Functions

In the real domain, a common nonlinear transfer function is the hyperbolic tangent function, which is bounded between -1 and $+1$ and analytic everywhere. For a complex transfer function to be bounded and analytic everywhere, it has to be a constant function (Clarke, 1990), as Liouville's theorem (Saff and Snider, 2003) states that entire functions (functions that are analytic on the whole complex plane) which are bounded are always constants. The function $\tanh(z)$ in the complex domain has singularities at every $(\frac{1}{2} + l)\pi i$, $l \in \mathbb{N}$. Using functions like $\tanh(z)$ (without any constraint) leads to non-convergent solutions (Nitta, 1997). For this reason, early researchers (e.g. Georgiou and Koutsougeras, 1992) did not seriously consider such functions as suitable complex transfer functions.

Traditionally, the complex transfer functions that have been used focussed mainly on overcoming the unbounded nature of the analytic functions in the complex domain. Some complex transfer functions basically scaled the magnitude of the complex signals but preserved their arguments (i.e. phases) (Georgiou and Koutsougeras, 1992; Hirose, 1992), hence they are less effective in learning non-linear variations in the argument. A more traditional approach has been to use a "split" complex nonlinear transfer function (Nitta, 1997), where the real and imaginary components are used as separate real inputs for the transfer function. This approach avoids the unbounded nature of the nonlinear complex function but results in a nowhere analytic complex function, as the Cauchy-Riemann equations are not satisfied (Saff and Snider, 2003).

Recently, a set of elementary transfer functions has been proposed by Kim and Adali (2002) with the property of being *almost everywhere* (*a.e.*) bounded and analytic in the complex domain. The complex hyperbolic tangent $\tanh(z)$, is among them, provided the complex optimization is performed with certain constraints on z. If the magnitude of z is within a circle of radius $\frac{\pi}{2}$, then the singularities do not pose any problem, and the boundedness property is also satisfied. In reality, the dot product of the input and weight vectors may be $\geq \frac{\pi}{2}$. Thus a restriction on the magnitudes of the input and weights has to be considered.

For the NLCPCA model (Fig. 1), the magnitude of input data can be scaled (e.g. by dividing each element of the rth row of $\mathbf{Z}$ by the maximum magnitude of an element in that row, so each element of Z has magnitude ≤ 1). The weights at the first hidden layer are randomly initalized with small magnitude, thus limiting the magnitude of the dot product between the input vector and weight vector to be about 0.1, and a weight penalty term is added to the objective function J to restrict the weights to small magnitude during optimization. The weights at subsequent layers are also randomly initialized with small magnitude and penalized during optimization by the objective function

$$J = \sum_{j=1}^{n} \left\| \mathbf{z}_j - \mathbf{z}'_j \right\|^2 + p \left(\sum_{l=1}^{q} \left\| \mathbf{w}_l^{(1)} \right\|^2 + \left\| \mathbf{w}^{(2)} \right\|^2 + \sum_{l=1}^{q} \left\| \mathbf{w}_l^{(3)} \right\|^2 \right) , \tag{7}$$

where $\mathbf{w}^{(i)}$ denotes the weight vectors (including the bias parameters) from layers $i = 1, 2, 3$, and p is the weight penalty parameter.

3 Implementation

Since the objective function J is a real function with complex weights, the optimization of J is equivalent to finding the minimum gradient of J with respect to the real and the imaginary parts of the weights. All the weights (and biases) are combined into a single weight vector $\mathbf{s}$. Hence the gradient of the objective function with respect to the complex weights can be split into (Georgiou and Koutsougeras, 1992):

$$\frac{\partial J}{\partial \mathbf{s}} = \frac{\partial J}{\partial \mathbf{s}^{\mathrm{R}}} + i \frac{\partial J}{\partial \mathbf{s}^{\mathrm{I}}} \tag{8}$$

where $\mathbf{s}^{\mathrm{R}}$ and $\mathbf{s}^{\mathrm{I}}$ are the real and the imaginary components of the weight vector respectively. During optimization the real and the imaginary components of the weights were separated and kept in a single real weight vector while optimization was done by the MATLAB function "fminunc", using a quasi-Newton algorithm.

For the input data sets described later, the input variables were normalized by removing their mean and the real components were divided by the largest standard deviation among the real variables while the imaginary components were divided by the largest standard deviation among the imaginary components.

The number of hidden neurons, q, in the encoding/decoding layer of the NN model (Fig. 1) was varied from 2 to 10. Large values of q had smaller mean square errors (MSE) during training but led to overfitted solutions due to the excessive number of network parameters. Based on a general principle of parsimony, $q = 3$ to 6 was found to be an appropriate number for the NN in this study. Suitable values of the penalty parameter p ranged from 0.01 to 0.1. For $q = 6$, an ensemble of 25 randomly initialized neural networks were run. Also, 20% of the data was randomly selected as test data (also called validation data by Bishop, 1995) and withheld from the training of the NN. Runs where the MSE was larger for the test data set than for the training data set were rejected to avoid overfitted solutions. The NN with the smallest MSE over the test data was selected as the solution for the NLCPCA mode 1 and compared with the CPCA mode 1.

4 Testing NLCPCA on data sets

The NLCPCA is applied to a test problem with 3 complex variables:

$$z_1(t) = \frac{t}{\pi} + i\cos\left(\frac{\pi t}{2}\right), \quad z_2(t) = \frac{t}{\pi} + i\sin\left(\frac{\pi t}{2}\right), \quad z_3(t) = \frac{t\cos t}{3.2} + i\left(\frac{(t\cos t)^2}{5} - 1\right), \tag{9}$$

where t denotes a real value from $-\pi$ to π at increments of $\pi/100$. This noiseless data set with 200 samples is analyzed by the NLCPCA and CPCA, with the results shown in Figs. 2 and 3.

In Fig. 2 the features of the complex time series in all 3 variables have been more accurately captured by the NLCPCA mode 1 than by the CPCA mode 1. In all three variables, the magnitude and argument of the complex time series extracted by the NLCPCA mode 1 closely resemble the input signal, whereas the CPCA shows considerable difference. For example, for z_1 between the 10th to 50th time points, the

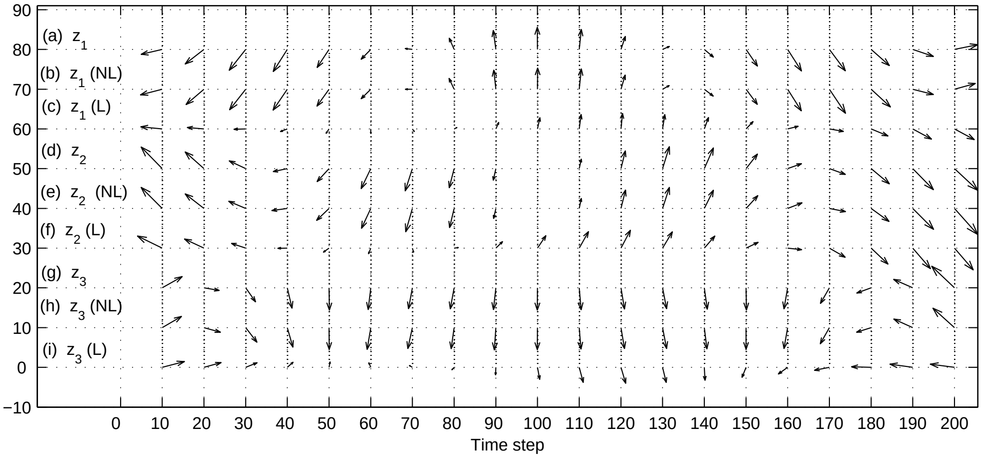

Figure 2: Plot of the complex time series (for every 10th data point) of z_1, z_2, z_3 and their predicted series by the NLCPCA mode 1 (NL) (with $q = 6$ and $p = 0.01$) and by the CPCA mode 1 (L). Each complex number is plotted as a vector in the complex plane. The time series have been vertically displaced by multiples of 10 for better visualization.

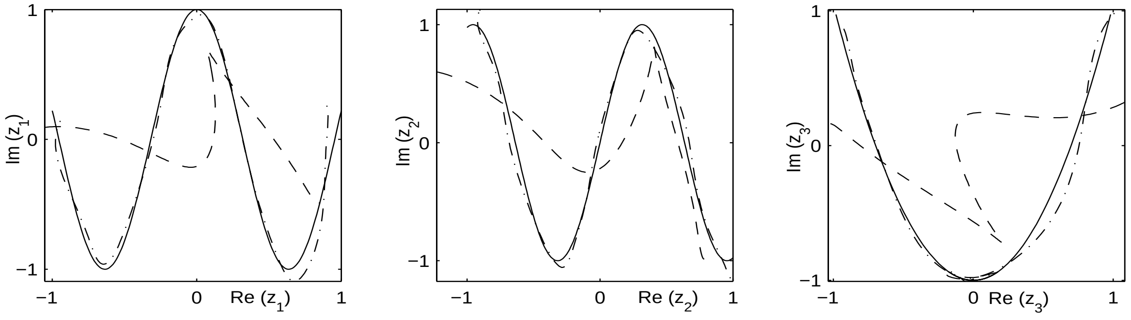

Figure 3: Complex plane (imaginary component versus real component) plots of the complex time series (a) $z_1(t)$ (b) $z_2(t)$ and (c) $z_3(t)$ (all in solid line). The NLCPCA mode 1 (dot-dash line) is also shown together with the CPCA mode 1 (dashed line).

CPCA mode 1 results are smaller in magnitude and are oriented more horizontally than the original signal. Fig. 3, a plot in the complex plane of the three variables, also shows clearly the close resemblance to the original signal by the NLCPCA mode 1 as opposed to the CPCA mode 1. The NLCPCA mode 1 accounts for 99.2% of the total variance, versus 53.6% for the CPCA mode 1. The root mean square error (RMSE) is 0.057 for NLCPCA mode 1, versus 0.631 for CPCA mode 1.

The performance of NLCPCA in a slightly noisy data set (with 10% Gaussian noise added to (9)) was studied next. Again the NLCPCA mode 1 was able to capture the essential features of the underlying signal in the noisy data set. The NLCPCA mode 1 explained 97.7% of the total variance of the noisy data set, in contrast to 53.3% by the CPCA mode 1. When the CPCA and NLCPCA modes recovered from the noisy data set were compared with the original noiseless data set, the RMSE of 0.1073 for the NLCPCA compares favorably with 0.633 for the CPCA. Relative to the noiseless data, the NLCPCA had correlation skills of 99% for all three variables (for both real and imaginary components), while CPCA had correlations ranging from 52-80%. The NLCPCA was further tested at higher noise levels, where the amount of Gaussian noise added was increased from 10% to 50% that of the signal, with the extracted mode remaining satisfactory.

What happens if one chooses to work with real variables? Each complex variable can be replaced by its real and imaginary components and NLPCA performed on the real data. Consider the following example where

$$z_1 = t, \qquad z_2 = t^2, \qquad z_3 = t^3 \,. \tag{10}$$

Unlike the previous example where t is a real variable, here t is a complex variable, where its real component is a Gaussian random variable with unit variance, and similarly for the imaginary component. The data is noiseless, with 200 samples. For this example, NLCPCA is compared with NLPCA with 2 bottleneck neurons and NLPCA with 1 bottleneck neuron. The complex bottleneck neuron in NLCPCA has 2 degrees of freedom, hence the NLPCA with 2 real bottleneck neurons is a more appropriate comparison. For similar number of model parameters, the NLCPCA was able to capture more variance than both versions of NLPCA in Table 1. This suggests that the NLCPCA is able to represent complex functions more efficiently than the NLPCA.

q	NLCPCA		2 bottleneck NLPCA		1 bottleneck NLPCA	
	% var.	# param.	% var.	param.	% var.	param.
2	62	48	56	29	30	24
3	80	68	58	41	43	34
4	100	88	68	53	49	44
5			70	65	49	54
6			81	79	52	64
7			84	89	54	74
8					63	84
9					66	94

Table 1: The % variance of the noiseless test data set explained by the first modes of the NLCPCA (with 1 complex bottleneck neuron), NLPCA with 2 bottleneck neurons, and NLPCA with 1 bottleneck neuron. The number of neurons in the encoding/decoding layer is q, and the total number of (real) parameters are also listed. For the NLCPCA, every complex parameter is counted as 2 real parameters.

5 NLCPCA as nonlinear Hilbert PCA

Another common application of CPCA is in Hilbert PCA, where a real data set is first complexified by a Hilbert transform, and then analyzed by CPCA (Horel, 1984; von Storch and Zwiers, 1999). Here we shall use NLCPCA to perform nonlinear Hilbert PCA (NLHPCA).

A Hilbert transformation complexifies a real time series $x(t)$ by adding an imaginary component $y(t)$, defined to be the original real time series phase-shifted by $\frac{\pi}{2}$ at each frequency ω, yielding $z(t) = x(t)+iy(t)$. Suppose $x(t)$ has the Fourier representation (von Storch and Zwiers, 1999)

$$x(t) = \sum_{\omega} a(\omega) e^{-2\pi i \omega t} \,. \tag{11}$$

Its Hilbert transform is

$$y(t) = x^{\mathrm{HT}}(t) = \sum_{\omega} a^{\mathrm{HT}}(\omega) e^{-2\pi i \omega t}, \tag{12}$$

with

$$a^{\mathrm{HT}}(\omega) = \begin{cases} ia(\omega) & \text{for} \quad \omega \leq 0 \\ -ia(\omega) & \text{for} \quad \omega > 0 \,. \end{cases}$$

For a simple test problem imagine there are three stations recording ocean waves coming towards the shore. The first station is far from the coast, so the wave $(x_1(t))$ looks sinusoidal in shape; the second measurement $(x_2(t))$ is closer to the shore, so the wave is steeper, with a tall, narrow crest and a shallow, broad trough; and the third one $(x_3(t))$ is closest to the shore, so the wave is even steeper due to strong nonlinear dynamics (Pond and Pickard, 1983). Let $\omega = \pi/12$, and

$$x_1(t) = \sin(\omega t) \quad x_2(t) = f_2(\omega t - \frac{\pi}{8}) \quad x_3(t) = f_3(\omega t - \frac{\pi}{2}), \tag{13}$$

where we use the idealized wave forms

$$f_2(\theta) = \begin{cases} 2 & \text{for} \quad 0 < \theta < \frac{2\pi}{3} \\ -1 & \text{for} \quad \frac{2\pi}{3} < \theta < 2\pi, \end{cases}$$

$$f_3(\theta) = \begin{cases} 3 & \text{for} \quad 0 < \theta < \frac{\pi}{2} \\ -1 & \text{for} \quad \frac{\pi}{2} < \theta < 2\pi. \end{cases}$$

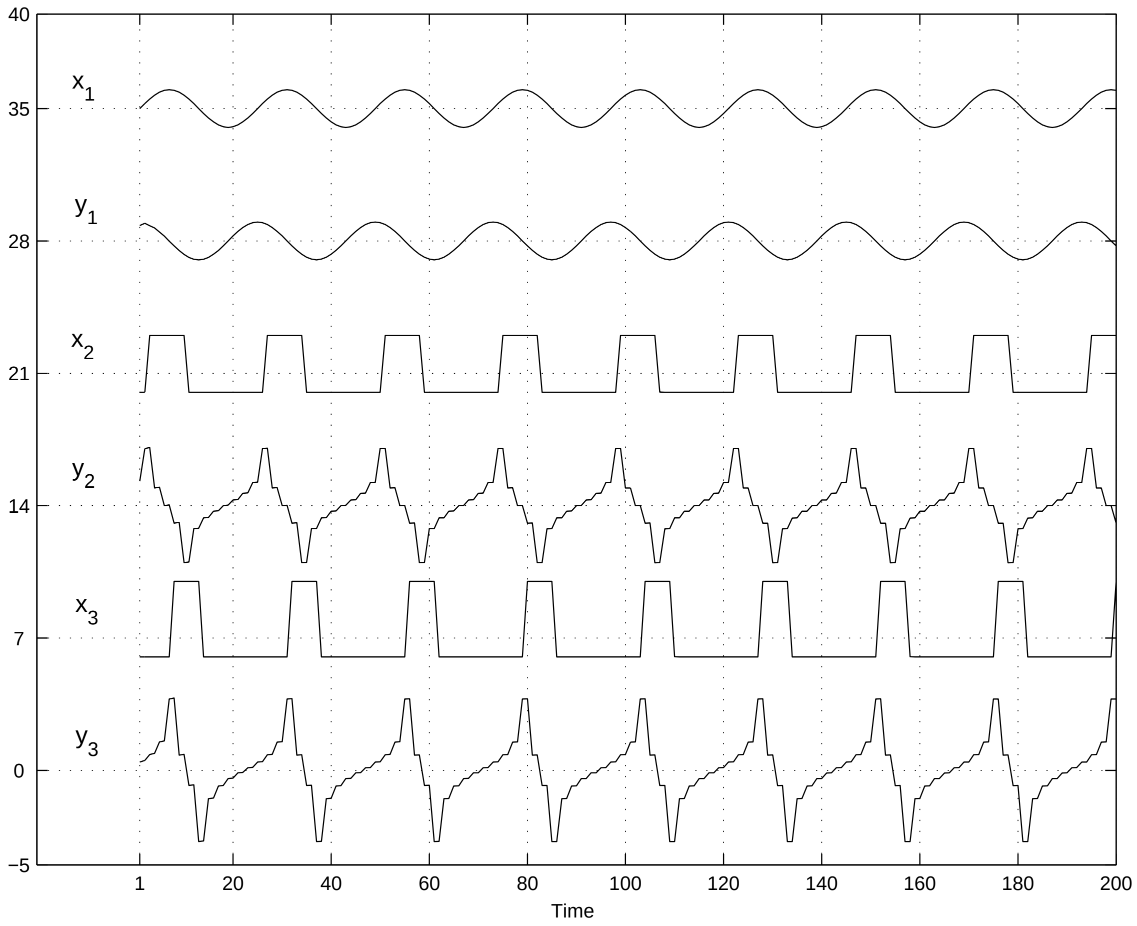

Figure 4: The real (**x**) and the imaginary (**y**) components of the complexified time series by Hilbert transform in the three dimensions (only first 200 points shown). The time series have been vertically displaced by multiples of 7 for better visualization.

The real time series together with their Hilbert transforms are shown in Fig. 4.

To each of the real time series of length 360 points, 10% Gaussian noise was added. The time series were then complexified via the Hilbert transform. These complex time series (Hilbert transforms) were then analyzed by NLCPCA and CPCA. The noisy time series and the extracted mode 1 solutions by NLCPCA and by CPCA are shown in Fig. 5. The NLCPCA mode 1 explained 97.7% of the total variance of the noisy data set, in contrast to the 85.4% explained by the CPCA mode 1. The correlations between the original noiseless signal and the retrieved signals by NLCPCA and CPCA mode 1 from the noisy data showed that the NLCPCA attained correlations of 99% for all three dimensions, while the CPCA correlations ranged from 85% to 95%. The RMSE between the mode 1 solution extracted from the noisy data and the original noiseless signal was 0.269 for the NLCPCA, versus 0.736 for the CPCA.

A further test was performed with the Gaussian noise increased from 10% to 50%. The NLCPCA mode 1 was found to explain 85.9% of the total variance of the noisy data set compared to 82.1% by the CPCA mode 1. The correlations between the original noiseless signal and the retrieved signals by NLCPCA and CPCA mode 1 from the noisy data showed that the NLCPCA correlations ranged from 79% to 89% whereas the CPCA mode 1 correlations were lower (75% to 86%). The RMSE between the mode 1 solution extracted from the noisy data and the original noiseless signal was 0.850 for the NLCPCA, versus 1.148 for the CPCA.

6 NLHPCA of tropical Pacific sea surface temperature data

To demonstrate the NLCPCA on an actual data set, we apply the technique as nonlinear Hilbert PCA to the tropical Pacific sea surface temperature (SST) data to capture the El Niño-Southern Oscillation (ENSO) signal. Centered in the tropical Pacific, the ENSO phenomenon is the largest interannual signal in the global climate system, irregularly producing warm episodes called El Niño and cool episodes called La Niña. This data set (for the period January, 1950 to December, 1999) came from NOAA (Smith et. al., 1996). The region studied is the whole tropical Pacific from 125^{o}E to 65^{o}W, 21^{o}S to 21^{o}N with a 2^o by 2^o grid. The climatological seasonal mean was removed from the raw data, followed by smoothing with a 3-month running mean.

The processed SST data matrix was then complexified using the Hilbert transform (12). CPCA was then performed on this matrix, with the first 3 CPCA modes accounting for 61.5%, 12.2% and 5.8%, respectively, of the total variance. In the complex plane of the first PC, the data scatter is oriented primarily along the real axis. The spatial pattern from the real part of the HPCA mode 1 when the real part of its PC is minimum (Fig. 6a) displays a cool SST pattern associated with the La Niña state. The spatial pattern when the real part of the PC is maximum (Fig. 6b) shows a warm SST pattern associated with El Niño. The spatial patterns are stationary, i.e. the patterns are the same between Figs. 6a and b

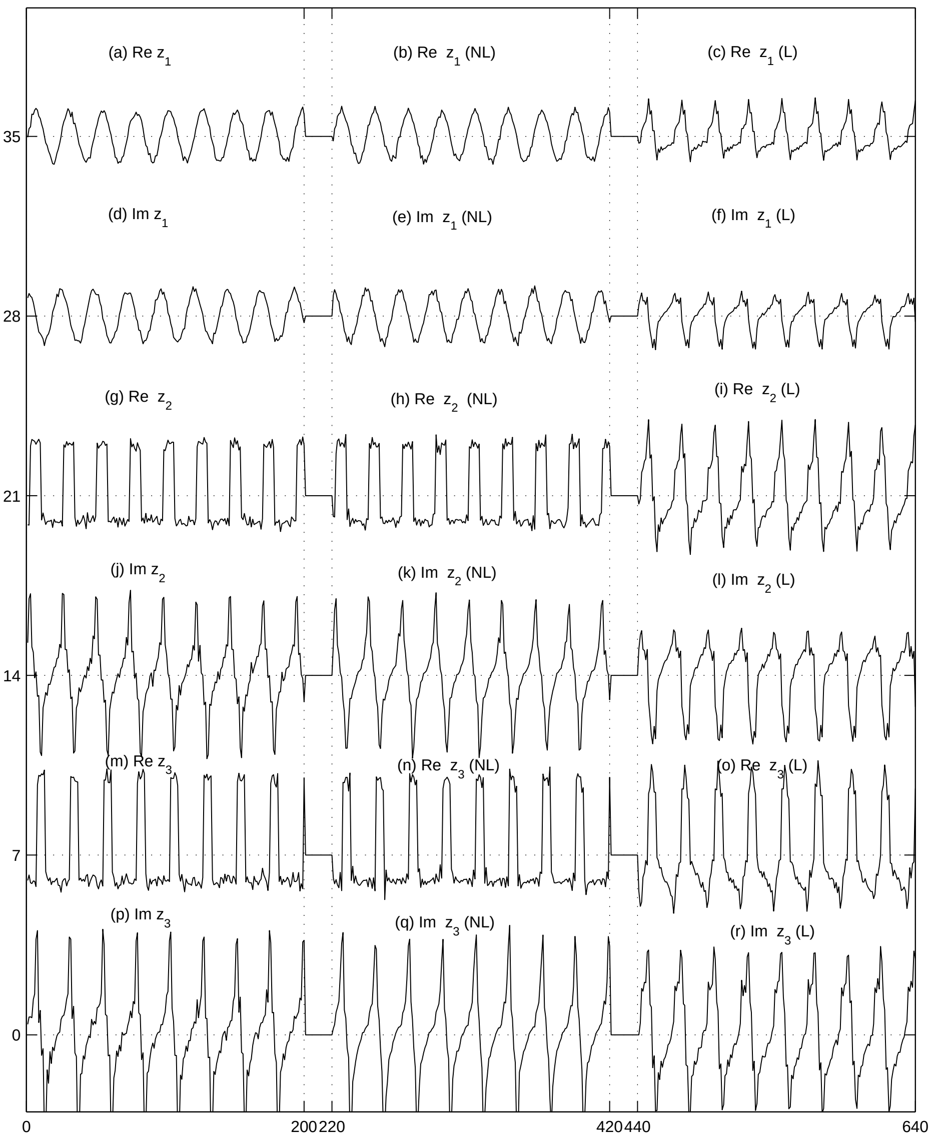

Figure 5: The real and the imaginary components of the noisy complex time series (200 points only) in the three dimensions z_1, z_2, z_3 and the corresponding predicted time series by NLCPCA mode 1 (NL) (with $q = 6$, and $p = 0.01$) and by the CPCA mode 1 (L). The time series have been vertically displaced by multiples of 7 for better visualization. The length of each time series is 200 points, with each component starting from 0 for the noisy data, from 220 for the NLCPCA result, and from 440 for the CPCA result. There is a horizontal gap of 20 points between them for better visualization.

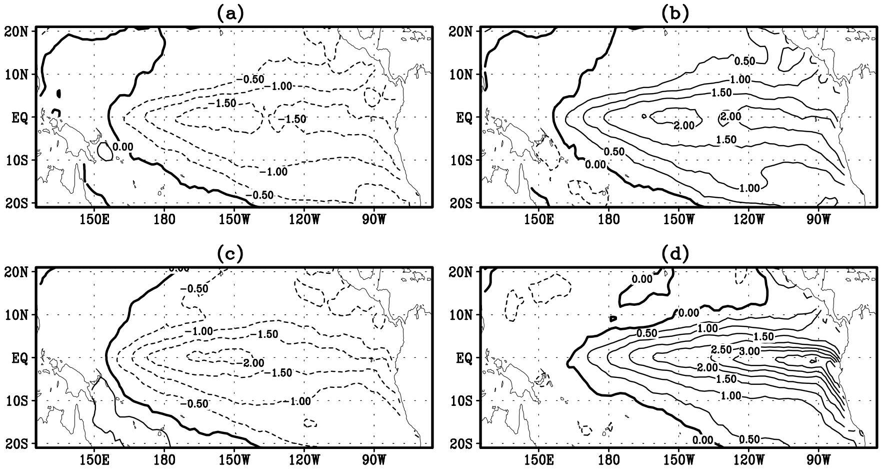

Figure 6: The spatial pattern of the tropical Pacific monthly sea surface temperature anomalies from the real part of the Hilbert PCA mode 1 when the real part of the PC is (a) minimum (corresponding to strong La Niña conditions) and (b) maximum (strong El Niño conditions), and from the real part of the NLHPCA mode 1 when the real part of the nonlinear PC is (c) minimum (La Niña) and (d) maximum (El Niño). The contours are in units of 0.5°C, with the positive contours shown as solid curves, negative contours as dashed curves, and the zero contour, a thick line.

except for a change in sign and overall magnitude.

The imaginary component of the Hilbert PCA mode 1 reveals basically the same spatial pattern (not shown) as the real component, but leading it in time by a phase of about $\frac{\pi}{2}$, which is not surprising since $y(t)$ was computed from $x(t)$ using (12) by applying a $\frac{\pi}{2}$ phase-shift at each frequency ω.

The NLHPCA mode 1 was extracted using the NLCPCA approach described earlier (with $p = 0.004$ and $q = 6$). The input data to the model were the 3 leading complex PCs from the Hilbert PCA , i.e. the Hilbert PCA was used as a pre-filter to reduce the number of input variables to three. The nonlinear mode 1 accounted for 63.6% of the variance versus 61.5% for the linear mode 1. The spatial pattern from the real part of the NLHPCA mode 1 when the real part of its nonlinear PC is minimum (Fig. 6c) displays the La Niña state, while the maximum reveals the El Niño (Fig. 6d) state. Unlike the patterns Figs. 6a and b for the linear mode 1, the patterns for the nonlinear mode 1 in Figs. 6c and d are not stationary— the El Niño warming is strongest in the eastern equatorial Pacific off the Peruvian coast, while the La Niña cooling is strongest in the central equatorial Pacific. This asymmetrical structure of the oscillation agrees well with observations. The strength of the El Niño warming and the La Niña cooling in the nonlinear mode are also more intense than in the linear mode, and again in better agreement with observations during strong El Niña and La Niña. The imaginary component of the NLHPCA mode 1 reveals basically the same spatial pattern (not shown) as the real component, but leading it by a phase of about $\frac{\pi}{2}$. Hence both the real and imaginary components of the NLHPCA mode 1 represent the same ENSO signal, but with a phase lag between the two.

7 Summary and Conclusion

PCA is widely used for dimension reduction and feature extraction. Its generalization to complex variables led to CPCA, which is also widely used when dealing with complex variables, 2-D vector fields (like winds or ocean currents), or complexified real data (via Hilbert transform). Both PCA and CPCA are linear methods, incapable of extracting nonlinear features in the data. To perform nonlinear PCA (NLPCA), Kramer (1991) used an auto-associative feed-forward neural network with 3 hidden layers. When a linear method such as PCA is applied to data with nonlinear structure, the nonlinear structure is scattered into numerous linear PCA modes — a confusing result which is largely alleviated when using NLPCA (Hsieh, 2004).

While complex NNs have already been developed for nonlinear regression problems, this paper extends complex NNs to the role of nonlinear CPCA (NLCPCA). The NLCPCA uses the basic architecture of the Kramer NLPCA model, but with complex variables (including complex weights and biases). Nonlinear transfer functions like the hyperbolic tangent can be used, though the argument of the tanh function in the complex plane must have its magnitude within a circle of radius $\frac{\pi}{2}$ to avoid the singularities of the tanh function. This is satisfied by initializing with weights (including biases) of small magnitudes, and using weight penalty in the objective function during optimization. The NLCPCA code (written in MATLAB) is downloadable from http://www.ocgy.ubc.ca/projects/clim.pred/download.html.

Application of the NLCPCA on test data sets shows that NLCPCA has better skills compared to the CPCA method: NLCPCA explains more variance, and the features extracted by NLCPCA are also much closer to the underlying signal (in terms of correlation and root mean square error). For similar number of model parameters, the NLCPCA explains more variance than NLPCA (with either 1 or 2 bottleneck neurons), where each complex variable has been replaced by 2 real variables before applying the NLPCA.

In Hilbert PCA, CPCA is applied to real data complexified by the Hilbert transform. The NLCPCA has also been used to perform nonlinear Hilbert PCA (NLHPCA). When applied to the tropical Pacific sea surface temperatures, the NLHPCA mode 1 extracted the ENSO signal, fully characterizing its magnitude and the asymmetry between El Niño and La Niña states.

8 Acknowledgements

This work is supported by the Natural Sciences and Engineering Research Council of Canada and the Canadian Foundation for Climate and Atmospheric Sciences.

References

[1] Bishop, C. M. (1995). *Neural networks for pattern recognition.* Oxford: Oxford University Press.

[2] Cherkassky, V. and Mulier, F. (1998). *Learning from Data.* New York: Wiley, 464 pp.

[3] Clarke, T. (1990). Generalization of neural network to the complex plane. *Proceedings of International Joint Conference on Neural Networks, 2*, 435-440.

[4] Georgiou, G. and Koutsougeras, C. (1992). Complex domain backpropagation. *IEEE Transactions on Circuits and Systems II, 39*, 330-334.

[5] Hirose, A. (1992). Continuous complex-valued backpropagation learning. *Electronic Letters, 28*, 1854-1855.

[6] Horel, J.D. (1984). Complex principal component analysis: theory and examples. *Journal of Climate & Applied Meteorology, 23*, 1660-1673.

[7] Hsieh, W.W. (2001). Nonlinear principal component analysis using neural networks. *Tellus, 53A*, 599-615.

[8] Hsieh, W. W. (2004). Nonlinear multivariate and time series analysis by neural network methods. *Reviews of Geophysics, 42*, RG1003, doi:10.1029/2002RG000112.

[9] Jolliffe, I. T. (2002). *Principal component analysis.* Berlin: Springer, 502 pp.

[10] Kim, T. and Adali, T. (2002). Fully complex multi-layer perceptron network for nonlinear signal processing. *Journal of VLSI Signal Processing, 32*, 29-43.

[11] Kramer, M. A. (1991). Nonlinear principal component analysis using autoassociative neural networks. *AIChE Journal, 37*, 233-243.

[12] Legler, D. M. (1983). Empirical orthogonal function analysis of wind vectors over the tropical Pacific region. *American Meteorological Society, 64*, 234-241.

[13] Leung, H. and Simon, H. (1991). The complex backpropagation algorithm. *IEE Transactions on Signal Processing, 39*, 2101-2104.

[14] Monahan, A. H. (2001). Nonlinear principal component analysis: tropical Indo-Pacific sea surface temperature and sea level pressure. *Journal of Climate, 14*, 219-233.

[15] Nitta, T. (1997). An extension of the back-propagation algorithm to complex numbers, *Neural Networks, 10*, 1391-1415.

[16] Pond, S. and Pickard, G.L. (1983). *Introductory Dynamical Oceanography.* Pergamon, 329 pp.

[17] Preisendorfer, R.W. (1998). *Principal component analysis in meteorology and oceanography.* Elsevier, 425 pp.

[18] Saff, E. B. and Snider, A. D. (2003) *Fundamentals of complex analysis with applications to engineering and science*. Englewood Cliffs, N.J.

[19] Smith, T. M., Reynolds, R. W., Livezey, R. E. and Stokes, D. C. (1996). Reconstruction of historical sea surface using empirical orthogonal functions. *Journal of Climate, 14*, 1403-1420.

[20] Strang, G. (1988). *Linear algebra and its applications.* San Diego: Harcourt, Brace, Jovanovich Publishers.

[21] Von Storch, H. and Zwiers, F. W. (1999). *Statistical analysis in climate research.* London: Cambridge University Press.

[22] Wallace, J. M. and Dickinson R. E. (1972). Empirical orthogonal representation of time series in the frequency domain. Part I: theoretical considerations, *Journal of Applied Meteorology, 11*, 887-892.