Chapter 6
Nonlinear Method of Reduction of Dimensionality Based on Artificial Neural Network and Hardware Implementation

J.R.G. Braga, V.C. Gomes, E.H. Shiguemori, H.F.C. Velho, A. Plaza, and J. Plaza

6.1 Introduction

The technological development of imaging sensors of high spectral resolution, called multi- or hyper-spectral sensors, enables the acquisition of information on dozens up to thousand of spectral bands. Due to the large amount of available information, the reduction of the dimensions for the provided data, without loss of information, is a challenge [Ch13].

There are several schemes for reducing the dimensionality of data, one of them is the Principal Component Analysis (PCA) [An09]. Such analysis deals with linear transformation, and this limitation can influence the data classification for hyper-spectral sensors [LiEtAl12]. This is a motivation to study of nonlinear techniques for data reduction. One of such techniques is the Nonlinear Principal Component Analysis (NL-PCA), based on artificial neural networks (ANN) [LiEtAl12, DeLiDu09].

In this chapter, a multi-layer perceptron ANN [SiEtAl13] classifier, with back propagation for training, is employed. The general procedure to configure an ANN is an empirical one, where the ANN architecture is defined by an expert. Here, a self-configuring strategy is applied, where the optimal NN architecture is obtained

J.R.G. Braga (✉) • H.F.C. Velho
National Institute for Space Research, Av. dos Astronautas 1758,
São José dos Campos, SP, Brazil
e-mail: jgarciabraga@gmail.com; haroldo@lac.inpe.br

V.C. Gomes • E.H. Shiguemori
Department of Science and Aerospace Technology, São José dos Campos, SP, Brazil
e-mail: vcconrado@gmail.com; elcio@ieav.cta.br

A. Plaza • J. Plaza
University of Extremadura, Av. de la Universidad s/n, 10003 Cáceres, Spain
e-mail: aplaza@unex.es; jplaza@unex.es

C. Constanda, A. Kirsch (eds.), *Integral Methods in Science and Engineering*,
DOI 10.1007/978-3-319-16727-5_6

by solving an optimization problem. A new metaheuristic, named Multi-particle Collision Algorithm (MPCA) [LuBeVe08], is used to compute the minimum value for the objective function.

Finally, all optimal MLP-NNs are implemented on a hardware component: Field Programmable Gate Arrays. The use of hardware divide allows a fast parallel image processing with low energy demand.

6.2 Methodology

Figure 6.1 exhibits the methodology followed in this study.

6.2.1 Principal Component Analysis

The Principal Component Analysis (PCA) can be used for data reduction by eliminating less representative information [GoWo00]. The PCA reduction is based on selecting a smaller data set, but with almost the same variance from the original data. An algorithm for finding the principal components from a data set is expressed below:

1. Given a data set with n vectors with dimension m;

$$x_1 = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_m \end{bmatrix} \dots x_n = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_m \end{bmatrix}.$$

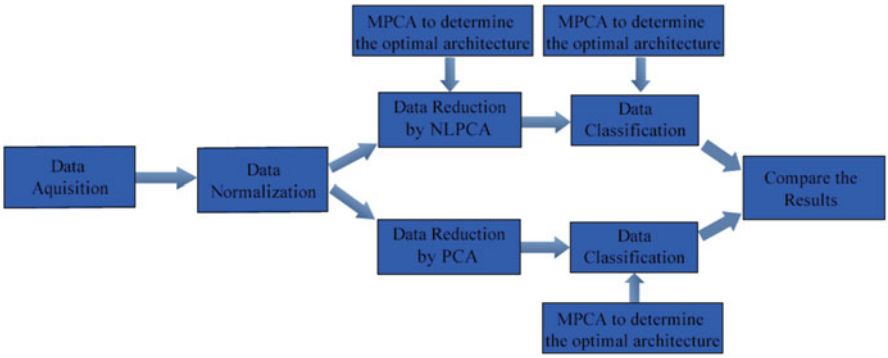

Fig. 6.1 Flowchart of the methodology used.

2. From these vectors, calculate the average μ_x.
3. Compute a new vector from the data set: $v_i = x_i - \mu_x$ $(i = 1, 2, 3, \ldots, n)$.
4. Multiplying the vector v_i by its transpose: $A_i = v_i \times v_i)^T$.
5. Covariance matrix: perform the sum of matrices above and divide by n:

$$M_{\text{cov}} = \frac{1}{n} \sum_{i=1}^{n} A_i \ . \tag{6.1}$$

6. Compute all the eigenvalues and eigenvectors of the covariance matrix (the QR method, or even the deflation technique [An09]). The eigenvector set consists of the principal components from the data set.
7. For generating the data set with reduced size, determine a reference eigenvalue. After that, consider only the reduced matrix containing the maximum eigenvalue (in module) up to the reference eigenvalue.

6.2.2 Artificial Neural Network

The Artificial Neural Network (ANN) is a machine designed to emulate the human brain [Ha01], where:

(a) knowledge is acquired through a learning process;
(b) the basic unit of operation is the artificial neuron;
(c) connections among neurons, called synapses, store the acquired knowledge.

The ANN is usually implemented using electronic components, it can be simulated by programming in a digital computer. The output of an ANN is given by

$$y_k = \varphi(v_k) \quad (k = 1, 2, \ldots, m) \tag{6.2}$$

where $\varphi(.)$ is the activation function, and v_k is a linear combination of all inputs x_j $(j = 1, 2, ..., n)$ multiplied by their respective synaptic weight w_{kj}. The activation function is the nonlinear component for this mapping. Heaviside, sigmoid, hyperbolic tangent functions are usually used as an activation function in an artificial neuron.

Figure 6.2 displays a representation for an artificial neuron. More than one hidden layer can be employed to define an ANN. A very popular topology for ANN is the multi-layer perceptron (MLP). Figure 6.2 shows a schematic of artificial neuron.

The most popular algorithm to determine the connection weights (learning phase) is the error back-propagation algorithm [RuHiWi86]. The latter algorithm is an example of supervised learning by error correction [Ha01]. Such learning algorithm can be divided into two distinct steps:

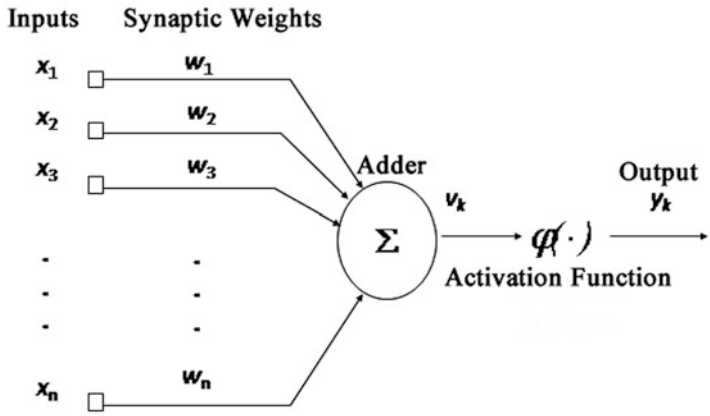

Fig. 6.2 Representation for an artificial neuron.

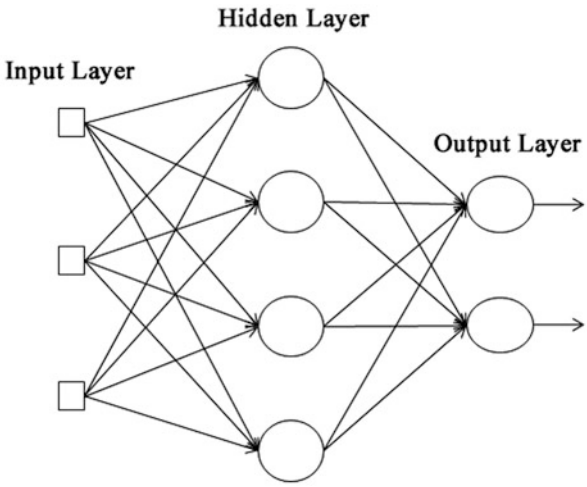

Fig. 6.3 Multi-layer perceptron ANN architecture.

(i) Each input pattern produces a response (output), and a value of error is obtained by comparison with the target set.
(ii) The weights are updated from the calculated error.

6.2.3 Self-Associative Artificial Neural Network

Consider a fully connected MLP neural network with three hidden layers—see Figure 6.4. The purpose of such NN is to produce an output identical to the input data [DeLiDu09, LiEtAl12].

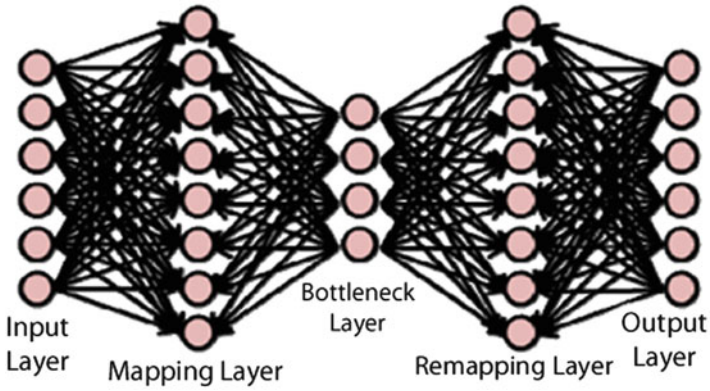

Fig. 6.4 Outline for a self-associative ANN.

The ANN with the architecture showed in Figure 6.4 is the operator of Nonlinear Principal Component Analysis (NL-PCA). The numbers of neurons in the input and output layers are the same. The mapping layer and re-mapping layer have a sufficient amount of neurons (equal). The output of this ANN is an approximation of the input data. The bottleneck layer has a (much) smaller amount of neurons than the other hidden layers. This is a nonlinear representation of the input data with dimension reduction. For practical purposes, we will not be dealing with raw input data, but with data emerging from the bottleneck layer [DeLiDu09].

6.2.4 Multi-Particle Collision Algorithm

Artificial neural networks have huge success in many applications. However, a tedious job that requires participation of an expert is the configuration of a neural network. Here, the problem of finding an optimal configuration for the neural network is formulated as an optimization problem, where the objective function is expressed as:

$$J(z) = penalty \times \left(\frac{\rho_1 \times E_{\text{train}} + \rho_2 \times E_{\text{gen}}}{\rho_1 + \rho_2} \right) \tag{6.3}$$

where $\rho_1 = 1$ and $\rho_2 = 0.1$ are the same values proposed by [CaRaCh11], which are adjustment factors that magnify the relevance attributed to the training error E_{train} (see Eq. 6.4), and generalization error E_{gen} (see Eq. 6.5), respectively [CaRaCh11].

$$E_{\text{train}} = \frac{1}{N}\sum_{k=1}^{N}(y_k - \hat{y}_k)^2 \tag{6.4}$$

$$E_{\text{gen}} = \frac{1}{M-(N+1)}\sum_{k=N+1}^{M}(y_k - \hat{y}_k)^2 \tag{6.5}$$

with y_k and $\hat{y}_k$ being the ANN output and the target value, respectively. The unknown vector z has 5 entries: # hidden layers (max = 3), # neurons for each hidden layer (max = 32), the learning ratio, momentum parameter – both used during the training phase, and type of activation function (only three: logarithmic, sigmoid, and hyperbolic tangent).

The *penalty* term is used to look for a simpler ANN, with the smallest number of neurons and the fastest convergence for calculating the connection weights. However, the *penalty* term will not be used in our applications.

The minimum for the objective function $J(z)$ (Eq. 6.3) is computed by the Multi-Particle Collision Algorithm (MPCA) [LuBeVe08], based on Particle Collision Algorithm [SaOl86]. The MPCA was modified by [AnEtAl14] to find the best value for objective function 6.3.

6.3 Results

The data set used was obtained from the Institute of Advanced Studies (IEAv, Brazil) and covers an area from São José dos Campos (SP), Brazil. Images were acquired by air-transported Hyperspectral Scanner Sensor (HSS), with 37 bands from the electromagnetic spectrum into range $[0.44\ \mu m,\ 4\ \mu m]$. The image spatial resolution for the HSS sensor ranging between 2 and 9 meters [Ca03]. There is a ground truth of 4 regions of interest used to evaluate the land classification. Figure 6.5 shows a region on São José dos Campos area, with the 4 regions of interest and the respective ground truth.

An MLP-NN is used as an image classifier. For self-configuring the NN, 105 images were selected, where each region is represented by the average of the pixels in a 3×3 matrix for each band. The data was split into three sets: training set, validation set, and testing set – see Table 6.1.

Before the data reduction by PCA or NL-PCA, and classification by MLP-NN, the pixels (radiance) were normalized:

$$p_N = \frac{p - p_{\text{Min}}}{p_{\text{Max}} - p_{\text{Min}}} \tag{6.6}$$

where p is the raw (pixel) data, p_N is the normalized pixel value, and p_{Min} and p_{Max} are the lowest the largest pixel values found in the data set.

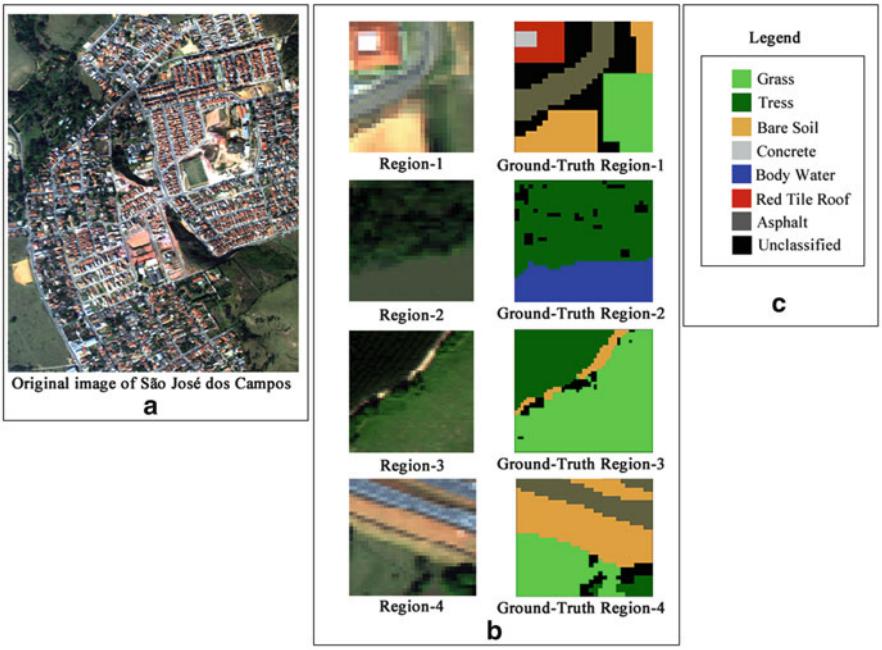

Fig. 6.5 (A) Original image obtained by HSS sensor, (B1) left: 4 regions of interest, right: the ground-truth for 4 cited regions, and (C) the legend for the classes.

Table 6.1 Data organization for training and testing the MLP-NN.

Numbers of patterns to training	55
Numbers of patterns to validation	15
Numbers of patterns to test	35

After the data normalization, the PCA method was applied. Only 6 principal components represent 99% the variability of the data. The MPCA was employed to find the optimal configuration of the MLP-NN to promote the data reduction by NL-PCA method. The MPCA was also used to determine the optimal architecture for the neural classifier.

Three ANNs were designed: (A) for performing the NL-PCA, (B) neural classifier with input data from standard PCA, and (C) neural classifier with input data from NL-PCA. The optimal configuration obtained with MPCA meta-heuristic is shown in Table 6.2.

To evaluate the data classification, the κ-index was used. The κ-index is a measure to quantify the deviation (classified data) from the exact values. The evaluation results for classification of κ-index average for 4 regions, overall accuracy, and average accuracy are shown in Table 6.3.

Table 6.2 Optimal architectures of the MLP found by MPCA.

	Configuration A	Configuration B	Configuration C
Input Layer	37	6	7
First Hidden Layer	25	25	20
Second Hidden Layer	7	—	—
Third Hidden Layer	25	—	—
Output Layer	37	3	3
Activation Function	Hyperbolic Tangent	Hyperbolic Tangent	Hyperbolic Tangent
Learning Rate	0.05	0.25	0.22
Momemtum	0.9	0.83	0.87

Table 6.3 Evaluation results obtained from classification by PCA of 4 regions using HSS sensor.

	Total Accuracy	Average Accuracy	κ-Index
NL-PCA + Classification	68.58%	61.61%	0.55
PCA + Classification	67.86%	65.55%	0.59

6.3.1 Execution of NLPCA in Hardware

The data reduction by the NL-PCA method was also implemented on the hardware device Xilinx Virtex II Pro FPGA. The VHDL[1] was used to configure the FPGA. The Cray XD1 hybrid computer system has 6 interconnected processing nodes, where each node has 2 AMD processors (CPU) and one FPGA.

If an FPGA is configured as an artificial neural network, the device can be identified as a *neuro-computer*. The implementation of the MLP-NN on FPGA has four different modules: (a) the MAC (**M**ultiplier **A**nd **A**ccumulator): designed to do the product between inputs and weights (or bias); (b) artificial neuron: using the MAC and control structures; (c) combination of neurons: the inputs are connected by a single bus; (d)LUT (**L**ook**U**p **T**able) unit: the neurons can receive data, and the results (outputs) are flowing to the LUT unit – defined to address 524,288 values of activation function. Finally, the layers can be concatenated in series forming the MLP-NN. Figure 6.6 shows all implemented components of our neuro-computer.

The activation phase of the optimal architecture for both MLP-NNs, one used as NL-PCA operator and another one to perform the classification, was implemented in the FPGA. A comparison between the results produced by the implementation of software and hardware is performed, the 4 regions used to evaluate the classification in software were used to evaluate the classification in hardware. Table 6.4 shows the average of result of κ-index, overall accuracy, and average accuracy of 4 regions performed in FPGA.

[1] VHDL: **V**HSIC (Very High Speed Integrated Circuits) **H**ardware **D**escription **L**anguage.

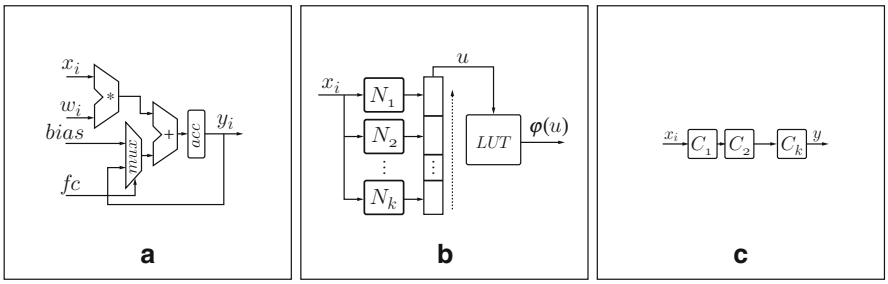

Fig. 6.6 Implementation of MLP-NN on FPGA: (A) MAC unit, (B) pipeline for LUT unit, (C) the MLP implemented in FPGA.

Table 6.4 Results obtained for classification by means of PCA with 4 HSS sensor regions.

	Total Accuracy	Average Accuracy	κ-Index
NLPCA + Classification in Hardware	68.58%	61.61%	0.55

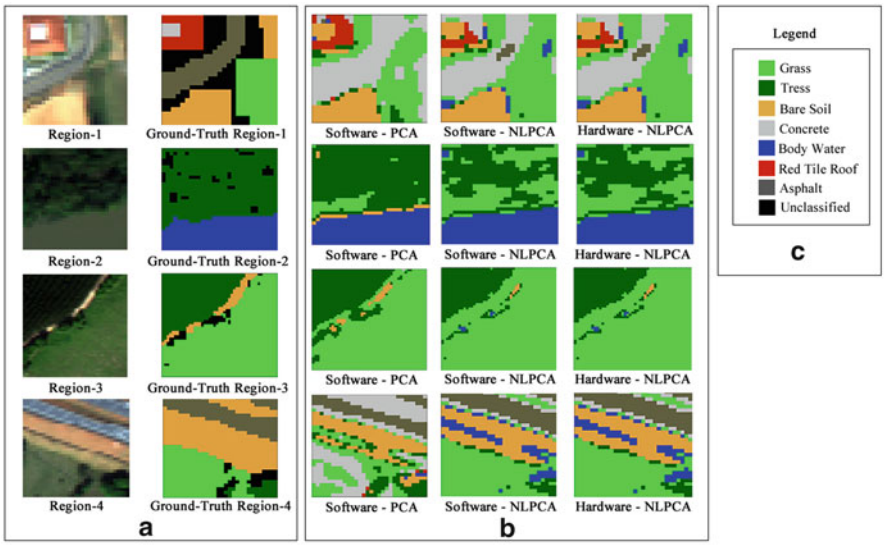

Fig. 6.7 Neural classifier for 4 regions with data reduction: PCA and NL-PCA on software, and NL-PCA on FPGA.

Figure 6.7 displays the classification results by using data reduction considering two strategies: PCA + NN-classifier (software), and NL-PCA + NN-classifier (software), and NL-PCA + NN-classifier (hardware = FPGA). The results express a good performance of NL-PCA for data reduction. The MLP-NN implementation on FPGA produced very good results in comparison with the software implementation.

6.4 Conclusions

A case study was presented to evaluate the NL-PCA method, a nonlinear scheme for reducing data dimensionality. The NL-PCA utilizes a self-associative MLP-NN as the reduction operator. The standard procedure by using PCA was also used for comparison. The data reduction was employed to deal with image processing with data from multi- or hyper-spectral sensors. In our context, image processing means image classification. An artificial neural network was designed as an image classifier.

A self-configuring scheme, formulated as an optimization problem, was applied to define the best configuration for all ANNs employed. The optimization problem was solved by the MPCA meta-heuristic. Such procedure does not need an expert to define a workable neural network.

Finally, the ANN implemented on FPGA produced good results in comparison with software implementation. This is an important result, because the system can be embedded in aircrafts or satellites, allowing a HPC (High Performance Computing) environment working in parallel (data acquisition, pre-processing (data reduction), and image processing (image classification)) with low energy consumption.

References

[AnEtAl14] Anochi, J. A., Velho, H. F. C., Furtado, H. C. M., Luz E. F. P.: Self-configuring two types neural networks by MPCA. 2nd International Symposium on Uncertainty Quantification and Stochastic Modeling (2014)

[An09] Andrecut, M.: Parallel GPU implementation of iterative PCA algorithms. Journal of Computational Biology, 1593–1599, 16 (2009)

[CaRaCh11] Carvalho, A., Ramos, F.M., and Chaves, A.C.: Metaheuristics for the feedforward artificial neural network (ANN) architecture optimization problem. Neural Computing and Applications, 1273–1284, 20 (2011)

[Ca03] Castro, A.P.A.: Edge Detection and Autonomous Navigation using Artificial Neural Networks. M.Sc. Thesis on Applied Computing, Instituto Nacional de Pesquisas Espaciais (INPE), São José dos Campos, SP, Brazil, 2003 (Portuguese).

[Ch13] Chein-I, C.: Hyperspectral Data Processing: algorithm and analysis. Wiley, Hoboken, NJ (2013)

[DeLiDu09] Del Frate, F., Licciardi, G., and Duca, R.: Autoassociative Neural Networks For Features Reduction of Hyperspectral Data. IEEE Geoscience and Remote Sensing Letters, 447–451, 9 (2009)

[GoWo00] Gonzalez, R.C. and Woods, R.E.: Digital Image Processing. Blucher, Saõ Paulo, SP (2000).

[Ha01] Haykin, S.: Artificial Neural Networks. Bookman, Porto Alegre, RS (2001)

[LiEtAl12] Licciardi, G., Reddy, P.M., Chanussot, J., and Benediktsson, J.A.: Linear Versus Nonlinear PCA for the Classification of Hyperspectral Data Based on the Extended Morphological Profiles. IEEE Geoscience and Remote Sensing Letters, 447–451, 9 (2012)

[LuBeVe08] Luz, E.F.P., Becceneri, J.C., Velho, H.F.C.: A new multi-particle collision algorithm for otimization in a high-performance environment. Journal of Computational Interdisciplinary Sciences, 1–7, 1 (2008)

[RuHiWi86] Rumelhart, D.E., Hinton, G.E., and Willians, R.J.: Learning representations by back-propagating errors. Nature, 533–536, 323 (1986)

[SaOl86] Sacco, W. F. and Oliveira, C.R.E.A.: A new stochastic optimization algorithm based on a particle collision metaheuristic. 6th World Congress of Structural and Multidisciplinary Optimization, Rio de Janeiro (1986)

[SiEtAl13] Silva, W. and Habermann, M. and Shiguemori, E. H. and Andrade, L. L. and Castro, R. M.: Multispectral Image Classification using Multilayer Perceptron and Principal Components Analysis. 1st BRICS Countries Congress on Computational Intelligence, Porto de Galinhas (2013)