ISSN: 0975-833X

Available online at http://www.journalcra.com

International Journal of Current Research
Vol. 13, Issue, 01, pp.15987-15992, January, 2021

DOI: https://doi.org/10.24941/ijcr.40671.01.2021

INTERNATIONAL JOURNAL OF CURRENT RESEARCH

RESEARCH ARTICLE

ARTIFICIAL NEURAL NETWORK APPROACH FOR NONLINEAR PRINCIPAL COMPONENTS ANALYSIS

Canan Demir[1*] and Sıddık Keskin[2]

[1*]Yuzuncu Yıl University, Vocational School of Health Care, Van, Turkey
[2]Yuzuncu Yıl University, Faculty of Medicine, Department of Biostatistics, Van, Turkey

ARTICLE INFO

Article History:
Received 23rd October, 2020
Received in revised form
24th November, 2020
Accepted 18th December, 2020
Published online 30th January, 2021

Key Words:

Artificial Neural Network,
Dimensional reduction,
Nonlinear Principal Components Analysis.

ABSTRACT

Background: Nonlinear Principal Component Analysis (NLPCA) is one of the explanatory dimension reduction techniques and presents numerical and graphical results for variable sets including linear or nonlinear relationships. In Nonlinear Principal Component Analysis, categorical and ordinal variables, as well as numerical variables, can be included in the analysis. Linearity assumption for observed variables is not required for Nonlinear Principal Component Analysis. In this study, an artificial neural network approach for NLPCA is explained and applied. **Methods:** The hypothyroid data with 19 variables from 422 patients were used in the application. In order to identify the difference of NLPCA from PCA, the results obtained using Principal Component Analysis (PCA) together with NLPCA were interpreted by presenting in tables and graphics. **Results:** The first two principal components explained 95.65% of the total variance in the NLPCA, while they explained 90.08% of the total variance in the PCA. **Conclusion:** In the analysis conducted via reducing the number of observations, it was observed that NLPCA has a high explanation rate compared to PCA, regardless of the number of observations. This can be attributed to NLPCA's ability to identify nonlinear relationships. Accordingly, it can be said that NLPCA gives effective results in the analysis made with both numerical and categorical variables.

Citation: ***Canan Demir and Sıddık Keskin.*** **2021.** "Artificial Neural Network Approach for Nonlinear Principal Components Analysis", *International Journal of Current Research*, 13, (01), 15987-15992.

INTRODUCTION

Univariate and multivariate methods for determining linear relationships between variables require assumptions related to the data set, such as continuity of variables and linearity of relationships between variables. On the other hand, relationships between variables may not always be linear. One of the multivariate analysis methods based on the linear relationship between variables is Principal Component Analysis (11). In case the variables are continuous and the relationships between them are linear, the effectiveness of the Principal Component Analysis is high. However, in scientific studies, besides continuous variables; categorical or discrete variables may also be involved. In such cases, principal component analysis is not used. One of the alternative solution methods developed for datasets containing variables that are not continuous and have no linear relationship between them is the Nonlinear Principal Components Analysis (NLPCA) (12). This method of analysis, with a linear or nonlinear relationship between them, is a descriptive dimension reduction method that provides numerical and visual results for datasets containing continuous, categorical or discrete variables (2). In practice, the data are often multidimensional. For example, a 16 × 16 matrix is 256 in size. However, the original dataset can be represented with fewer dimensions. In this case, the dataset is reduced to a smaller size without much loss of information. These dimensions are called components and are defined by a curve in the original data field (17). In many scientific studies, PCA is often used. However, given that PCA is a linear method and the relationships between variables in many fields, especially in engineering sciences, are not linear, it can be stated that it will be more appropriate to use NLPCA instead of PCA (3). NLPCA can be considered as a nonlinear generalization of PCA. However, NLPCA is superior to PCA since it has high explanatory power with few variables (14).

Artificial neural networks are computational systems developed in order to realize the skills such as the ability to derive, create and discover new information through learning, which is one of the features of the human brain (9). Artificial Neural Networks (ANN) is a flexible method that determines the relationship between them without requiring any assumption about the variable structure for the solution of complex structure problems that cannot be solved manually. ANN can be used especially for classification and dimension

Corresponding author:** ***Canan Demir,
Yuzuncu Yıl University, Vocational School of Health Care, Van, Turkey.

reduction purposes. The ANN calculates the weights of the network step by step to assign the units to their classes with the least error using the back propagation algorithm and minimizing the network error (10). ANN performs machine learning using examples, its programs and operation method are not similar to known programming methods.

In order to the ANN to run reliably, it must first be trained and its performance tested. ANN are functional tools used in areas such as prediction, classification, pattern recognition, data-based process modeling and nonlinear process control (16). In this study, the NLPCA method for nonlinear dimension reduction was investigated on the basis of ANN. In addition, the results obtained by practicing in the field of health sciences to show the effectiveness of NLPCA and to expand its usability have been interpreted.

MATERIALS AND METHODS

Material: In the study, from the free-access data site as application material (http://mlr.cs.umass.edu/ml/machine-learningdatabases/thyroiddisease/hypothyroid.data Access date: 10.09.2018) 19 variables hypothyroid data of 422 patients provided were used and the variables and their features are given in Table 1.

Table 1.Variables in the study and their properties

Variable name	Category	Type
Thyroxine treatment	(1) Yes, (0) No	Nominal
Anti-thyroid treatment	(1) Yes, (0) No	Nominal
A history of Hypothyroidism	(1) Yes, (0) No	Nominal
Pregnancy	(1) Yes, (0) No	Nominal
Tumor	(1) Yes, (0) No	Nominal
Goiter	(1) Yes, (0) No	Nominal
A history of thyroxine	(1) Yes, (0) No	Nominal
Having thyroid surgery	(1) Yes, (0) No	Nominal
A history of Hyperthyroidism	(1) Yes, (0) No	Nominal
Comorbidity	(1) Yes, (0) No	Nominal
Lithium	(1) Yes, (0) No	Nominal
Hypothyroidism	(1) Yes, (0) No	Nominal
Gender	(1) Male,.. (2) Female	Nominal
Age	Continuous	Continuous
TSH	Continuous	Continuous
TT4	Continuous	Continuous
FTI	Continuous	Continuous
T3	Continuous	Continuous
T4U	Continuous	Continuous

Method: Linear Principal Component Analysis is a widely used method, however, this method requires the variables to be continuously variable and the relationships between them to be linear. If the relationship between variables is not linear, linear PCA is generalized and NLPCA has been developed to detect nonlinear relationships. NLPCA is a nonlinear extension of PCA.

In applications, the data can be expressed as $x(t) = (x_1,..., xi)$. Here each variable x_i contains ($i = 1, ..., l$) n observations. PCA tries to find u as a linear combination of x_i and a vector associated with a. This relationship can be written as follows.

$$u(t)= a.x(t) \quad (1)$$

Thus;

$$<\|x(t)-a.u(t)\|^2> \text{ is minimized} \quad (2)$$

In Equation (2), u is called the first fundamental component. The first eigenvector of the covariance matrix (also called the Empirical Orthogonal function (EOF)) a usually defines a spatial model. The error is also received with the second basic component from x-au and continues for the number of these components. In practice, principal components are obtained simultaneously (7).

The main difference between NLPCA and PCA; NLPCA allows a nonlinear mapping from x to u, while PCA only allows a linear mapping. It contains 3 hidden layers of variables between the input and output layers of the neural network variables in Figure 1 (or "neuron") to realize NLPCA. Defined as the transfer function, f_1 maps from the input column vector x of length l, to the column vector of length m, which is denoted by $h^{(x)}$, the first hidden layer. Accordingly, $h_k^{(x)}$ is written as follow;

$$h_k^{(x)}=f_1((W^{(x)}x+b^{(x)})_k) \quad (3)$$

$w^{(x)}$ in the equation (3) is the weight matrix with the dimension "m x 1" and $b^{(x)}$ is the m-dimension column vector with the bias parameters ($k = 1, ..., m$). Similarly, a second transfer function, f_2, maps from the encoder layer to the hidden layer showing the nonlinear basic component, and for this nonlinear basic component u is written as follow;

$$u= f_2(w^{(x)} .h^{(x)}+ ^{(x)}) \quad (4)$$

While f_2 is usually taken as a linear function, the transformation function f_1 is nonlinear (although its function is not certain, it can usually be a hyperbolic tangent or sigmoid function). Next, a transformation function f_3 maps from u to the last hidden layer, $h^{(u)}$. Thus $h_k^{(u)}$ is written as follow ($k = 1,\ldots, m$);

$$h_k^{(u)}=f_3((w^{(u)}u+b^{(u)})_k) \quad (5)$$

This function is followed by equation (6).

$$= f_4((W^{(u)}h^{(u)}+ ^{(u)})_i) \quad (6)$$

Here the error function $j = <\|x-\|^2> W^{(x)}, b^{(x)}, w^{(x)}, ^{(x)}, w^{(u)}, b^{(u)}, W^{(u)}$ and $^{(u)}$ is minimized by finding optimal values (6). The mean squared error between the original data x and the neural network output is thus minimized.

The total number of free (weight and bias) parameters used by NLPCA is $(m+f+1)(M_1+M_2)+m+f$. The number of hidden neurons in both the coding and decoding layers and the choice of M follows a general rule. A larger M enhances the network's nonlinear modeling ability, however it can also lead to overfitting. For a lower M, the accuracy may be low as the network has limited representation capacity. The value of M relates to the complexity of nonlinear functions that can be produced by the network. If f_4 is a linear function and M = 1, equation (6) shows that all is linearly related to a single hidden neuron, thus there can only be a linear relationship between variables. For nonlinear solutions, M must be greater equal 2 (12).

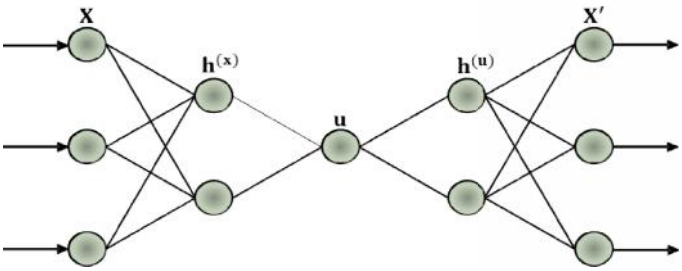

Figure 1. Neural network model for Nonlinear Principal Components Analysis

In Figure 1, there are 3 hidden layers (indicated by a circle) between the input layer x on the left side and the output layer on the right side. Next to the input layer, there is the coding layer, followed by "bottleneck" (with a single neuron u) and analysis layers, respectively. The nonlinear function maps from the high-dimensional input space to the low-dimensional bottleneck space, followed by mapping back to the original space shown as output from the bottleneck space. The outputs here ensure that the error function with $J = \langle \|x-\|^2 \rangle$ is minimized as close to the input variables as possible. Compression or dimension reduction of data is provided by u, the neuron in the bottleneck, and this is called the nonlinear principal component (6).

Multi-layer Perceptron: Single-layer network (Perceptron), one of the first models of artificial neural networks, was first developed by Frank Rosenblatt. The main feature of single-layer network models is the ability of these models to solve linear problems. Nonlinear problems cannot be learned with such networks. Multi-layer networks have been developed to solve this problem. This model developed by Rumelhart et al. is called error propagation model or back propagation model. This model uses a learning method called Delta learning rule. The structure of multi-layer networks is given in Figure 2. Multi-layer networks are feed forward; it has three layers, the input layer, the hidden layer and the output layer. The input and output layers have the same number of neurons for the signals.

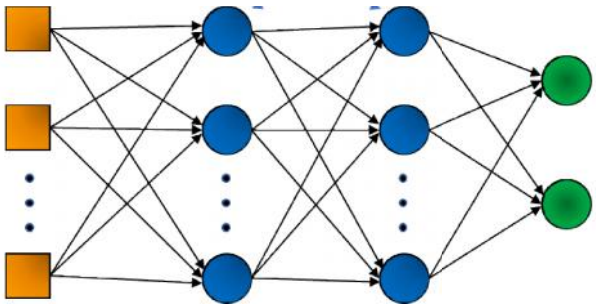

Figure 2. Multi-layer Perceptron

No information is processed in the input layer. This layer receives information from the external environment and sends it to the intermediate layer. Each processing element in the input layer is linked to the processing element in the next hidden layer. The hidden layer processes the information from the input layer and sends it to the output layer. Again, each processing element in the hidden layer is linked to all processing elements in the output layer. There can be more than one hidden layer. The number of hidden layers and the number of processor elements in hidden layers is found by trial and error. The output layer processes the information from the hidden layer and transmits the outputs to the outside world (4). Single-layer networks have significant limitations. Minsky and Papert (1969) have demonstrated that the multi-layer perceptron network can overcome many restrictions. The error of hidden (intermediate) layer units is determined by the back propagation of the errors of the output layer units. For this reason, this method is often referred to as the back propagation rule. Back propagation can also be considered as the generalization of the delta rule for multilayered networks of nonlinear activation functions (15).

Backpropagation: Standard back propagation network; it has an input layer, an output layer, and at least one hidden layer. There is no theoretical limit for the number of hidden layers, however usually one or two hidden layers are used. Each layer is fully connected to subsequent layers (1). Multi-layer networks operate on the principle of supervised learning. Both inputs and outputs values must be presented to the network during training. The task of the network is to produce the output corresponding to that input for each input. The learning rule of the multi-layer network is the generalized version of the Delta learning rule based on the least squares method. For this reason, the learning rule is also called “Generalized Delta Rule”. This rule consists of two stages. The first is the forward calculation phase, where the network output is calculated, and the second is the reverse calculation phase, where the weights are changed (4,5). Forward calculation: This phase is the calculation phase of the network output. At this stage, information processing begins with the display of data in the training set from the input layer to the network. As mentioned before, no information processing takes place in the input layer. The incoming entries are sent to the middle layer without any changes. This is indicated by the equation $y_i = x_i$. Each unit in the middle layer receives the information from all units in the input layer by weighting them with connecting weights. Net input to the units in the middle layer is calculated as follow;

The first is the forward calculation phase, where the network output is calculated, and the second is the reverse calculation phase, where the weights are changed (4,5). Forward calculation: This phase is the calculation phase of the network output. At this stage, information processing begins with the display of data in the training set from the input layer to the network. As mentioned before, no information processing takes place in the input layer. The incoming entries are sent to the middle layer without any changes. This is indicated by the equation $y_i = x_i$. Each unit in the middle layer receives the information from all units in the input layer by weighting them with connecting weights. Net input to the units in the middle layer is calculated as follow;

$$n^1 = \left(\sum_{i=1}^{J} w_{i;} x_i + b_i\right) \quad (7)$$

w_{ir}: i. input layer element,
r: weight value of the connection that connects to the middle layer unit,
x_i: Output of the i. processor unit in the input layer
n_1: It shows the output value of the middle layer unit.
Net input to the unit in the output layer is expressed as follow;

$$n^2 = \sum_{k-1}^{s} w_k^2 a_k^1 + b^2 \quad (8)$$

Network output;

$$\hat{y} = g\{\sum_{k-1}^{s} w_k f\left(\sum_{i-1}^{j} w_i x_i + b^1\right) + b^2\} \quad (9)$$

and total error is computed by the following equation;

$$E_D = \frac{1}{2}\sum_{i-1}^{j}(y - \hat{y})^2 \quad (10)$$

Usually the sigmoid or hyperbolic tangent function is used as the transfer function, but this is not a requirement. Since the derivative of the function must be taken in the back propagation, it should be stated that the selected function is a derivable function (8). Backward calculation: The back propagation step involves running the entire network backwards. The difference between the output of the last layer and the desired output is usually modified by the derivative of the transfer function and back propagate to the previous layers. Thus, the error is reduced in the next iteration. This process proceeds for the previous layer (s) until the input layer is reached. Inputs to be used as training sets and corresponding outputs are normalized (Equation 11)

The steps can be listed as follows:

- Inputs to be used as training sets and corresponding outputs are normalized (Equation 11)

$$x_y = \frac{x_i - x_m}{x_m - x_m} \quad (11)$$

- The initial parameters, weights and threshold of the network are determined

- Each unit in the middle layer receives the information from all the unit in the input layer by weighting them with their connection weights. Net input to units in the middle layer is calculated by the following equation;

$$n_k^1 = \left(\sum_{i=1}^{j} w_{i,} x_i + b_1\right) \quad (12)$$

- The output of the intermediate layer is determined using an activation function for the net input to the units in the intermediate layer;

$$a_k^1 = f\left(\sum_{i-1}^{j} w_i x_i + b_1\right) \quad (13)$$

- Each unit in the output layer receives information from all units in the intermediate layer by weighting them with connection weights. Net input to units in the output layer is calculated by the following equation;

$$n^2 = \sum_k^s w_k^2 a_k^1 + b_2 \quad (14)$$

- Output of the output layer is determined using a linear activation function for the output layer;

$$g(\sum_k^s w_k^2 a_k^1 + b^2) = \sum_k^s w_k^2 a_k^1 + b^2 \quad (15)$$

- The error term, which is the difference between the actual value and the outputs produced by the network, is the sum of the squares error is calculated by the following equation;

$$E_D = S. \quad = \frac{1}{2}\sum_{i=1}^{j}(y - \hat{y})^2 \quad (16)$$

- If the Sigmoid function is used in the output unit, the error (δ) is calculated as follows;

$$\delta_i(t) = \frac{\partial E_D(t)}{\partial n_k^1(t)} = \sum_k \frac{\partial E_D(t)}{\partial n^2(t)} \cdot \frac{\partial n^2(t)}{\partial n_k^1(t)}$$

$$\delta_i(t) = f'(n_k^1(t)) \sum_k w_k^{(2,1)} \Delta_i$$

$$\Delta_i(t) = -\frac{\partial E_D(t)}{\partial n^2(t)} = g'(n^2(t)) e_i$$

$$\delta = (y - \hat{y})\hat{y}(1 - \hat{y}) \quad (17)$$

- After calculating the change amount, new weights in *t*. iteration; updated weights between the input layer and the hidden layer are calculated as follows:

$$\frac{\partial E_D(t)}{\partial w_k^{(2,1)}} = \frac{\partial E_D(t)}{\partial n^2(t)} \cdot \frac{\partial n^2(t)}{\partial w_k^{(2,1)}}$$

$$\Delta w_i = e_i x_i$$

$$\Delta w_i(t) = \alpha(y - \hat{y}) x_i$$

$$w_i(t+1) = w_i(t) + \Delta w_i(t)$$

$$w_k^{1(t+1)} = w_k^{1(t)} + \eta \Delta w^1 \quad (18)$$

And the updated weights between the hidden layer and the output layer are calculated as follows.

$$\frac{\partial n^2(t)}{\partial w_k^{(2,1)}(t)} = a_k^1(t)$$

$$\frac{\partial n_k^1(t)}{\partial w_k^{(1,1)}(t)} = x_i$$

$$w_k^{2(t+1)} = w_k^{2(t)} + \eta \Delta w^2 \quad (19)$$

The updated value in *t* iteration of bias value is calculated as follow;

$$\frac{\partial E_D(t)}{\partial b^2(t)} = \frac{\partial E_D(t)}{\partial n^2(t)} \cdot \frac{\partial n^2(t)}{\partial b^2(t)}$$

$$b_i(t+1) = b_i(t) + \Delta b_i(t)$$

$$\Delta b_i(t) = \alpha(y - \hat{y}) \quad (20)$$

At the end of these processes, all the weights of the network will be have been changed. An iteration is completed by making both forward and backward calculations. Then, a new sample is given and the second iteration is started. Here, the network error is expected to decrease in each iteration and this is shown in Figure 3. Network error tends to decrease with increasing iteration number. After a certain number of iterations, the error does not decrease further. This means that the network stops learning and a better result cannot be achieved. These computations are continued until the difference between the desired output and the calculated output is minimized. The end of learning is done with a stop criterion. This criterion generally takes place when the method decreases to an acceptable level (8).

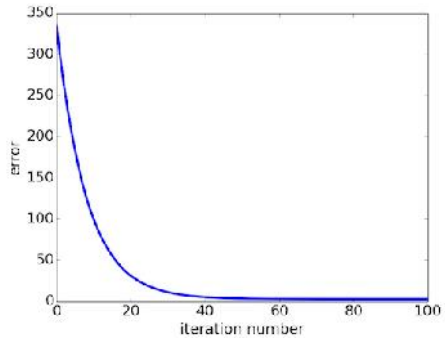

Figure 3. The relationship between the number of iterations and the error

RESULTS

NLPCA works by training a feed forward neural network to perform mapping, where network inputs are reproduced in the output layer. The architecture of the neural network used for the implementation of NLPCA is shown in Figure 4.

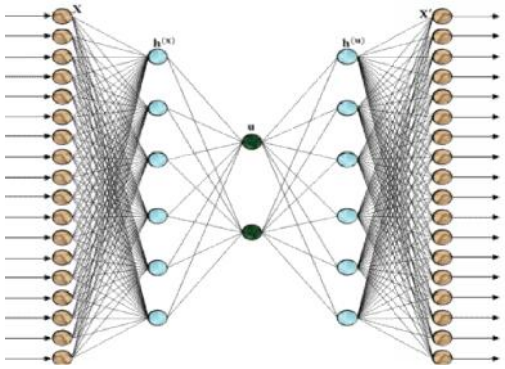

Figure 4. 18-6-2-6-18 NLPCA model

This architecture is a five-layer neural network, 3 of which are hidden. This five-layer neural network creates an auto-associative neural network. In the auto-associative neural network model, essential components are obtained in the bottleneck layer. If the number of bottleneck neurons is one (as shown in Figure 1), one-dimensional nonlinear principal component is obtained. That is, it is a one-dimensional nonlinear curve that best fits these data. If the number of bottleneck neurons is two, a two-dimensional nonlinear curve is obtained that best fits the data. In the network architecture used; a total of 5 layers were used: input layer, coding layer, bottleneck layer, decoding layer and output layer. Nonlinear hyperbolic tangent function is used in the coding and decoding layer, and linear function is used in the other layers. The Conjugate Gradient Descent (CGD) algorithm was preferred as the training algorithm since it has a high learning speed compared to other algorithms and the learning rate constants are calculated automatically and adaptively thus they do not need to be specified before training. The original data set consists of an 18x422 matrix with 12 categories and 6 continuous variables. The number of hidden neurons in both the coding and decoding layers is determined as $M_1 = M_2 = 6$ in the network used to realize NLPCA. The total number of free (weight and threshold) parameters used by NLPCA was calculated with $(m + f + 1) (M_1 + M_2) + m + f$ (M_1 and M_2, respectively, the number of neurons in the coding and decoding layers, m variable number, f is the number of neurons in the bottleneck layer). In this case, the total number of calculated parameters for the 2 principal components is 271.

Table 2. Accounted variance for PCA and NLPCA

	Accounted variance (%) for PCA			Accounted variance (%) for NLPCA		
	Dimension 1	Dimension 2	Total	Dimension 1	Dimension 2	Total
n=422	70.16%	19.92%	90.08%	92.20%	3.45%	95.65%
n=300	67.10%	0.02%	67.12%	91.66%	3.49%	95.15%
n=200	66.48%	0.02%	66.50%	93.22%	2.57%	95.79%
n=100	65.49%	0.02%	65.51%	94.68%	0.01%	94.69%

In order to see the difference of NLPCA from PCA, both PCA and NLPCA were applied to the data set containing categorical and continuous variables. For n = 422, the two-dimensional (2D) PCA approach accounts for 90.08% of the total variance, while the 2D NLPCA approach accounts for 95.65% of the total variance. For n = 300, the 2D PCA approach accounts for 67.12% of the total variance, while the 2D NLPCA approach accounts for 95.15% of the total variance. For n = 200, the 2D PCA approach accounts for 66.50% of the total variance, while the 2D NLPCA approach accounts for 95.79% of the total variance. For n = 100, the 2D PCA approach accounts for 65.51% of the total variance, while the 2D NLPCA approach accounts for 94.69% of the total variance (Table 2). The results show that NLPCA successfully reduced dimensionality regardless of the number of observations (Figure 5).

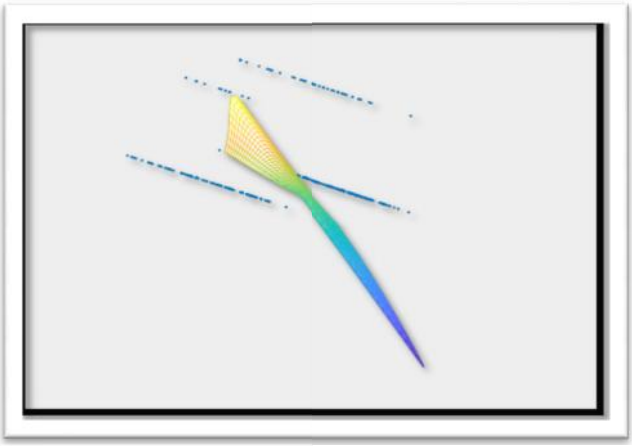

Figure 5. 18-6-2-6-18 NLPCA

DISCUSSION

In the study, the applications of both PCA and NLPCA were made by examining the ANN approach for PCA as well as NLPCA in order to explain NLPCA better and to examine its effectiveness. In the analysis conducted via reducing the number of observations, it was observed that NLPCA has a high explanation rate compared to PCA, regardless of the number of observations. This can be attributed to NLPCA's ability to identify nonlinear relationships. Accordingly, it can be stated that NLPCA gives effective results in the analysis made with both numerical and categorical variables. In a study, a NLPCA method integrating the basic curve algorithm and neural networks was used. Both simulation and real problems have been shown that NLPCA is a good approach for both applications and can be used to solve important process problems (3). In another study, an algorithm based on NLPCA was developed to detect ischemic heartbeats from the ECG signal of the patients and to accurately classify the pulses. In the classification made with NLPCA, it was observed that a training set containing only two non-linear components and 1000 normal samples from each file was correctly classified with a high rate of 80% for normal beats and 90% for ischemic beats (18).

In another study, the NLPCA method based on an auto-associative neural network was used and it was stated that NLPCA performed better than PCA in reconstructing the water quality data of the Piabanha watersheds and also explained most of the data variance (19).
In the study, NLPCA method that finds and models Nonlinear Principal Components in artificial neural networks is presented and the NLPCA method has been found to be a better approach to solving nonlinear problems. It has also been shown that the NLPCA method applied to health data explains the variance better regardless of the number of observations.

Acknowledgments

This work is derived from Canan Demir's PhD thesis.
For analysis of the data, Matlab R2013 package program was used for Matthias Scholz's Nonlinear PCA toolbox at http://www.nlpca.org/ open access internet address.

Conflict of Interest: The authors declare no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding: This study did not take any specifc donation from funding organizations in the public, commercial, or not-for-proft sections.

Key points:

- NLPCA has been applied to real data.
- NLPCA has also given successful results in data in the field of health.
- Explained variance of NLPCA was found greater than that of PCA regardless the number of observations.

REFERENCES

(1) D. Anderson, G. McNeill, Artificial Neural Networks Technology, Rome Laboratory RL/C3C Griffiss AFB, New York, 1992.
(2) C. Demir Nonlinear Principal Components Analysis and Application in Health Master Thesis, Van Yuzuncu Yil University, 2010.
(3) D. Dong, TJ. McAvoy, 1996. Nonlinear Prıncıpal Component Analysis-Based On Principal Curves and Neural Networks. Computers chem. Engng. 20, 65-78.
(4) A. Gulbag, Quantification of Volatile Organic Compounds with Artificial Neural Network and Fuzzy Logic Based Algorithms, Ph.D. Thesis, Sakarya University, 2006.
(5) H. Guler, Estimation of the Effect of Alloy Elements on the Corrosion Behavior of Zinc-Aluminum Alloys by Artificial Neural Network, Master Thesis, Sakarya University, 2007.
(6) WW. Hsieh, Nonlinear principal component analysis by neural networks, 2001. Tellus. 53, 599-615.
(7) IT. Jolliffe, Principal Component Analysis, Springer-Verlag, New York, 1986.
(8) VSA. Kargı, Artificial Neural Network Models and Application in a Textile Company, Ph.D. Thesis, Uludağ University, 2013.
(9) MO. Kaya, Estimation of Prostate Specific Antigen (Psa) with Different Artificial Neural Network Models, Master Thesis, Fırat University, 2010.
(10) YE. Kuyucu, Comparison of Logistic Regression Analysis (LRA), Artificial Neural Networks (ANN) and Classification and Regression Trees (C & Rt) Methods and An Application in Medicine, Master Thesis, Gaziosmanpasa University, 2012.
(11) SY. Kung, KI. Diamantaras, A Neural Network Learning Algorithm for Adaptive Principal Component Extraction (APEX). I Hyperlink "https://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=132"nt Hyperlink "https://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=132". Hyperlink "https://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=132"C hyperlink "https://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=132"onf hyperlink "https://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=132". hyperlink "https://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=132"A hyperlink "https://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=132"coustigs hyperlink "https://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=132", hyperlink "https://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=132"S hyperlink "https://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=132"peech, hyperlink "https://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=132"and hyperlink "https://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=132" hyperlink "https://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=132"si hyperlink "https://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=132"gn hyperlink "https://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=132". hyperlink "https://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=132" hyperlink "https://ieeexplore.ieee.org/xpl/ mostRecentIssue.jsp?punumber =132"Procs-Albq. 90; 3-6 April 1990; NM, USA.
(12) MA. Kramer, 1991. Nonlinear principal component analysis using auto-associative neural networks, AIChE Journal. 37, 233-243.
(13) MathWorks. Matlab version R2013b. Natick: 2013.
(14) AH. Monahan, 2000. Nonlinear Principal Component Analysis by Neural Networks: Theory and Application to the Lorenz System, J. Climate. 13, 821-835.
(15) H. Okut, Bayesian Regularized Neural Networks for Small n Big p Data. Rosa JLG. Artificial Neural Networks-Models and Applications, London, InTechOpen, 27-48, 2016.
(16) U. Ozer, Constrained Neural Networks, Master Thesis, Boğaziçi University, 2012.
(17) Scholz M. Nichtlineare Haupt component enanalyse auf Basis neuronaler Netze Diplomarbeit, Humboldt-Universität zu Berlin, 2002.
(18) T. Stamkopoulos, K. Diamantaras, N. Maglaveras, M. Strintzis, 1998. ECG Analysis Using Nonlinear PCA Neural Networks for Ischemia Detection, IEEE-Trans. Sign. Procs. 46, 3058-3069.
(19) MD. Villas-Boas, F. Olivera, JPS. de-Azevedo, 2017. Assessment of the water quality monitoring network of the Piabanha River experimental watersheds in Rio de Janeiro, Brazil, using auto-associative neural networks. Environ Monit Assess. 189, 439-454.

****