Autoencoders as an alternative approach to Principal Component Analysis for dimensionality reduction. An application on simulated data from psychometric models

Monica Casella[a], Pasquale Dolce[b], Michela Ponticorvo[a] and Davide Marocco[a]

[a] *University of Naples Federico II, Department of Humanistic Studies, Naples, Italy*
[b] *University of Naples Federico II, Department of Public Health, Naples, Italy*

Abstract

Dimensionality reduction is defined as the search for a low-dimensional space that captures the "essence" of the original high-dimensional data. Principal Component Analysis (PCA) is one of the most used dimensionality reduction technique in psychology and behavioral sciences for data analysis and measure development. However, PCA can capture linear correlations between variables, but fails when this assumption is violated. In recent years, a variety of nonlinear dimensionality reduction techniques have been proposed in other research fields to overcome this limitation. In this paper, we focus on non-linear autoencoder, a multi-layer perceptron, with as many inputs as outputs and a smaller number of hidden nodes. We investigate the relation between the intrinsic dimensionality of data and the autoencoder's internal nodes in a simulation study, comparing autoencoders and PCA performances in term of reconstruction error. The evidence from this study suggests that autoencoder's ability in dimensionality reduction is very similar to PCA, and that there is a relation between internal nodes and data dimensionality.

Keywords 1

Dimensionality reduction, Principal Component Analysis (PCA), Autoencoder, Artificial Neural Networks

1. Introduction

The transformation of data from high-dimensional space into a meaningful low-dimensional space, which ideally corresponds to the intrinsic dimensionality of the original data, is referred as "dimensionality reduction". This transformation is important in several domains because it mitigates undesired properties of high-dimensional spaces such as the curse of dimensionality [1].

However, determining the number of dimensions of a data set requires researchers to take several important decisions: in particular, the choice of extraction method and the decision about how many components to retain are considered among the most critical in psychological scale development [2].

Intrinsic dimensionality, as an important intrinsic characteristic of high-dimensional data, can be defined as the minimum number of coordinates which are necessary to describe data points without significant information loss: because the process of dimensionality reduction inevitably leads to information loss, it is very important to preserve the main and important characteristics of the original data as much as possible. So, dimensionality reduction is not only related to data compression, but also to feature extraction [3].

Traditionally, dimensionality reduction is performed using linear techniques such as Principal Components Analysis (PCA), which is one of the most used statistical techniques in behavioral sciences and is a standard part of measure development [4].

Proccedings of the Third Symposium on Psychology-Based Technologies (PSYCHOBIT2021), October 4–5, 2021, Naples, Italy
EMAIL: mo.casella@studenti.unina.it (A. 1); pasquale.dolce@unina.it (A. 2); michela.ponticorvo@unina.it (A. 3); davide.marocco@unina.it (A. 4)
ORCID: 0000-0002-6017-602X (A. 1); 0000-0002-7588-6067 (A. 2); 0000-0003-2451-9539 (A. 3); 0000-0001-5185-1313 (A. 4)


PCA was first introduced by Pearson in 1901 and developed by Hotelling in 1933 [5,6]. The central idea of PCA is to reduce the dimensionality of a dataset with a large number of interrelated variables preserving as much variability as possible. This reduction is achieved by finding new variables, the principal components, which are linear functions of the original variables, which are uncorrelated and sorted so that the first few retain most of the variation present in all the original variables [7]. However, assumptions required for PCA are not always satisfied in psychology and behavioral sciences. In fact, PCA assumes that the relationships between variables are linear, and all variables should be assessed on an interval or ratio level of measurement. Therefore, PCA may not always be the most appropriate method of analysis [8].

In contrast to the traditional linear techniques for dimensionality reduction, machine learning techniques can deal with complex nonlinear data, and they represent a valuable alternative to classical methods. In this context, a considerable amount of work has been done on non-linear extensions of PCA and a variety of approaches has been proposed. Among others, Autoencoders seem a valuable alternative to PCA for dimensionality reduction.

Autoencoder, also called auto-associative neural network or bottleneck network, is a multi-layer perceptron with as many inputs as outputs and a smaller number of hidden feature units. During training, the targets for the output units are set to be equal to the inputs. The weights in the network are then trained to minimize the square error of the reconstruction [9]. Because of this learning strategy, it can be shown that the linear autoencoder, with *n* features, converges to the *n-th* dimensional PCA subspace [10,11]. An extension of the linear autoencoder consists in the introduction of a nonlinear mapping by adding nonlinear hidden layers. Such a neural network effectively performs a nonlinear principal component analysis, overcoming limits of linear dimensionality reduction [12]. Autoencoders are applied to many problems, from facial recognition to customer segmentation [13,14], but they're absent in psychometric research. Furthermore, although it is known that autoencoders have good performance in data compression, little research has been conducted on the relationship between the intrinsic dimensionality of the data and the number of internal nodes.

In line with these considerations, the aim of this paper is to investigate in a systematic way a possible relation between the number of hidden layer nodes and the intrinsic dimensionality of data, comparing PCA and autoencoders reconstruction error on artificial datasets.

Datasets are generated from factor-based population, a choice due to their diffusion in psychometric research [15].

The rest of article is organized as follows: first, methods are briefly described, and the study is presented in a more detailed way; then, experimental procedures, data analysis and results are showed; finally, section 4 concludes the paper and discusses several future research directions.

2. Methods

In many areas among the social and life sciences the amount of high-dimensional data has rapidly increased within the past year: to handle such real-world data adequately, dimensionality needs to be reduced into meaningfully expression in low-dimensional space.

In this section two approaches for dimensionality reduction will be described: subsection 2.1 discuss Principal Component Analysis, the most famous linear dimensionality reduction technique; then, we describe Autoencoders, an alternative non-linear approach for dimensionality reduction more recently proposed.

2.1. Principal Component Analysis (PCA)

PCA can be defined as the orthogonal projection of data onto a lower dimensional linear space, such that the variance of the projected data is maximized [16].

To derive the form of PC's, suppose x is a vector of p variables with a covariance matrix S. The first step is to search for a linear function $\alpha_1' x$ of the elements of x having maximum variance:

$$\alpha_1' x = \alpha_{11} x_1 + \alpha_{12} x_2 + \ldots + \alpha_{1p} x_p = \sum_{j=1}^{p} \alpha_{1j} x_j \quad (1)$$

where α_1' is a vector of p constants $\alpha_{11}x_1, \alpha_{12}x_2, \ldots, \alpha_{1p}x_p$ and ' denotes transpose.
The variance of the projected data is given by:

$$var[\alpha_1' x] = \alpha_1' S \alpha_1 \quad (2)$$

and is maximized under the normalization constraints $\alpha_1' a_1 = 1$ using the techinque of Lagrange multipliers. It follows that:

$$\alpha_1' S \alpha_1 = \lambda(\alpha_1' a_1 - 1) \quad (3)$$

where λ is the Lagrange multipliers. By setting differentiation with respect α_1 equal to zero, the solution of this problem can be obtained as a unit eigenvector of the covariance matrix S corresponding to the largest eigenvalue. Thus, α_1 is the eigenvector corresponding to the largest eigenvalue of S, and $var(a_1 x) = \alpha_1' S \alpha_1 = \lambda 1$ the largest eigenvalue.

In general, the *k*th PC of x is $\alpha_k x$ and its variance is λ_k , where λ_k is the largest eigenvalue of S and α_k is the corresponding eigenvector or, also, the vector of loadings for the *k*th component. We can define additional principal components in an incremental fashion by choosing each new direction to be that which maximizes the projected variance amongst all possible directions orthogonal to those already considered. To summarize, principal component analysis involves evaluating the covariance matrix S of the dataset and then finding the *k* eigenvectors of *S* corresponding to the *k* largest eigenvalues.

PCA can be also viewed as a linear projection of data points into a lower dimensional space such that the squared reconstruction loss is minimized. In general, a dimension reduction technique provides an approximation $\hat{x}(t)$ to $x(t)$ which is the composition of two functions *f* and *g*:

$$x(t) = \hat{x}(t) + \epsilon(t) = g\left(f\left(x(t)\right)\right) + \epsilon(t) \quad (4)$$

The projection function $f : R^p \rightarrow R^z$ projects the original *P*-dimensional data $x(t)$ onto a *Z*-dimensional subspace, while the expansion function $g : R^Z \rightarrow R^P$ defines a mapping from the *Z*-dimensional space back into the original *P*-dimensional space with $\epsilon(t)$ as the residue. The feature extraction problem may involve the determination of functions *f* and *g*. The mean square error (MSE) in reconstructing the original data is:

$$MSE = E\left[\left\|x - g\left(f(x)\right)\right\|^2\right] \quad (5)$$

It can be shown that PCA is the algorithm which obtains the smallest MSE among all techniques with linear projection and expansion functions *f* and *g* [17].

2.2. Autoencoders

Autoencoder, also called auto-associative neural network, is a multi-layer perceptron having the same number of outputs as inputs, designed to learn an approximation to the identity function, so as the output is as similar to the input as possible [18]. This is achieved by minimizing an error function which captures the degree of mismatch between the input vectors and their reconstructions, typically a sum-of-squares error of the form:

$$E(w) = \frac{1}{2}\sum_{j=1}^{P}\left\|y(x_j, w) - x_j\right\|^2 \quad (6)$$

When used with a hidden layer smaller than the input/output layers and linear activations only, as represented in Figure 1, the autoencoder performs a compression scheme which was shown to be equivalent to PCA [10, 11]. In fact, both principal component analysis and the neural network are using linear dimensionality reduction and are minimizing the same sum-of-squares error function.

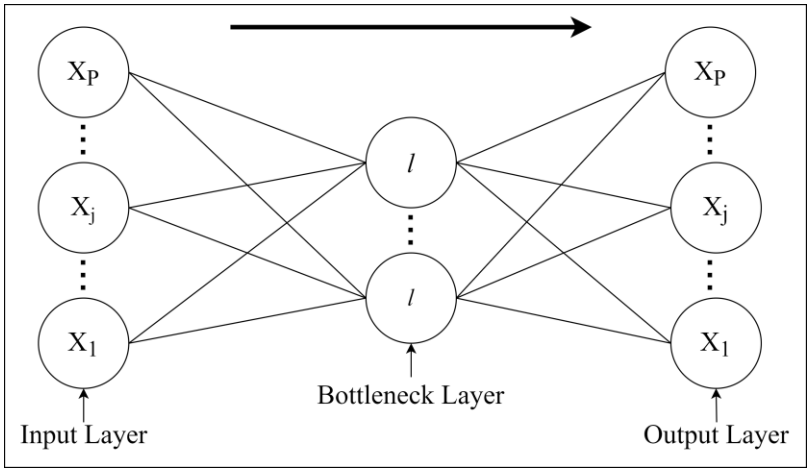


Figure 1: Linear Autoencoder with a single hidden layer (l = linear activation function)

In 1991, an interesting non-linear generalization was introduced by Kramer [12]. The network described by Kramer is again trained by minimization of the error function (6).

We can view this network as two successive functional mappings f and g as indicated in Figure 2. The first mapping f projects the original P-dimensional data into a Z-dimensional subspace S defined by the activations of the units in the second hidden layer. Because of the presence of the first hidden layer of nonlinear units, this mapping is very general, and is not restricted to being linear.

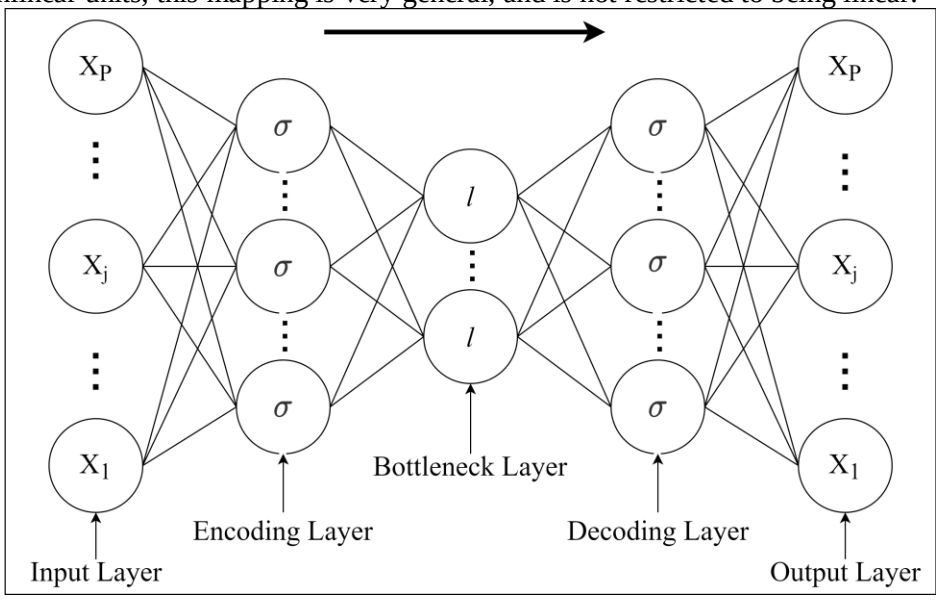


Figure 2: Non-linear autoencoder introduced by Kramer (σ = sigmoidal activation function, l = linear activation function)

Let consider a network with p neurons in the input and output layers, k neurons in the mapping and demapping layers and a single neuron in the bottleneck layer. Without biases, the projection functions f has the form:

$$f(x) = \sum_{i=1}^{k} w_{1i}^{(2)} \sigma \sum_{j=1}^{P} w_{ij}^{(1)} x_j \tag{7}$$

Similarly, the second half of the network defines an arbitrary functional mapping g from the Z-dimensional space back into the original P-dimensional input space and takes the form:

$$g(y) = \left[g_1(y) \dots g_p(y)\right]^T = \left[\sum_{i=1}^{k} w_{1i}^{(4)} \sigma\left(w_{i1}^{(3)} y\right) \sum_{i=1}^{k} w_{ji}^{(4)} \sigma\left(w_{i1}^{(3)} y\right)\right] \tag{8}$$

where $w_{ij}^{(m)}$ is the weight between the *i-th* neuron of layer $m + 1$ and the *j-th* neuron of layer *m*, and σ is a non-linear function, usually a sigmoid or a hyperbolic tangent function.

This process has a simple geometrical interpretation, as indicated for the case P = 3 and Z = 2 in Figure 3. The function *f* defines a projection of points from the original P-dimensional space into the Z-dimensional subspace S; then, the function *g* maps from a Z-dimensional space S back into a P-dimensional space and therefore defines the way in which the space S is embedded within the original *x*-space. Since the mapping *g* can be nonlinear, the embedding of S can be nonplanar, as indicated in the figure.

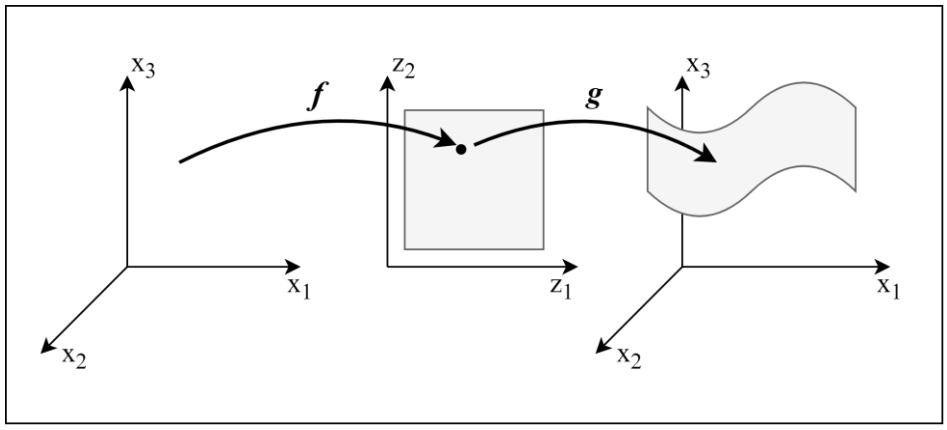


Figure 3. Geometrical interpretation of the mappings performed by the network in Figure 2 for the case of *P* = 3 inputs and *Z* = 2 units in the middle hidden layer.

Autoencoder has the advantage of not being limited to linear transformations and can learn more complicated relations between visible and hidden units, although it contains standard principal component analysis as a special case. However, unlike PCA, the coordinates of the output of the bottleneck are correlated and are not sorted in descending order of variance. Moreover, computationally intensive nonlinear optimization techniques must be used, and there is the risk of finding a suboptimal local minimum of the error function [16]. One solution to mitigate this problem was introduced by Hinton in 2006, who proposed a "layer-wise pretraining" procedure for binary data using restricted Boltzmann machines [19].

2.3. Objectives of the work

Dimensionality reduction implies capturing the "essence" of the data, that is, extracting the most important information. In PCA, this is achieved by selecting the principal components that explain most of the relationships among the variables and, so, reflect the intrinsic dimensionality of data, but little research has been done on autoencoder ability in dimensionality extraction.

In 2016, Wang et al. [20] investigated a possible relation between the number of hidden layer nodes, the performance of autoencoder and the intrinsic dimensionality of data. This study was conducted on MNIST and Olivetti face datasets by recording the change of performance of the classifier when the dimensionality of the projected representation varies. Results of this study showed a possible relation between the hidden nodes, the intrinsic dimensionality of MNIST dataset and the autoencoders accuracy.

Similarly, in this paper we want to investigate this relation in a more systematic way. The aim is to compare PCA and autoencoder ability in dimensionality extraction on different factor-based simulated datasets. Our hypothesis is that autoencoder's representation of data lying in bottleneck layer captures the most important data characteristics and is in relation with data intrinsic dimensionality. So, the performance of autoencoder should be optimal when the number of internal nodes is equal to data dimensionality. More details and results are showed in the next section.

3. A simulation study

In order to investigate the relation between data dimensionality and autoencoder's internal nodes, a simulation approach is chosen, because of the possibility to analyze different scenarios by varying only the selected design-factors. In this section, the simulation study is described in detail, and results are showed.

3.1. Simulation Design and Data Generation

Analyses were conducted on artificial data generated from different factor-based population, using the R package Lavaan [21]. Relationships in the model were set assuming the theoretical path model represented in Figure 4 and then data were simulated considering the given values of the parameters.

The simulation study considered different scenarios, varying the following design-factors: sample size, number of components and number of observed variables. The considered levels for each design-factor are presented in Figure 4. The total number of scenarios obtained from the combination of these levels of the design-factors was equal to 48 (4 sample sizes × 3 number of components × 4 number of observed variables). For each considered scenario, we generated one dataset.

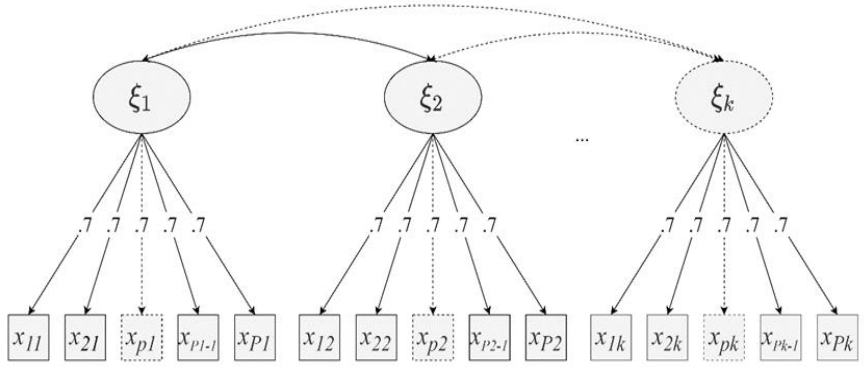
Model used for generating data:			
Generated sample size:	n = 200 n = 500 n = 700 n = 1000	Number of components:	$N_K = 3$ $N_K = 5$ $N_K = 7$
Loadings:	All factor loadings are equal to 0.7	Number of observed variables for each component:	$N_P = 3$ $N_P = 5$ $N_P = 7$ $N_P = 9$

Figure 4: Simulation plan

3.2. Data analysis and Results

Non-linear autoencoders used for dimensionality reduction were implemented in Python using Keras module [22]; the non-linearity of choice was the hyperbolic tangent activation function (tanh) except for the bottleneck and the output layers which used a linear activation function.

All layers but for the bottleneck had the same number of neurons, equal to the number of observed variables. The bottleneck layer's nodes varied from one to the number of observed variables. That is, for a dataset with *n* observed variables, *n* autoencoders were trained on the whole dataset, with neurons in the bottleneck layer varying from 1 to *n*. MSE was computed for each autoencoder.

Weights were initialized based on the uniform distribution suggested by Glorot and Bengio [23] and Adam optimizer was used with 0.0001 learning rate as it offers both fast training and good generalization performance [24].

Finally, PCA was performed using Scikit-learn module in Python [25] and MSE was computed for each possible number of components (from 1 to the number of observed variables).
Figures 5, 6, and 7 show results for three scenarios:

a) 3 components and 9 observed variables.
b) 5 components and 25 observed variables.
c) 7 components and 63 observed variables.

These scenarios are, respectively, the smallest, the medium and the largest among those obtained from all the possible combinations of the chosen design-factors.

Results show that MSE for both Autoencoders and PCA are very similar and are about the same when sample size is sufficiently large. Plots always display a downward curve, starting high on the left, falling rather quickly, and then flattening out at some point: this "elbow" coincides with the intrinsic data dimensionality. This pattern is always repeated, except in models with many observed variables and low sample size. In these cases, autoencoders results don't follow the same trend as PCA and don't provide information about the data dimensionality.

For the maximum number of components, MSE score for PCA is equal to zero, because considering all components data are perfectly reconstructed, and all the variation is retained. MSE scores for autoencoder, when the number of nodes is near or equal to the number of observed variables, are low, but doesn't display the same pattern as PCA and are not equal to zero.

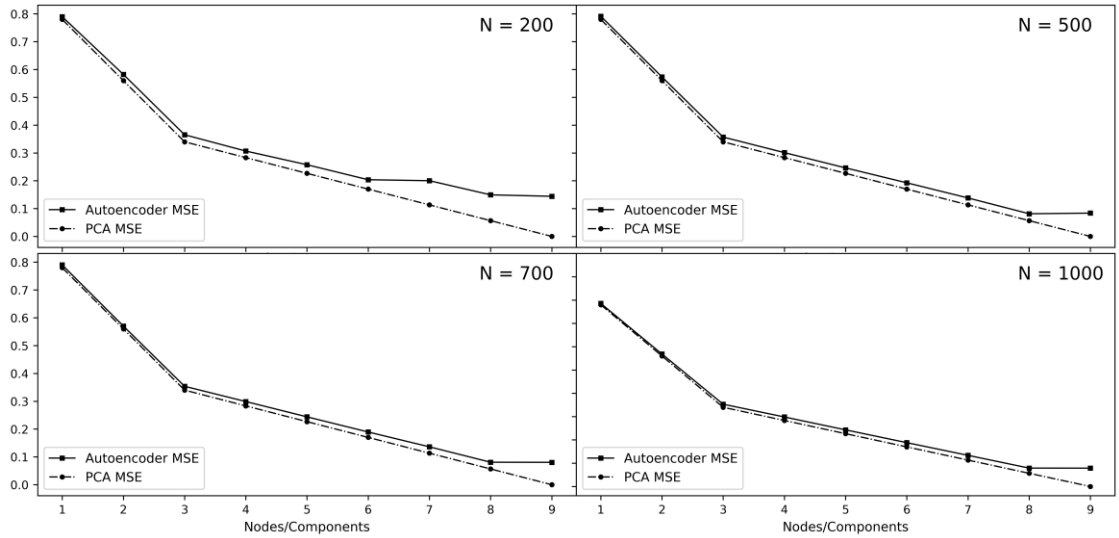


Figure 5. MSE for 3-dimensional simulated dataset with 9 variables and different sample size

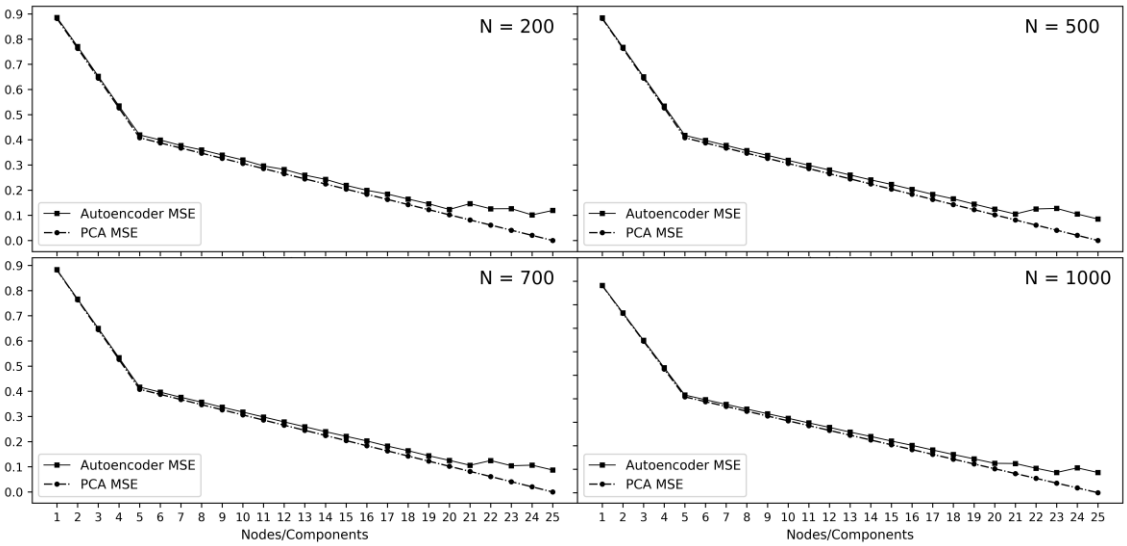


Figure 6. MSE for 5-dimensional simulated dataset with 25 variables and different sample size

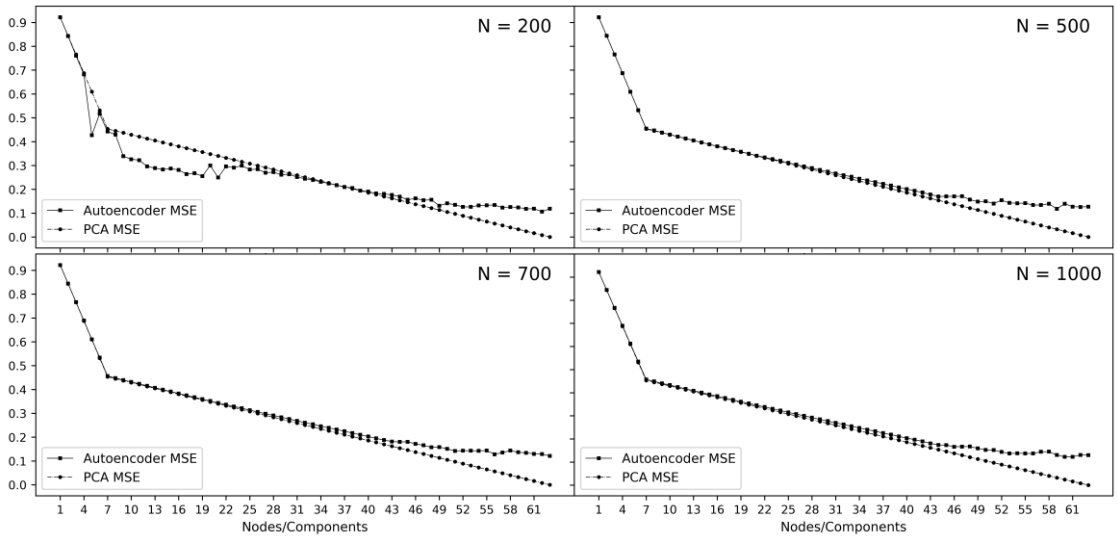


Figure 7. MSE for 7-dimensional simulated dataset with 63 variables and different sample size

4. Discussion and future research

In this work, we have compared PCA and non-linear autoencoder and we have hypothesized a relationship between the number of autoencoder's internal nodes and the intrinsic dimensionality of data. Results shows that autoencoder can perform dimensionality reduction as well as PCA, with an adequate sample size. Furthermore, results show that neurons in internal layer have a relation with the dimensionality of data. In fact, after the "elbow" of the graph, in which the number of neurons coincides with the dimensionality of data, the decrease in the MSE is slower and thus not sufficient to compensate for the increase in complexity. Because the choice of number of hidden neurons is *a priori* choice, it is useful to know that nodes of the bottleneck layer have a relation with data dimensionality.

The most important difference between PCA and autoencoders is that autoencoders can utilize non-linear activation functions at the different layers of the neural network whereas, in PCA, dimensionality reduction is done in a linear transformation. The use of non-linear activation functions is what makes autoencoders a more flexible method for learning patterns in data.

However, in this work, PCA and Autoencoder seems to have the same behavior in dimensionality reduction: even if both methods offer information on the intrinsic dimensionality of data, it is important to consider that relations between simulated variables are linear; therefore, it is likely that a linear method performs better. However, relations met in real world are not always linear: for this reason, a future work will evaluate autoencoder performances on simulated data with non-linear relations between variables. Moreover, future research will also focus on autoencoder performances on real datasets. This future work will allow deeper understanding of the similarities and differences between these methods.

PCA and autoencoders share architectural similarities, but despite this fact, an autoencoder by itself does not have PCA properties. Incorporating some PCA constraints, autoencoder's solution would have the following benefits: a) uniqueness; b) components would be uncorrelated and sorted in descending order of variance; and c) when reducing the data from dimension p to dimension z_k, the first z_j vectors ($z_j < z_k$) would be the same as the solution for reduction from dimension p to z_j [26].

Despite machine learning methods are increasingly prevalent in several areas of psychology [27, 28], autoencoders are absent in psychometric research. Nevertheless, we believe that autoencoder can be used where some traditional methods show their limits. For example, in future research, autoencoders will be applied to the development of short form of psychological test. In fact, despite the potential benefits of using shorter measures, development of short forms shows several limitations: first, development of an abbreviated measure can be a relatively laborious process and second, most short forms of existing measures are not guaranteed to achieve optimality because their developers typically consider only a small fraction of possible alternate forms. In this context, neural networks can help to

automatize and optimize short-form development process [29]. In particular, an autoencoder trained on a long-form of a measure, could be useful in selecting the short form that better reconstructs the long form, among the many possible and alternative short-forms. In this case, keeping the number of hidden neurons equal to the number of dimensions in the original test could help to choose short-forms that have the same dimensionality of the original measure.

In conclusion, despite additional investigations are required, we believe that autoencoders, which are already widely used in other scientific fields, are an interesting alternative to standard PCA also in psychometric models.

5. References

[1] Van Der Maaten, Laurens, Eric Postma, and Jaap Van den Herik. "Dimensionality reduction: a comparative review." Journal of Machine Learning Research 10. 13 (2009): 66-71.

[2] Steger, Michael F. "An illustration of issues in factor extraction and identification of dimensionality in psychological assessment data." Journal of personality Assessment 86.3 (2006): 263-272.

[3] Zebari, Rizgar, et al. "A comprehensive review of dimensionality reduction techniques for feature selection and feature extraction." Journal of Applied Science and Technology Trends 1.2 (2020): 56-70.

[4] Velicer, Wayne F., Cheryl A. Eaton, and Joseph L. Fava. "Construct explication through factor or component analysis: A review and evaluation of alternative procedures for determining the number of factors or components." Problems and solutions in human assessment (2000): 41-71.

[5] Pearson, Karl. "On lines and planes of closest fit to systems of points in space." The London, Edinburgh, and Dublin philosophical magazine and journal of science 2. 11 (1901): 559-572.

[6] Hotelling, Harold. "Analysis of a complex of statistical variables into principal components." Journal of educational psychology 24.6 (1933): 417.

[7] Jolliffe, Ian. "Principal component analysis." Encyclopedia of statistics in behavioral science (2005).

[8] Scholz, Matthias, and Ricardo Vigário. "Nonlinear PCA: a new hierarchical approach." Esann. 2002.

[9] Bourlard, Hervé, and Yves Kamp. "Auto-association by multilayer perceptrons and singular value decomposition." Biological cybernetics 59.4 (1988): 291-294.

[10] Baldi, Pierre, and Kurt Hornik. "Neural networks and principal component analysis: Learning from examples without local minima." Neural networks 2.1 (1989): 53-58.

[11] Sanger, Terence D. "Optimal unsupervised learning in a single-layer linear feedforward neural network." Neural networks 2.6 (1989): 459-473.

[12] Kramer, Mark A. "Nonlinear principal component analysis using autoassociative neural networks." AIChE journal 37.2 (1991): 233-243.

[13] Siwek, Krzysztof, and Stanislaw Osowski. "Autoencoder versus PCA in face recognition." 2017 18th International Conference on Computational Problems of Electrical Engineering (CPEE). IEEE, 2017.

[14] Alkhayrat, Maha, Mohamad Aljnidi, and Kadan Aljoumaa. "A comparative dimensionality reduction study in telecom customer segmentation using deep learning and PCA." Journal of Big Data 7.1 (2020): 1-23.

[15] Henson, Robin K., and J. Kyle Roberts. "Use of exploratory factor analysis in published research: Common errors and some comment on improved practice." Educational and Psychological measurement 66.3 (2006): 393-416.

[16] Bishop, Christopher M. Pattern Recognition and Machine Learning. New York: Springer, 2006.

[17] Y. Bengio, A. Courville and P. Vincent, "Representation Learning: A Review and New Perspectives," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 35, no. 8, pp. 1798-1828, Aug. 2013, doi: 10.1109/TPAMI.2013.50.

[18] Kerschen, Gaetan, and Jean-Claude Golinval. "Feature extraction using auto-associative neural networks." Smart Materials and Structures 13.1 (2003): 211.

[19] Hinton, Geoffrey E., and Ruslan R. Salakhutdinov. "Reducing the dimensionality of data with neural networks." science 313.5786 (2006): 504-507.
[20] Wang, Yasi, Hongxun Yao, and Sicheng Zhao. "Auto-encoder based dimensionality reduction." Neurocomputing 184 (2016): 232-242.
[21] Rosseel, Yves. "Lavaan: An R package for structural equation modeling and more. Version 0.5–12 (BETA)." Journal of statistical software 48.2 (2012): 1-36.
[22] Chollet, François. "keras." (2015).
[23] Glorot, Xavier, and Yoshua Bengio. "Understanding the difficulty of training deep feedforward neural networks." Proceedings of the thirteenth international conference on artificial intelligence and statistics. JMLR Workshop and Conference Proceedings, 2010.
[24] Kingma, Diederik P., and Jimmy Ba. "Adam: A method for stochastic optimization." arXiv preprint arXiv:1412.6980 (2014).
[25] Pedregosa, Fabian, et al. "Scikit-learn: Machine learning in Python." the Journal of machine Learning research 12 (2011): 2825-2830.
[26] Plaut, Elad. "From principal subspaces to principal components with linear autoencoders." arXiv preprint arXiv:1804.10253 (2018).
[27] Dwyer, Dominic B., Peter Falkai, and Nikolaos Koutsouleris. "Machine learning approaches for clinical psychology and psychiatry." Annual review of clinical psychology 14 (2018): 91-118.
[28] Dolce, Pasquale, Davide Marocco, Mauro N. Maldonato, and Raffaele Sperandeo. "Toward a Machine Learning Predictive-Oriented Approach to Complement Explanatory Modeling. An Application for Evaluating Psychopathological Traits Based on Affective Neurosciences and Phenomenology." Frontiers in psychology 11 (2020): 446.
[29] Gonzalez, Oscar. "Psychometric and machine learning approaches to reduce the length of scales." Multivariate Behavioral Research (2020): 1-17.