Computer Assisted Methods in Engineering and Science, **22**: 141–151, 2015.
Copyright © 2015 by Institute of Fundamental Technological Research, Polish Academy of Sciences

Some methods of pre-processing input data for neural networks

Krystyna Kuźniar, Maciej Zając
Pedagogical University of Cracow, Institute of Technology
Podchorążych 2, 30-084 Kraków, Poland
e-mail: kkuzniar@up.krakow.pl, mjzajac@up.krakow.pl

Two techniques of data pre-processing for neural networks are considered in this paper: (i) data compression with the application of the principal component analysis method, and (ii) various forms of data scaling. The novelty of this paper is associated with compressed input data scaling by the rotation (by the "stretching") in neural network. This approach can be treated as the new proposition for data pre-processing techniques. The influence of various types of input data pre-processing on the accuracy of neural network results is discussed by using numerical examples for the cases of natural frequency predictions of horizontal vibrations of load-bearing walls. It is concluded that a significant reduction in the neural network prediction errors is possible by conducting the appropriate input data transformation.

Keywords: neural networks, data pre-processing, input data, principal component analysis method, data scaling.

1. Introduction

Various explanations could be given to the role of data pre-processing and its need, especially in the case of neural network input data [1, 3, 11, 12, 17]. Although in a lot of cases, the pre-processing of neural network input data is not needed from the mathematical point of view, it can improve the neural network training process. Moreover, the form of pre-processing applied to the data is a very important factor in determining the success of a practical application of neural networks [1, 3, 7, 11]. The primary purpose of data pre-processing is to modify the input variables so they can better match the predicted output. The main purpose of neural network data transformation is to modify the distribution of the network input or output parameters.

The data transformations that are the most commonly applied in neural networks can be categorized into three groups:

- linear transformation (mostly scaling to the ranges of (0,1) or (−1,1)),
- statistical standardization (using deviation from the mean),
- various other mathematical functions.

Neural network error differences between two cases of transformations (linear transformation and distribution transformation) of data are presented, e.g., in [16] for two problems. A lot of papers discuss the results of using different pre-processing methods of neural network data in the cases of various fields of practical engineering problems, e.g., [2, 5, 6, 8, 15, 18].

Two techniques of neural network data pre-processing are analysed in this paper. The first one is associated with data compression (reduction of the dimensionality) with the application of the principal component analysis method [1, 4, 19]. The other methods of pre-processing are focused on various forms of data scaling [11, 17]. The novelty of this paper is associated with scaling by the rotation (by the "stretching") of compressed neural network input data. This approach can be treated as a new proposition for data pre-processing techniques.

As an illustration, the numerical results of the investigations on the influence of the above mentioned data pre-processing on the accuracy of the neural network prediction of the natural frequencies of horizontal vibrations of load-bearing walls are presented.

2. Applied techniques of data pre-processing

2.1. Principal component analysis

The compression of neural network input data (as well as output data) makes it possible to design "smaller" neural networks than those without data compression, i.e., a reduction in the number of network parameters.

The principal component analysis (PCA) is one of the methods applied to reduce the neural network input space dimension [4]. The reduction is achieved by transforming the data into a new set of variables, called principal components. In addition, it is possible by means of eigenanalysis to select only those principal components which preserve the most important features of the original space [4].

The PCA method relates to the linear transformation of process description in the form of N-elements vector $\mathbf{x}$ into K – elements vector $\mathbf{y}$, using matrix $\mathbf{W} \in R^K \times R^N$:

$$\mathbf{y} = \mathbf{W}\mathbf{x}. \tag{1}$$

Matrix $\mathbf{W}$ could be defined using eigenvectors $\mathbf{w}_i$ ($\mathbf{w}_i$ – successive eigenvectors, $i = 1, 2, \ldots, K$; $K < N$) of the autocorrelation matrix for vectors $\mathbf{x}$ (vector $\mathbf{x}$ could be the neural network input vector, for instance):

$$\mathbf{W} = [\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_K]^T. \tag{2}$$

Since $K < N$, the size of vector $\mathbf{y}$ is reduced as compared to $\mathbf{x}$. Thus, the PCA transformation changes the large number of input data into a set of components according to their importance.

The principal components are the projections of original input vectors $\mathbf{x}$ on principal directions related to eigenvectors. For example, the case of 2D output space is illustratively shown in Fig. 1. The principal directions 1 and 2 are associated with the eigenvectors of maximal and minimal variances of data, respectively.

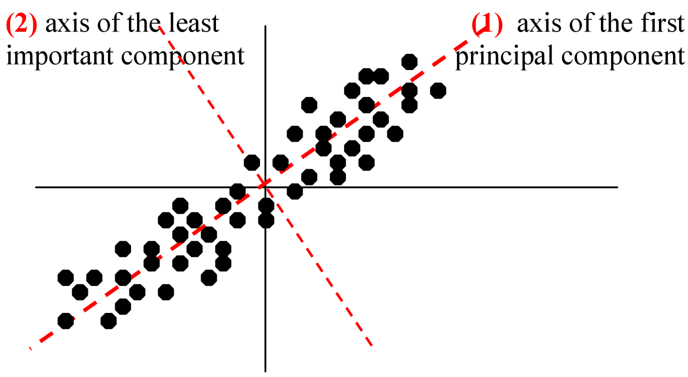

Fig. 1. Illustration of the principal components for data from two-dimensional space (based on [9]).

Two variants of PCA method application were considered: (i) input data transformation into principal components using "local compression" (LC), and (ii) input data transformation into principal components using "global compression" (GC). In the first one, for each of the neural network input vectors, the autocorrelation matrix was set up and the linear eigenvalue problem was analysed separately. In the second variant, only one autocorrelation matrix associated with all of the input vectors was computed.

2.2. Scaling of data

The main goal of data scaling is the transformation of the actual (experimental or obtained from calculations) data into dimensionless data or into required space (range). Sometimes, the necessity of neural network data scaling is caused by the choice of activation function. Such a situation takes place, e.g., in the case of output data from the neural network with sigmoid activation functions adopted in the output layer.

A number of input variable transformations by scaling are proposed in this paper. The following formulae are discussed:

a) scaling to the interval (0.1 – 0.9)

$$\text{S1:} \qquad x_s = \frac{0.9 \cdot (x - x_{\min}) - 0.1 \cdot (x - x_{\max})}{x_{\max} - x_{\min}}, \tag{3}$$

b) dividing by the maximum value in the real range of data (or another value)

$$\text{S2:} \qquad x_s = \frac{x}{x_{\max}}, \tag{4}$$

c) using polynomial functions

$$\text{S3:} \qquad x_s = x^{\alpha}, \tag{5}$$

d) using exponential functions

$$\text{S4:} \qquad x_s = e^{\alpha x}, \tag{6}$$

where x_s – scaled value, x – real value, $x_{\min}$ and $x_{\max}$ – minimal and maximal values in the real range respectively, α – adopted constant.

3. Illustrative examples

3.1. Analysed problem

The analysed problem is related to the neural prediction of the first natural frequencies of horizontal vibrations of modified load-bearing walls [10].

A problem with modernisation of structures, a result of contemporary occupants' expectations when it comes to living standards, appears in many older, existing apartment buildings (e.g., prefabricated buildings). New door openings and the widening of existing door openings in the load-bearing walls are some examples of wall modifications. Naturally, wall geometrical modifications cause changes in wall dynamic properties, including natural frequencies of vibrations, among other things. Computation of the natural frequencies of the modified structures is usually necessary in engineering practice in the cases of dynamic influences, as can be seen especially, for instance, in building design in seismic areas.

Typical medium-height reinforced concrete (Young's modulus – 29 GPa, Poisson's ratio – 0.17, density – 2500 kg/m^3) load-bearing walls were considered with a 2.7 m, 5.4 m, 11.7 m width, a 14 m (five storeys $\times$ 2.8 m) height, and a 0.14 m thickness. The small and large changes in the wall stiffness and mass, resulting from the size and position of new door openings, were taken into account. A series of door openings, one above the other on all of the storeys (system door openings) were considered. They were "shifted" from the edge of the wall with a step of 0.3 m. The widths of door openings were taken from the range of 0.9 m – 4.8 m with a step of 0.3 m. The example of one of the analysed walls (with finite element mesh) is schematically shown in Fig. 2a.

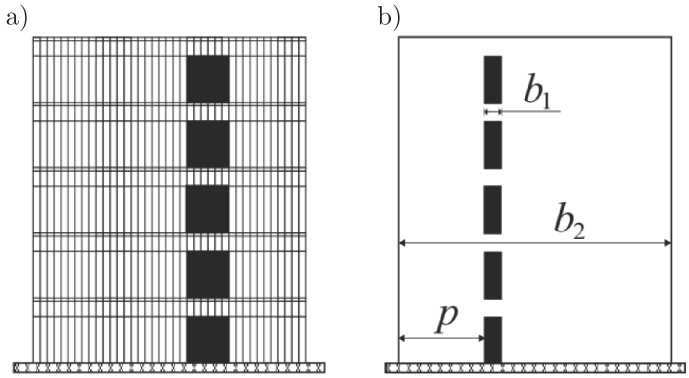

Fig. 2. a) The example of the wall with a modification in the form of system door openings – a series of door openings, one above the other on all of the storeys; b) the parameters considered in the neural network input vectors.

The influence of the type of data pre-processing technique on the accuracy of the neural prediction of frequencies was investigated.

3.2. Computations using neural networks

Back-propagation neural networks (BPNNs) were applied for the computation of the first natural frequencies (f_1 [Hz]) of horizontal vibrations of the walls. All the neural networks were trained using the Levenberg-Marquadt learning algorithm [13], sigmoid activation function in the input and hidden layers, and linear activation function in the output layer. One hidden layer was proposed for each of the networks.

Neural network input vectors were composed of the following parameters (Fig. 2b): p – the coordinate of the door opening position, b_1 – the door opening width, b_2 – the wall width, f_{1S}, f_{2S} – the first and second natural frequencies of the wall without door openings. The first natural frequencies (f_1) were the outputs of the neural networks.

The finite element method [14] was applied to generate patterns of neural networks according to the cases of modifications. The factor of symmetry in the cases of door opening positions was taken into consideration and a total number of 215 patterns were prepared. They were divided into three sets: training (about 60% – $L = 129$ patterns), validating (about 20% – $V = 44$ patterns) and testing (about 20% – $T = 42$ patterns).

The accuracy of neural networks with various input data pre-processing routines was estimated by mean square error (MSE) and relative errors (ep):

$$MSE = \frac{1}{Q}\sum_{p=1}^{Q}(z^{(p)} - y^{(p)})^2, \tag{7}$$

$$ep = |1 - y^{(p)}/z^{(p)}| \cdot 100\%, \tag{8}$$

$$ep\max = \max_p ep, \tag{9}$$

$$ep\,\text{average} = \frac{1}{Q}\sum_{p=1}^{Q} ep, \tag{10}$$

where $z^{(p)}$, $y^{(p)}$– target and neurally computed outputs for p-th pattern, and $Q = L, V, T$ – number of the learning (L), validating (V) and testing (T) patterns, respectively.

The numerical efficiency of the networks was also evaluated by the coefficient of linear correlation for the testing (r_T) and the success ratio (SR). The SR function enabled to estimate what percentage of patterns SR (%) gives the neural prediction with the error not greater than ep (%)

3.3. Results

The influence of the neural network input data scaling S1 and S2 on the accuracy of the neural prediction of the modified wall frequencies could be significant, which is visible by comparing the errors of neural networks collected in Table 1, as well as the curves of the SR in Fig. 3.

Table 1. Neural network errors in the identification of natural wall frequencies with the application of S1 and S2 data scaling.

Steps of the input data pre-processing	BPNN structure	*MSE*			*ep* max [%]			*ep* average [%]			r_T
		L	V	T	L	V	T	L	V	T	
Without pre-processing	5-5-1	0.00098	0.00795	0.01122	1.38	3.26	4.40	0.29	0.56	0.58	0.9993
S1	5-5-1	0.00039	0.00162	0.00107	1.08	2.25	1.63	0.16	0.31	0.25	0.9999
S2	5-5-1	0.00146	0.00571	0.00330	2.07	3.29	1.70	0.32	0.58	0.38	0.9998

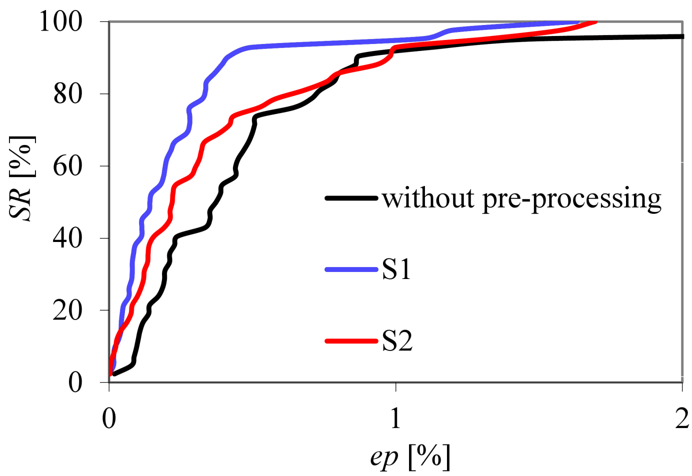

Fig. 3. A comparison of the success ratio (SR) for the approximations of neural networks with the application of S1 and S2 data scaling (testing patterns).

Looking at the columns in Table 1 with relative errors, it is clear that the values of the maximum as well as the average relative testing errors were reduced by over 50% using data scaling. In turn, the reduction of MSE for testing is even greater than for relative errors.

It is also visible (Fig. 3) that 100% of the testing patterns were obtained with relative errors less than 2%, in the cases of neural networks with scaled input vectors.

From the results of the data pre-processing using PCA method with GC as well as with LC, it is clear that the first principal component reaches more than 99% of the total variance of data in all the considered cases. Then, the first principal component is predominant. Therefore, the five parameters (p, b_1, b_2, f_{1S}, f_{2S}) from the network input vector could be compressed to the first principal component only. As a result, a very small network (1-3-1) could be designed. But there are some “difficulties” for neural prediction of the ambiguous relationship between the first principal components of the modified wall parameters and the corresponding wall frequencies, see Fig. 4.

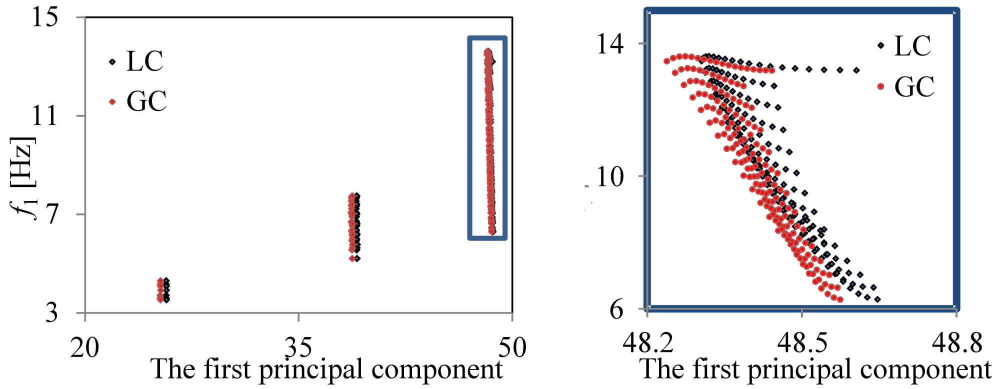

Fig. 4. The first principal components of the wall parameters vs. the wall frequencies (f_1).

This difficulty is confirmed by the results obtained from networks with the application of LC and GC data pre-processing (input vectors are compressed), see Table 2 and Fig. 5. Therefore, the rise in values of neural network errors, in the cases of compressed input application, could be produced not only by the effect of lossy compression (using the PCA method) but first of all by the above-mentioned ambiguous relationship.

Table 2. Neural networks errors in the identification of natural wall frequencies with the application of LC and GC data pre-processing.

Steps of the input data pre-processing	BPNN structure	*MSE*			*ep* max [%]			*ep* average [%]			r_T
		L	*V*	*T*	*L*	*V*	*T*	*L*	*V*	*T*	
Without pre-processing	5-5-1	0.00098	0.00795	0.01122	1.38	3.26	4.40	0.29	0.56	0.58	0.9993
LC	1-3-1	3.48180	3.89550	4.59810	58.26	33.54	59.70	16.9	14.6	18.6	0.6514
GC	1-3-1	0.40886	0.87529	0.90390	27.05	25.69	28.58	5.51	6.42	6.32	0.9429

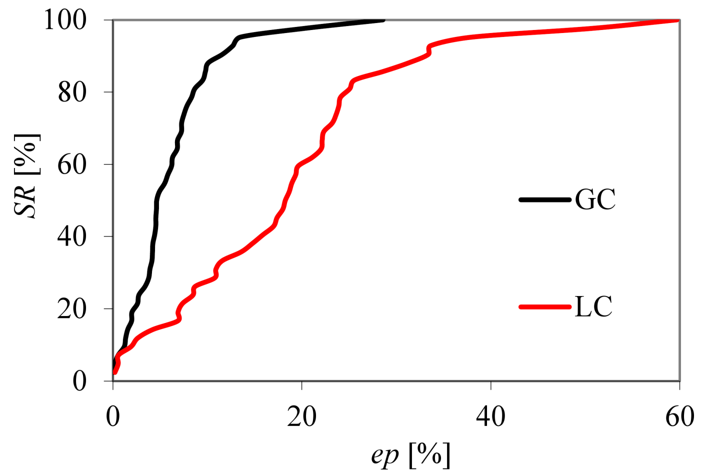

Fig. 5. A comparison of the success ratio (*SR*) for the approximations of neural networks with the application of LC and GC data pre-processing.

Although input data after LC and GC are very similar (Fig. 4), the differences between the results of the neural prediction for the two kinds of pre-processing methods are significant. For example, the relative average testing errors differ about 66% (Table 2).

Looking at the *SR* graphs in Fig. 5, it is visible that in the case of the neural network with LC of input vectors, as much as 40% of the patterns are computed with the relative errors greater than 20%. The main reason for such poor results in the case of LC could be connected with the calculations of the autocorrelation matrix. In the case of GC there is only one autocorrelation matrix, thus different input vectors are compressed to different principal components; whereas, in the case of LC, for every input vector its own autocorrelation matrix is constructed. Therefore, two input vectors, representing different information but having the same elements of vector, will compress to the same value as principal component, which makes neural prediction more difficult. The example of such a situation is shown in Fig. 6.

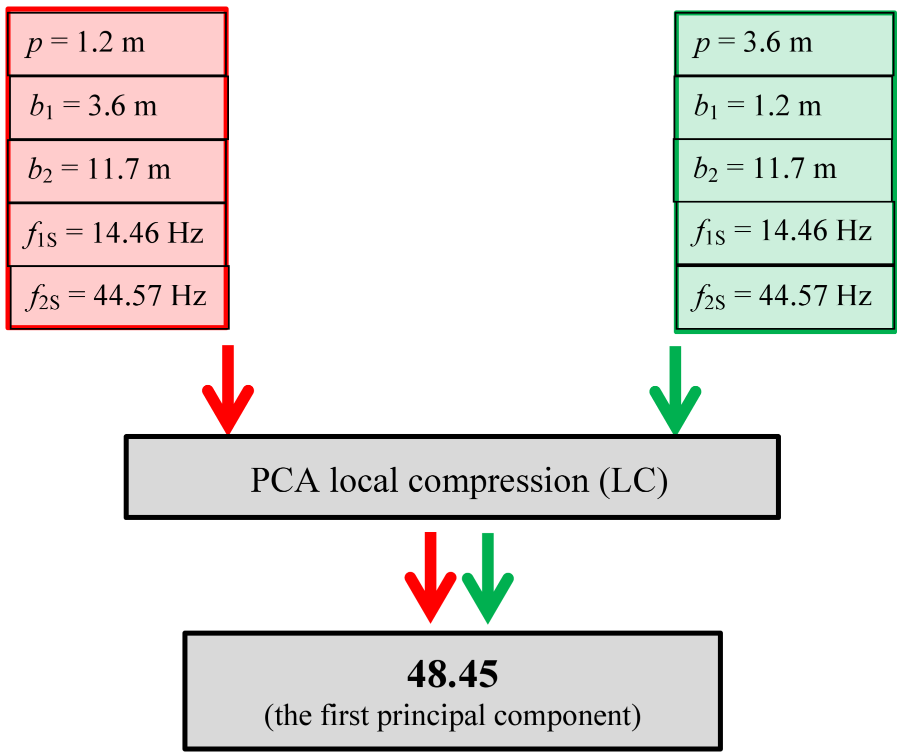

Fig. 6. The example of local compression (LC) of two different neural network input vectors to the same value of the first principal component.

The next proposed step in the data pre-processing concerns the various types of scaling of compressed input vectors. The main goal of this operation is the rotation and "stretching" of the input data to make the relation input-output more unambiguous and, as the result, to improve the accuracy of neural network prediction.

Figure 7 shows, as the example, the influence of the sequence of various types of scaling methods (S1, S3, S4), together with GC of input data on the relationship: the scaled first principal component of input data – the first natural frequency, in the case of the 11.7 m wall. The black dots represent GC data without scaling, the red and green dots – scaling while using a polynomial function with different alpha parameters, and the blue dots – scaling while using an exponential function.

The results presented in Table 3 and in Fig. 8 confirm that the input data compression supported by the application of even such simple proposed scaling methods (S3 – polynomial function, S4 – exponential function) can improve neural network prediction.

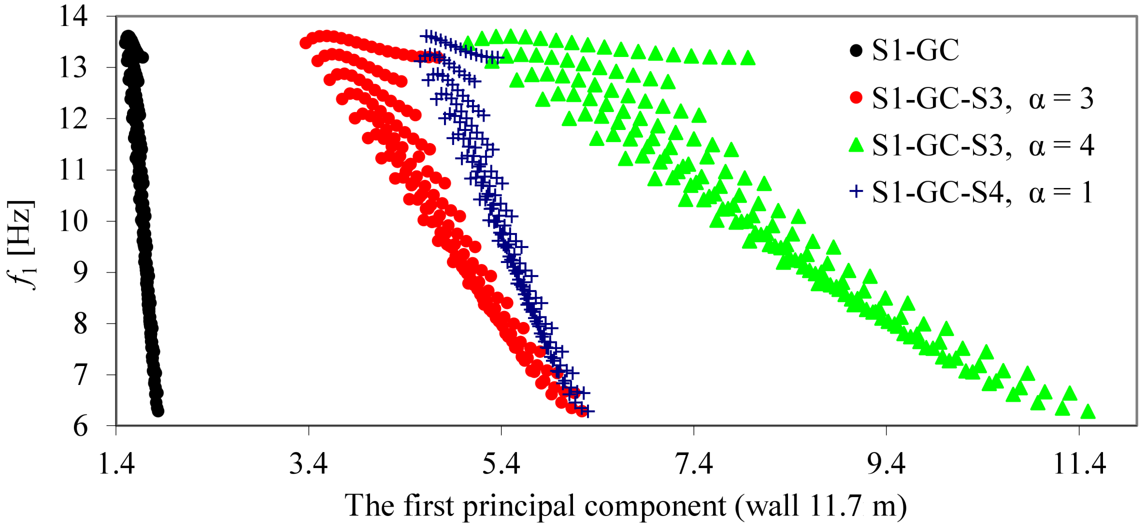

Fig. 7. The relationship: the scaled first principal component of input data – the first natural frequency, in the case of the 11.7 m wall.

Table 3. Neural networks errors in the identification of natural wall frequencies with the application of GC and scaling data pre-processing.

Steps of the input data pre-processing	BPNN structure	*MSE*			*ep* max [%]			*ep* average [%]			r_T
		L	*V*	*T*	*L*	*V*	*T*	*L*	*V*	*T*	
GC	1-3-1	0.40886	0.87529	0.90390	27.05	25.69	28.58	5.51	6.42	6.32	0.9429
S1-GC-S3 ($\alpha = 4$)	1-3-1	0.20809	0.54944	0.55494	18.86	19.94	22.55	3.73	5.25	4.89	0.9653
S1-GC-S4 ($\alpha = 1$)	1-3-1	0.30217	0.58572	0.57076	29.73	21.12	22.32	4.73	5.70	5.08	0.9641
S2-GC-S3 ($\alpha = 3$)	1-3-1	0.24310	0.50120	0.51222	27.02	19.26	21.57	4.09	5.07	4.68	0.9686
S2-GC-S4 ($\alpha = 1$)	1-3-1	0.31102	0.58046	0.55122	30.55	23.33	21.63	4.79	5.88	4.99	0.9650

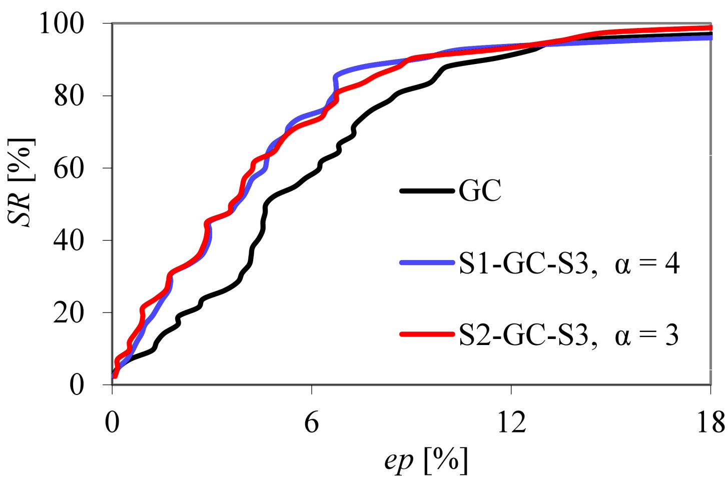

Fig. 8. A comparison of the success ratio (*SR*) for the approximations of neural networks with the application of GC and scaling data pre-processing (testing).

This effect is also clearly visible in a comparison of the values of the first natural frequencies computed by BPNNs with the target values presented in Fig. 9. It is concluded that in the cases of neural networks in which compressed input vectors are additionally scaling, the accuracy of the

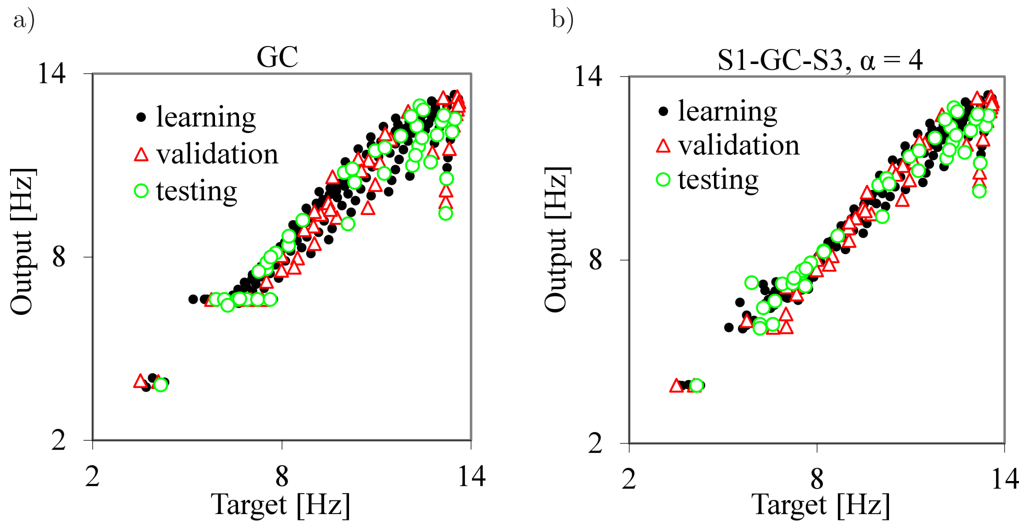

Fig. 9. A comparison of the values of the first natural frequencies computed by BPNNs with target values.

obtained results is better (the positions of points are closer to diagonal, Fig. 9). In Fig. 9, the sets of learning, validation and testing patterns are shown separately.

Additionally, in Fig. 10, the actual outputs (the target values) and predicted values of the first natural frequencies, only for the testing patterns, are compared in the cases of some discussed neural networks. It is visible that in the cases of scaling of the data, predicted frequencies are closer to actual outputs, especially for the higher values of frequencies.

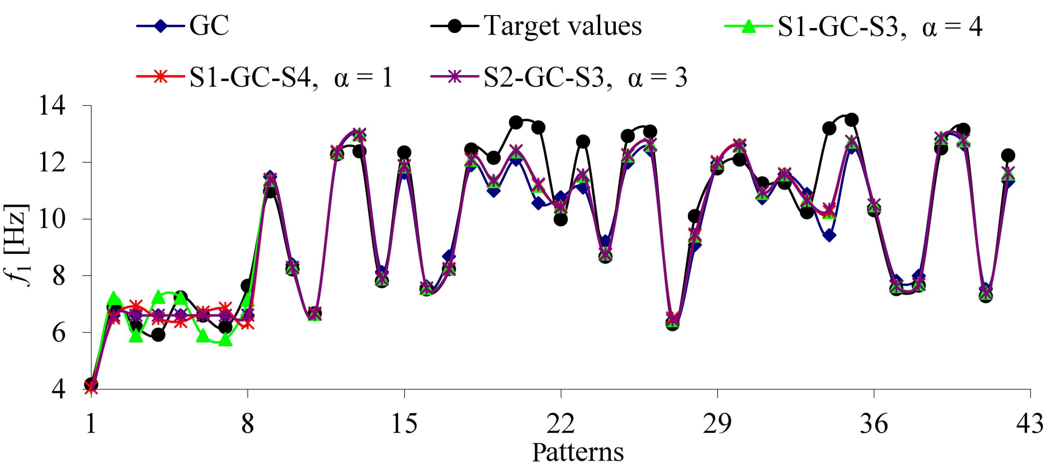

Fig. 10. A comparison of the actual outputs (target values) and predicted values of the first natural frequencies (only for the testing patterns).

Thus, in all the proposed and discussed cases of neural network data pre-processing, an improvement in the accuracy of the computed values of the first natural frequencies of the walls is observed, most importantly, for the testing patterns. For instance, pre-processing by scaling of input vectors with the full information of data (without the compression) produces the improvement in the testing error *MSE* up to 91%, see Table 4. Also, in the cases of compressions using PCA together with the additional pre-processing of data by scaling, the obtained results confirm an improvement in accuracy. For example, in the case of the neural network with S1-GC-S3 ($\alpha = 3$) data pre-processing, the reduction in the relative testing error reaches up to 26%, see Table 4.

Table 4. A reduction of the prediction errors in the cases of selected neural networks with data pre-processing.

Error		Neural network error reduction (%)	
		NN with scaling S1 vs. NN without data pre-processing	NN with S1-GC-S3 ($\alpha = 3$) data pre-processing vs. NN with PCA GC
MSE	*L*	60.2	40.5
	V	79.6	42.7
	T	90.5	43.3
ep max [%]	*L*	21.7	0.10
	V	31.0	25.0
	T	63.0	24.5
ep average [%]	*L*	44.8	25.8
	V	44.6	21.0
	T	56.9	26.0

4. Conclusions

The influence of neural network input data scaling on the accuracy of the neural prediction of the modified wall frequencies could be significant.

The PCA pre-processing method enables us to map the input data into a space of lower dimensionality (compression of data). It is worth mentioning that GC gives much better results than LC.

In the case of the special relationship between the compressed data and the neural network output (as it is visible, for example, in the case of the first principal components of the modified wall parameters and the wall frequencies), scaling by the rotation (by the "stretching") of the compressed neural network input data is advisable.

References

[1] C.M. Bishop. *Neural networks for pattern recognition*. Oxford, Clarendon Press, 1996.

[2] J.J. Davis, A.J. Clark. Data preprocessing for anomaly based network intrusion detection: a review. *Computers and Security*, **30**: 1–23, 2011.

[3] A. Famili, W-M. Shen, R. Weber, E. Simoudis. Data preprocessing and intelligent data analysis. *Intelligent Data Analysis*, **1**: 3–23, 1997.

[4] S. Haykin. *Neural networks – a comprehensive foundation*. 2nd Edition, Prentice Hall Intern. Inc., Upper Saddle River, NY, 1999.

[5] D-S. Joo, D-J. Choi, H. Park. The effects of data preprocessing in the determination of coagulant dosing rate. *Water Research*, **34**(13): 3295–3302, 2000.

[6] I. Klevecka, J. Lelis. Pre-processing of input data of neural networks: the case of forecasting telecommunication network traffic. *Telektronikk*, **104**(3/4): 168–178, 2008.

[7] P. Koprinkova, M. Petrova. Data-scaling problems in neural-network training. *Engineering Applications of Artificial Intelligence*, **12**: 281–296, 1999.

[8] K. Kuźniar, Z. Waszczyszyn. Neural analysis of vibration problems of real flat buildings and data pre-processing. *Engineering Structures*, **24**: 1327–1335, 2002.

[9] K. Kuźniar, Z. Waszczyszyn. Neural networks for the simulation and identification analysis of buildings subjected to paraseismic excitations. Chapter XVI in: N. Lagaros and Y. Tsompanakis [Eds.], *Intelligent computational paradigms in earthquake engineering*, Idea Group, Hershey, PA, USA, 2007, 393–432.

[10] K. Kuźniar, M. Zając. Data pre-processing in the neural network identification of the modified walls natural frequencies. Proceedings of the 19th International Conference on Computer Methods in Mechanics CMM-2011, Warszawa, 9–12 May, pp. 295–296, 2011.

[11] A. Marciniak, J. Korbicz, J. Kuś. Data pre-processing. [in:] W. Duch, J. Korbicz, L. Rutkowski, R. Tadeusiewicz [Eds.], Vol. 6, Biocybernetics and Biomedical Engineering 2000, Neural Networks. Exit, Warszawa, 2000 [in Polish].

[12] T. Masters. Practical neural networks recipes in C++, Polish ed. by WNT, Warszawa, 1996.
[13] Neural network toolbox for use with Matlab 7.6.0.324, User's Guide, 2008.
[14] Release 11.0, Documentation for Ansys, 2007.
[15] M. Shanker, M.Y. Hu, M.S. Hung. Effect of data standardization on neural network training. *Omega International Journal of Management Science*, **24**(4): 385–397, 1996.
[16] J.J. Shi. Reducing prediction error by transforming input data for neural networks. *Journal of Computing in Civil Engineering*, **14**(2): 109–116, 2000.
[17] Z. Waszczyszyn Advances of soft computing in engineering. CISM Courses and Lectures, vol. 512, Springer, Wien/New York, 2010.
[18] Z. Waszczyszyn. Artificial neural networks in civil and structural engineering: ten years of research in Poland. *Computer Assisted Mechanics and Engineering Sciences*, **13**(4): 489–512, 2006.
[19] Z. Waszczyszyn. Artificial neural networks in civil engineering: another five years of research in Poland. *Computer Assisted Mechanics and Engineering Sciences*, **18**(3): 131–146, 2011.