Learning shape-preserving autoencoder for the reconstruction of functional data from noisy observations

Adam Krzyżak[1][0000−0003−0766−2659] Wojciech Rafajłowicz[2][0000−0003−4347−1358]
Ewaryst Rafajłowicz[3][0000−0003−4347−1358]

[1] krzyzak@cs.concordia.ca Department of Computer Science and Software Engineering, Concordia University, 1455 De Maisonneuve Blvd. West, Montreal, Quebec, Canada H3G 1M8 and Department of Electrical Engineering, Westpomeranian University of Technology (WUT), Szczecin, Poland **
[2] wojciech.rafajlowicz@pwr.edu.pl Faculty of Information and Communication Technology, Wroclaw University of Science and Technology, Wrocław, Poland
[3] ewaryst.rafajlowicz@pwr.edu.pl as above

Abstract. We propose a new autoencoder preserving input functions' general shape (monotonicity, convexity) after their reconstruction without imposing a priori constraints. These properties are inherent to the coefficients of the Bernstein-Durrmeyer polynomials that serve here as theoretical descriptors. Their estimates, computed from noisy observations by the coder, play the role of latent variables. The approach is purely nonparametric, i.e., no prior finite-dimensional model is assumed. The answer to the question of how many latent variables should be used for an acceptable reconstruction accuracy of a family of functional data is inferred from learning based on the proposed approximation of the Akaike Information Criterion. A distinguishing feature of this autoencoder is that the coder and encoder are designed as precomputed and stored matrices with the Bernstein polynomial entries. Thus, after selecting the number of latent variables, the autoencoder usage has a low computational complexity since it is linear with respect to observations. The proposed computational algorithms are tested on real data arising in mechanical engineering when control of damping vibrations is necessary.

Keywords: Functional data · Shape-preservation · Autoencoder design · Bernstein-Durrmeyer polynomials · Nonparametric estimation

1 Introduction

An autoencoder (AE) is a neural network that tries to replicate its input to an output. It consists of a coding algorithm that transforms input vectors into a latent vector of a smaller dimension, which is supposed to compress the input information into meaningful form. The latter can be stored and/or transmitted

** Part of this research was carried out by the first author during his visit of WUT while on sabbatical leave from Concordia University.

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36021-3_26

to an encoder that tries to restore the input vector. The idea of reconstructing functions from a smaller number of descriptors that play role of the latent variables arose very early [12], even if the term "autoencoder" was not in common usage at a time.

Autoencoders have been first introduced by Rumelhart, Hinton and Williams in [20]. Their goal was to learn internal informative representation of the data useful in various applications such as clustering and principal component analysis. If encoder and decoder are linear, then latent representation of the resulting linear autoencoder [1] performs Principal Component Analysis (PCA) [16]. This demonstrates that autoencoder is generalization of PCA, which yields latent space in a form of low-dimensional non-linear manifold rather than low-dimensional hyperplane.

Simple autoencoders have a tendency to reconstruct inputs accurately rather than to build latent informative representation. To prevent this undesirable effect additional regularization has been introduced in autoencoders. Sparse autoencoders [14] use L_1 regularization, which induces sparsity of latent representation. Another approach is based on Kullback-Leibler divergence, which is a distance measure between probability distributions. Denoising Autoencoders [23] carry out regularization by removing additive Gaussian noise added to the input. In Denoising Autoencoders the emphasis is on making them resistant to perturbation of the input, while Contractive Autoencoders [19] put less emphasis on features, which do not play crucial role in decoder reconstruction activity. A real breakthrough in recent years came with the introduction of Variational Autoencoders (VAE) [8], [2], [11].

Recently, the shape-sensitivity approaches were intensely investigated. Their focus is on estimating descriptors from a signal derivative. An sample of papers representing this direction of research includes [9], [15], [10], among others,

The properties of the Bernstein polynomials are well known [13]. Their version exploited here is known as the Bernstein-Durrmeyer polynomials [4], [7]. In [17] BDP proved their usefulness for nonparametric regression estimation.

Advantages of the proposed autoencoder The aim of the proposed autoencoder is to reconstruct functions (signals) from observations corrupted by random errors. The empirical version of the Bernstein-Durrmeyer polynomials (BDP) coefficients are used as the latent variables (empirical descriptors) of the autoencoder. Such empirical descriptors proved their usefulness in pattern recognition problems as features of classifiers [18]. Here, we use them for reconstructing noisy signals.

The advantages of the proposed autoencoder can be summarized as follows.

1. Descriptors preserve the monotonicity and convexity of a function to be reconstructed. These properties manifest themselves automatically, i.e., only when they are present in the input function.
2. At the training phase, the number of latent variables of the autoencoder is selected by unsupervised learning.
3. This selection is carried out by the nonlinear algorithm but after its completion the reconstruction process itself has linear computational complexity.

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36021-3_26

4. There is no need to apply prefiltering or a regularization since BDP are sufficiently "stiff". This simplicity is obtained at a cost of reduced rate of restoration accuracy, but this issue is beyond the scope of this paper.

Shape-preserving properties of the descriptors, when applied to contour parts, can be useful in their understanding [22].

Notice that our autoencoder provides an explicit representation of signals. An appealing alternative is the approach known under the acronym SIREN (see [21] and followers). It offers an implicit representation of signals, images, and even partial differential equations by training a cellular neural network with periodic activation functions. As one can observe from many examples, such networks also have good shape-preserving properties but for simpler tasks with explicitly given noisy observations our approach is sufficiently reliable.

2 Derivation of the proposed autoencoder

This section introduces the proposed autoencoder. The proposed descriptors play the role of its latent variables.

Autoencoder input vectors Autoencoder inputs consists of vectors $\mathbf{y} = [y_1, y_2, \ldots, y_n]^{tr}$, where ^{tr} denotes transposition. They can arise as observations y_i's of function f at equidistant points t_i, $i = 1, 2, \ldots, n$ corrupted by random errors ϵ_i i.e.,

$$y_i = f(t_i) + \epsilon_i, \quad \text{or} \quad y_i = f_i + \epsilon_i, \quad i = 1, 2, \ldots, n, \tag{1}$$

or y_i's can arise as noisy observations of a time-series f_i's.

Let $\mathbb{E}$ be the expectation of a random variable, while $\mathbb{V}ar$ denotes its variance. Then, the classic assumptions: $\mathbb{E}(\epsilon_i) = 0$, $\mathbb{V}ar(\epsilon_i) = \sigma^2 < \infty$, $i = 1, 2, \ldots, n$, $\mathbb{E}(\epsilon_i\, \epsilon_j) = 0$, $i \neq j$, $i, j = 1, 2, \ldots, n$ are imposed, assuming that both σ^2 and the probability distribution of all ϵ_i's are unknown. The only exception from the latter assumption is made when we derive a likelihood function for training the encoder.

For simplicity all functions f considered in this paper are assumed to be defined on $T = [0, 1]$. This interval is covered by the subintervals $T_i = (t_{i-1}, t_i]$, $t_i = i/n$, $i = 1, 2, \ldots, n$ of the lengths $\Delta_n = 1/n$, where $t_0 = 0$.

When a time series is discussed, then its unobserved values f_i's and observations y_i's are also considered to be re-scaled to T interval and associated with points t_i's.

For learning the autoencoder the following sequence $\mathcal{L}_L$ of length $L > 1$ is given: $\mathcal{L}_L \stackrel{def}{=} \{\mathbf{y}^{(1)}, \mathbf{y}^{(2)}, \ldots, \mathbf{y}^{(L)}\}$, as the only available information, where $\mathbf{y}^{(l)}$'s are n-dimensional column vectors that arise as observations of functions $f_l \in C^0(T)$, $l = 1, 2, \ldots, L$, performed according to (1), where $C^0(T)$ denotes the class of all functions continuous on T.

Autoencoder Let $(N+1)$ be the number of latent variables (descriptors) of the autoencoder, where N has to be selected from the range $0 \leq N \leq N_{max} \leq n$ of integers and let $\mathbf{B}_N$ denote $(N + 1) \times n$ matrix with the following rows:

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36021-3_26

$[B_k^{(N)}(t_1),\, B_k^{(N)}(t_2), \ldots,\, B_k^{(N)}(t_n)]$, $k = 0, 1, \ldots, N$, where $B_k^{(N)}(t)$ k denotes the Bernstein polynomial of order $N \geq k$. which is given by $B_k^{(N)}(t) = \binom{N}{k} t^k (1-t)^{N-k}$, $t \in T$, $k = 0, 1, \ldots, N$, where, for simplicity, we assume that $B_k^{(N)}(t) \equiv 0$, if $k < 0$ or $k > N$. In computations, it is more convenient to use its well-known recurrent version.

Then, the general formulation of the autoencoder has the following form:

$$\hat{\mathbf{y}} \;=\; (N+1)\, \Delta_n\, \mathbf{B}_{N_{max}}^{tr}\, \mathbf{I}_{N_{max}}(N)\, \mathbf{B}_{N_{max}}\, \mathbf{y}, \tag{2}$$

where $\mathbf{y}$ is its input that is $n \times 1$ vector of observations (1), while $\hat{\mathbf{y}}$ is its $n \times 1$ output. In (2) $\mathbf{I}_{N_{max}}(N)$ denotes $N_{max} \times N_{max}$ diagonal matrix where $(N+1)$ first elements (counting from the upper left corner) are equal to 1, and $N_{max} - (N+1)$ are equal to zero. Its role is just to filter out proper submatrices from $\mathbf{B}_{N_{max}}$ and its transposition. In actual computations, only these submatrices are involved.

However, matrix $\mathbf{I}_{N_{max}}(N)$ plays an important role in pointing out that the process of learning N based on $\mathcal{L}_L$ is, in fact, nonlinear, but after completing it and substituting the selected $\tilde{N}$, say, into (2), the computations become linear in $\mathbf{y}$, which is important for applying the proposed encoder in real-time. Therefore, we call this autoencoder seemingly linear.

Roughly speaking, $\tilde{N}$ is selected as

$$\tilde{N} \;=\; \arg\min_{N}\, [-\ln(\mathbb{L}(\mathcal{L}_L,\, N) + \mathit{penalty}(N)]\,, \tag{3}$$

where $0 \leq N \leq N_{max}$, $\mathcal{L}_L$, $N)$ is the likelihood function, while $\mathit{penalty}(N)$ is derived from the Akaike's Information Criterion (AIC) for all $\mathcal{L}_L$.

Observe that in (2) matrix $\mathbf{B}_{N_{max}}$ and its transpositon are defined, instead of coming from a learning process, as it is typical for most of the encoders discussed in the literature. Clearly, the number of rows of submatrix $\mathbf{B}_{\tilde{N}}$ depends on learning through $\tilde{N}$ but the formulas for computing their elements are precisely defined.

Properties of the proposed autoencoder To motivate the proposed form of the encoder, it is convenient to split the autoencoder so as to express its latent variables explicitly, namely,

$$\tilde{\mathbf{d}}^{(N)}(\mathbf{y}) \;=\; (N+1)\, \Delta_n\, \mathbf{B}_N\, \mathbf{y}, \quad \hat{\mathbf{y}} \;=\; \mathbf{B}_N^{tr}\, \tilde{\mathbf{d}}^{(N)}(\mathbf{y}) \quad l = 1, 2, \ldots, L, \tag{4}$$

where $\tilde{\mathbf{d}}^{(N)}(\mathbf{y})$ is $N \times 1$ vector of the latent variables (subsequently called the vector of empirical descriptors). Explicitly, the elements of $\tilde{\mathbf{d}}^{(N)}(\mathbf{y})$ have the form:

$$\tilde{d}_k^{(N)}(\mathbf{y}) \;=\; (N+1)\, \Delta_n \sum_{i=1}^{n} y_i\, B_k^{(N)}(t_i), \quad k = 0, 1, \ldots, N. \tag{5}$$

From now to the end of this subsection, our explanations concentrate solely on the functional data for which the observations have the form (1), left formula. Taking the expectation of (5) we obtain

$$\mathbb{E}[\tilde{d}_k^{(N)}(\mathbf{y})] \;=\; (N+1)\, \Delta_n \sum_{i=1}^{n} f(t_i)\, B_k^{(N)}(t_i) \approx (N+1) \int_T f(t)\, B_k^{(N)}(t)\, dt, \tag{6}$$

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36021-3_26

$k = 0, 1, \ldots, N$ where $\approx$ denotes the approximation of the Riemann integral by its sum, which is sufficiently accurate for smooth f. The following expressions are further called (theoretical) descriptors of f.

$$d_k^{(N)}(f) \stackrel{def}{=} (N+1) \int_T f(t)\, B_k^{(N)}(t)\, dt \;=\; (N+1) \;< f, , B^{(N)} >, \qquad (7)$$

k=0, 1,. . . , N , where $< ., . >$ denoted the inner product in $L_2(T)$. They serve as reference points for the assessment of $\tilde{d}_k^{(N)}(\mathbf{y})$. Furthermore, $d_k^{(N)}(f)$'s have appealing shape preserving features. Namely,

recovery of constant: if $f = c$ is constant in T, then $d_k(f) = c, k = 0, 1, \ldots, , N$
level preservation: if $f(t) \geq c > 0$, $t \in T$, then $d_k(f) \geq c$, $k = 0, 1, \ldots, N$.
monotonicity preservation: if f has continuous derivative in T and $f'(t) > 0$ in T, then also $d_k(f)$, $k = 0, 1, \ldots, N$ is strictly increasing in T,
convexity preservation: if f is twice continuously differentiable in T and $f''(t) > 0$ in T, then sequence $d_k(f)$, $k = 0, 1, \ldots, N$ is strictly convex, i.e., its second order differences are positive.

Proofs of the above properties follow from the well-known formulas for the derivatives of the Bernstein polynomials (see [13]), using integration by parts of their products and the derivatives of f.

It can also be proven that for each $0 \leq k \leq N$ and for every finite and fixed N we have: $\mathbb{V}ar(\tilde{d}_k^{(N)}(\mathbf{y})) \leq \sigma^2 \frac{(N+1)^2}{n}$, while for f continuously differentiable the bias of $\tilde{d}_k^{(N)}(\mathbf{y})$ is of the order $O(N/n)$. Thus, also $\mathbb{E}[\tilde{d}_k^{(N)}(\mathbf{y}) - d_k(f)]^2$ converges to zero as $n \to \infty$.

3 Selecting N using the autoencoder

Our aim in this section is to propose the method of selecting the number of descriptors N when n is finite. Contrary to classic problems dedicated to selecting N for samples of a particular f, in our case, we need to select N that is suitable for a family of functions f.

Learning N by autoencoder The idea of selecting N is to design an autoencoder such that
a) it obtains $\mathbf{y}^{(l)}$, $l = 1, 2, \ldots, L$ as inputs
b) and subsequently, for each $l = 1, 2, \ldots, L$, converts them into latent variables $\tilde{d}_k^{(N)}(\mathbf{y}^{(l)})$, $k = 0, 1, \ldots, N$, which are the descriptors estimates, computed according to (5),
c) finally, for $l = 1, 2, \ldots, L$ the encoder outputs are computed as follows

$$\tilde{f}_i^{(N)}(\mathbf{y}^{(l)}) \;=\; \sum_{k=0}^{N} \tilde{d}_k^{(N)}(\mathbf{y}^{(l)})\, B_k^{(N)}(t_i) \quad i = 1, 2, \ldots, n, \qquad (8)$$

where $\tilde{f}_i^{(N)}(\mathbf{y}^{(l)})$'s are interpreted as estimates of $f_l(t_i)$'s.

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36021-3_26

Then, for the whole learning sequence we obtain

$$\tilde{\mathbf{d}}^{(N)}(\mathbf{y}^{(l)}) \;=\; (N+1)\,\Delta_n\,\mathbf{B}_N\,\mathbf{y}^{(l)}, \quad \tilde{\mathbf{f}}^{(N)}(\mathbf{y}^{(l)}) \;=\; \mathbf{B}_N^{tr}\,\tilde{\mathbf{d}}^{(N)}(\mathbf{y}^{(l)}), \tag{9}$$

$l = 1, 2, \ldots, L,$, where, according to (8), $n \times 1$ vector $\tilde{\mathbf{f}}^{(N)}(\mathbf{y}^{(l)})$ consists of $\tilde{f}_i^{(N)}(\mathbf{y}^{(l)})$, $i = 1, 2, \ldots, n$.

We state a version of the well-known (see [3] for review) Akaike's Information Criterion (AIC) that is suitable for our purposes. The necessity for the AIC generalization comes from selecting N, which is suitable for the whole family of curves.

Assume that the sampling schemes are of the same form as in (1), with i.i.d. noises having $\mathcal{N}(0, \sigma^2)$ distribution. This assumption is made in this section only, since the obtained formulas are interpretable and useful for a large class of probability distributions having zero mean and finite variance.

Corollary 1 (Approximate AIC criterion) *The approximate AIC (AAIC) for selecting N has the following form*

$$AAIC(N) \;=\; L\,\left[2\,\theta\,N + n\,\ln\left(\sum_{l=1}^{L}||\mathbf{y}^{(l)} - \mathbf{B}_N^{tr}\,\tilde{\mathbf{d}}^{(N)}(\mathbf{y}^{(l)})||^2\right)\right] + C(L,n), \tag{10}$$

where $C(L,n)$ is a constant that may depend on fixed n and L, but not on N, while $0 < \theta \leq 1$ is a correcting factor. The minimizer $\tilde{N}$ of $AAIC(N)$ is considered as an approximately optimal number of descriptors for all family f_l's.

We omit a long proof of this result. The AAIC properly generalizes the AIC to a family of functions in the sense that it is not a simple average of the AIC's for each f_l's and the corresponding sub-models.

Algorithm 1 (for learning the number of descriptors by the AAIC)

Input *L, n, $N_{max} < n$, N_{min} (or set $N_{min} = 3$), $\mathbf{y}^{(l)}$, $l = 1, 2, \ldots, L$.*

While $N_{max} \leq n$ **do**

Step 1 *Compute $\tilde{N}$ that minimizes the expression in the brackets in* (10) *over $N_{min} \leq N \leq N_{max}$.*

Step 2 *If $\tilde{N} = N_{max}$, set $N_{min} = N_{max} - 1$, enlarge N_{max} and go to Step 1, otherwize, STOP and output $\tilde{N}$.*

A correcting factor $\theta > 0$ in (10) is in most cases set to 1. However, when observation errors are large, it happens that the minus log-likelihood function decreases rather slowly with N. In such cases, it is reasonable to apply $\theta > 0$ strictly less than 1. Algorithm 1 was tested on synthetic data, providing satisfactory results. We do not display them by the lack of space.

Testing autoenkoder on real data As a benchmark for testing the proposed descriptors, we selected samples of acceleration signals that are publicly available [24].

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36021-3_26

Such signals arise when acceleration measurements are made in a bucket wheel excavator operator's cabin or other large machines. The bucket wheel excavator works in a quasi-periodic or a repetitive way since the buckets hit the ground at equidistant time instants. Their positions point out intervals that are considered here as the common domain T of signals to be classified. In our example, each signal is sampled at $n = 1024$ points of T (see Fig. 1) (right panel), where T has the duration of one second.

These samples form vectors $\mathbf{y}^{(l)}$'s. We have only 43 of them, but it is sufficient to run Algorithm 1. The resulting AAIC plot is shown in Fig. 1 (left panel) for $\theta = 0.5$. The minimum is clearly visible at $\tilde{N} = 40$ that is further taken as the number of descriptors used for reconstruction of the acceleration signals. For illustration purposes only, in Fig. 1 (right panel) the signal reconstructed by the autoencoder for $\tilde{N} = 40$ is also shown.

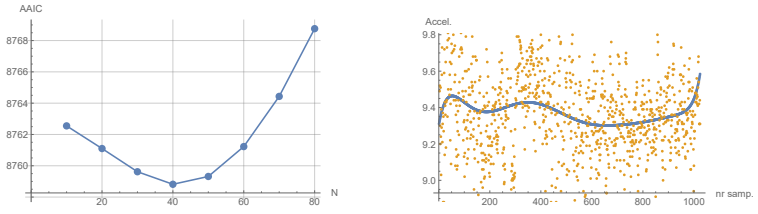

Fig. 1. Left panel – The result of applying Algorithm 1 to the accelerations samples – the AAIC plot for selecting N. Right panel – The result of applying Algorithm 1 to the accelerations samples – reconstruction of one sample by the autoencoder

Conclusions The proposed autoencoder is based on the Bernstein-Durrmeyer polynomials. It was pointed out that the autoencoder descriptors inherit shape-preserving properties of the Bernstein-Durrmeyer polynomials, such as monotonicity and convexity, provided that no noise is present. These properties can be observed also when observations are corrupted by intensive noise.

References

1. Baldi, P. and Hornik, K. Neural networks and principal component analysis: Learning from examples without local minima. Neural Networks, **2**(1), 53-58, (1989).
2. Bank, D., et al Autoencoders. arXiv:2003.05991, (2020).
3. Cavanaugh, J. E., Neath, A. A. The Akaike information criterion: Background, derivation, properties, application, interpretation, and refinements. Wiley Interdisciplinary Reviews: Computational Statistics, **11**(3), e1460, (2019).
4. Chen, W. and Ditzian, Z. Best polynomial and Durrmeyer approximation in $L_p(S)$ Indagationes Mathematicae, **2**, 437 – 452 (1991)
5. Chen, G. Y., Bui, T. D., Krzyżak, A. Rotation invariant feature extraction using Ridgelet and Fourier transforms. Pattern Analysis and Applications, **9**, 83-93, (2006).

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36021-3_26

6. Chen, G., Krzyżak, A., Qian, S. E. A new endmember extraction method based on least squares. Canadian Journal of Remote Sensing, **48**(2), 316-326, (2022).
7. Derrienic, M. M. On multivariate approximation by Bernstein-type polynomials. Journal of the Approximation Theory., **45**, 155 – 166, (1985).
8. Doersch, C. (2016). Tutorial on variational autoencoders. arXiv preprint arXiv:1606.05908.
9. Gałkowski, T., Krzyżak, A., Filutowicz, Z. A new approach to detection of changes in multidimensional patterns. Journal of Artificial Intelligence and Soft Computing Research, **10**(2), 125-136, (2020).
10. Harris, T., Tucker, J.D., Li, B., Shand, L.: Elastic depths for detecting shape anomalies in functional data. Technometrics Elastic depths for detecting shape anomalies in functional data. **63**, 1–11 (2020)
11. Kingma, D. P., Welling, M. An introduction to variational autoencoders. Foundations and Trends® in Machine Learning, **12**(4), 307-392, (2019).
12. Krzyżak, A., Leung, S. Y., Suen, C. Y. Reconstruction of two-dimensional patterns from Fourier descriptors. Machine Vision and Applications, **2**, 123-140, (1989).
13. Lorentz, G. G. *Bernstein Polynomials*. American Mathematical Soc., (2013)
14. Luo, W., Li, J., Yang, J., Xu, W. and Zhang, J.. "Convolutional sparse autoencoders for image classification.IEEE Transactions on Neural Networks and Learning Systems **29**7, 3289-3294, 2017.
15. Marron, J.S., Ramsay, J.O., Sangalli, L.M., Srivastava, A.: Functional data analysis of amplitude and phase variation. Statistical Science **30**(4), 468–484 (2015). https://doi.org/10.1214/15-STS524
16. Plaut, E.. From principal subspaces to principal components with linear autoencoders. ArXiv preprint arXiv:1804.10253 (2018).
17. Rafajłowicz, E., and Skubalska-Rafajłowicz, E. , Nonparametric Regression Estimation by Bernstein–Durrmeyer Polynomials, Tatra Mt. Math. Publ., **17**, 227–239 (1999)
18. Rafajłowicz, W., Rafajłowicz, E. and Więckowski J., Learning Functional Descriptors Based on the Bernstein Polynomials – Preliminary Studies, In: Artificial Intelligence and Soft Computing: 21st ICAISC 2022, Proc., Part I pp. 310-321, Springer. (2023).
19. Rifai, S., Vincent, P., Muller, X., Glorot, X. and Bengio, Y. Contractive autoencoders: Explicit invariance during feature extraction. In Proceedings of the 28th International Conference on Machine Learning, 833-840, 2011.
20. Rumelhart, D.E., Hinton, G.E. and Williams, R.J., Parallel distributed processing: Explorations in the microstructure of cognition, **1**, 26, 1986.
21. Sitzmann, V., Martel, J., Bergman, A., Lindell, D., Wetzstein, G. Implicit neural representations with periodic activation functions. Advances in Neural Information Processing Systems, **33**, 7462-7473, (2020).
22. Tadeusiewicz, R.: Automatic understanding of signals. In: Intelligent Information Processing and Web Mining. pp. 577–590, Springer (2004).
23. Vincent, P. A connection between score matching and denoising autoencoders. Neural Computation, **23**(7), 1661-1674, (2011).
24. Więckowski, J., Rafajłowicz, W., Moczko, P., Rafajłowicz, E.: Data from vibration measurement in a bucket wheel excavator operator's cabin with the aim of vibrations damping. Data in Brief p. 106836 (2021), `http://dx.doi.org/10.17632/htddgv2p3b.1`

Acknowledgements. The authors express their thanks to the anonymous referees for comments clarifying the presentation.

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36021-3_26