

Neural Approximation of Empirical Functions

J. ROJ*

Institute of Measurement Science, Electronics and Control at the Silesian University of Technology
Akademicka 10, 44-100 Gliwice, Poland

The paper presents the results of simulation studies of selected neural network structures used for non-linear function approximation based on a limited accuracy data. There was performed the analysis of the interdependence of the network structure and the size of the set of learning patterns. The approximation inaccuracy was expressed by the uncertainty interval width. The approximation properties of the neural method were compared with those of the piece-wise linear and polynomial: “cubic” and “spline” methods.

DOI: [10.12693/APhysPolA.124.554](https://doi.org/10.12693/APhysPolA.124.554)

PACS: 84.35.+i, 07.05.Mh, 06.20.-f, 06.20.F-, 06.20.fb, 02.60.Ed

1. Introduction

The characteristic feature of artificial neural networks is the ability of learning and generalizing the acquired knowledge for new, previously unknown data [1, 2]. This means that one of the main areas of their application are various approximation tasks. For this purpose there are generally used feed-forward multilayer perceptron (MLP) [3] and radial basis function (RBF) [1] neural networks. In many publications, e.g. [4–6], there is shown highly effective approximation of any function of many variables with these types of neural networks, called by some authors “universal approximators” [7, 8].

Thanks to its advantageous approximation properties, neural networks are also widely used in the measurement area [9–12]. They are often the last element of a processing chain which reconstructs the measurement values [13, 14] by solving the inverse equation of the analog processing phase. This requires knowledge of the inverse model describing the analog processing chain, which involves the need for its identification. For this purpose, there is required a set of the input quantity measurement results which are always obtained with a limited accuracy. In general, developing an enough accurate inverse model of the analog processing chain is often difficult, time-consuming and not always possible to carry out. The main advantage of a neural network in the described situation is that the neural learning process “creates” a right inverse model by itself, that is it approximates the inverse characteristics of the analog processing chain.

In a wider context, these issues can be considered as approximation or interpolation of empirical functions carried out on the basis of the measurement data of a limited accuracy. In this case, comparison of the inaccuracy of the neural representation of a nonlinear function with that of the classical (polynomial) approximation or interpolation methods can be of essential practical importance.

2. Interdependence of the neural network structure and the size of sets of learning patterns

The use of neural networks for solving a specific task requires, first of all, determination of the interdependence between the network structure and the size of the set of learning patterns. It is necessary to answer the question what is the minimum size of the set of learning patterns that allows sufficiently training the specified network structure to have the required generalizing properties. The characteristic feature of neural networks is “overlearning” phenomenon, occurring for too many patterns of learning and resulting in generation of excessive errors by the network. In publications there can be found the dependence of the size of the set of learning patterns on the Vapnik–Chervonenkis dimension (VCdim), describing the network complexity [15]. However, there is no simple relationship between a multilayer network structure and the VCdim size. There can be only provided an estimate of the upper and lower range of the VCdim [16] on the basis of the dependence

$$2 \operatorname{int} \left(\frac{H}{2} \right) N \leq \operatorname{VCdim} \leq 2N_w (1 + \log N_n), \quad (1)$$

where $\operatorname{int}(\cdot)$ is an integer function, N is the number of network inputs, H is the number of neurons in the hidden layer, N_w is the total number of connections in the network (the number corresponds to the weighting factors), N_n is the total number of neurons in the network.

It can be noted that the lower limit of the VCdim is approximately equal to the number of weights between the input and hidden layer, and the upper limit is more than twice of all the network weights. In practice, for the networks with sigmoidal continuous transfer functions it is assumed

$$\operatorname{VCdim} \cong 2N_w. \quad (2)$$

Estimating the VCdim value allows specifying the minimum size of training data set, but generally it is assumed that good generalization ability of the network occurs in the case when the size of the training set is 10 times greater than the VCdim value. This means that the greater the number of connections in the network (its

*e-mail: jerzy.roj@polsl.pl

complexity) is, the greater the size of the training data set must be.

Another key issue is the proper selection of the network structure for the type of the problem being solved. In a general case, there can be used the Kolmogorov theorem [17, 18] which proves the existence of solution to the problem of approximation of a function of several variables by superpositions of many functions of one variable. Transferring this theory to the neural networks field means that in the case of a continuous function which transforms the N -dimensional input set to the M -dimensional output set, approximation of this function is possible with use of a feed-forward network with one hidden layer, containing $2N + 1$ neurons.

The Kolmogorov theorem defines only the structure of the network dedicated to solving approximation problems, but does not specify the type of the transfer function to be used, or the method of learning. In practice, a simple application of the Kolmogorov theorem often does not allow achieving the optimum solutions which generally require the application of a larger number of neurons in the hidden layer. Thus, the network structure resulting from the quoted theorem may be a starting point for determining the optimal number of hidden layer neurons by gradually adding them to obtain the assumed accuracy of the approximation task.

3. Results of simulation researches

Simulation researches were carried out for the MLP and RBF neural networks, using for this purpose the Neural Network Toolbox library, available in Matlab. It was assumed that the task of the neural network was approximation of a strongly nonlinear function

$$y = 0.01\text{humps}(x), \tag{3}$$

where $\text{humps}(x)$ is a function of the internal Matlab environment defined as:

$$\begin{aligned} \text{humps}(x) = & \frac{1}{(x - 0.3)^2 + 0.01} \\ & + \frac{1}{(x - 0.9)^2 + 0.04} - 6. \end{aligned} \tag{4}$$

The graph of the function described by relation (3) is shown in Fig. 1.

This impulse function cannot be treated as a model of the measuring transducer because it is not monotonic, however, due to the relatively strong non-linearity, it is a good object of research. It can be said that the impulse function (3) is an extreme point of reference for the real characteristics of transducers (it is more nonlinear than they are), so the research results obtained for it are general.

The network structure implementing the approximation of one variable function, resulting from the Kolmogorov theorem, contains one hidden layer with three neurons (1-3-1 structure, the number of neurons in the input, hidden, and output layer, respectively). For a neural network with one input and output, it is necessary to generate a learning set composed of input

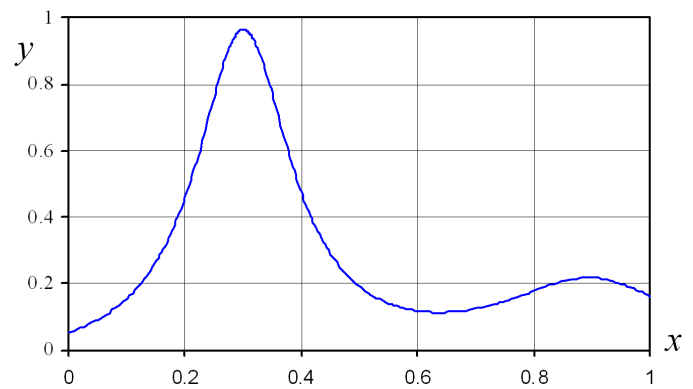


Fig. 1. Graph of the function given by Eq. (3).

$\mathbf{x} = [x_1, x_2, \dots, x_{N_u}]^T$ and output $\mathbf{y} = [y_1, y_2, \dots, y_{N_u}]^T$ vectors, where N_u is the number of learning patterns. The elements of this set were determined on the basis of the functional relationship (3) which the network has to learn. Estimation of the size of training set for this network, based on the VCdim (2), gives the number $N_{uVC} \cong 10(2N_w) = 10 \times 24 = 120$ (excluding neurons biases).

The values of the learning set elements were adequately rounded to model the effect of quantization process in a 12-bit A/D converter. After the learning process (using the Levenberg–Marquardt algorithm), the selected networks were tested by the not quantized, 10 000 element testing set.

In the general case, to evaluate the training level of the neural network, which includes both learning and testing processes, there are used various “global” measures of errors. The ones used most frequently for this purpose are: mean absolute error [19], sum squared error [20], mean square error [21] or root mean square error [22]. These parameters are useful only to compare the quality of a specific task realization by neural networks (of different structures, learned by different algorithms). Full metrological evaluation of a measuring chain in which neural processing is used (neural network acts as a measuring transducer) cannot be formulated on this basis. For this purpose, it is necessary to determine the uncertainty, the value of which is not directly related to these measures of errors. The method for determining the uncertainty width interval of the neural network output data on the basis of the error histogram obtained in a testing process is generally described in [23], in detail in [24].

The research results of the networks of selected structures trained by N_u -element learning sets, the values of which were quantized with 12 bit resolution are shown in Fig. 2. The parameter U_W^{α} is the uncertainty width interval (at the confidence level $\alpha = 0.95$) of the output results of the networks performing approximation of function (3).

In the next stage of the study there was compared the inaccuracy of neural representation of the nonlinear em-

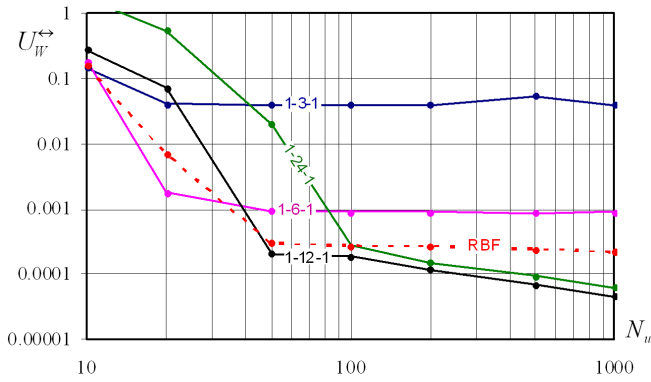


Fig. 2. The results of testing the MLP and RBF networks with different number of hidden neurons, approximating impulse function (3), trained by N_u -element pattern sets, whose values were quantized with 12-bit resolution. The number of RBF radial neurons is equal to the number of the training set elements.

pirical function with that of the conventional interpolation or approximation methods. These methods are used in practice in order to obtain a numerical algorithm executed by a processor. From the measurements taken there is obtained the data set, on basis of which there is built an inverse, analytical model of the analog processing chain. Comparison of the reconstruction results obtained from the neural networks and numerical methods enables taking decision on selection of one of these reconstruction methods.

The Matlab software allows using the appropriate functions to perform interpolation by the piecewise-linear (“linear”), third-order spline (“spline”) and piecewise cubic method (“cubic”) using the Hermite polynomials [25]. These methods were used to interpolate the impulse function (3), and the node values were quantized with a resolution of 8, 12, and 16 bit.

The quantized values of the interpolation nodes were used at the same time as elements of learning sets for the MLP neural network with 12 hidden neurons (MLP12) and RBF network with the number of radial neurons equal to the number of learning patterns (due to the specificity of learning the RBF network implemented in Matlab).

Figure 3 presents the results of the comparative study as the width interval of the uncertainty U_W^{\leftrightarrow} expressing the inaccuracy of neural representation of function (3). The attempts to use polynomial approximation in the described situation gave much worse results, despite a relatively high-order of the approximating polynomial and the large number of points used to identify the polynomial coefficients.

4. Concluding remarks

The following remarks can be formulated based on the performed studies. The results presented in Fig. 2 show

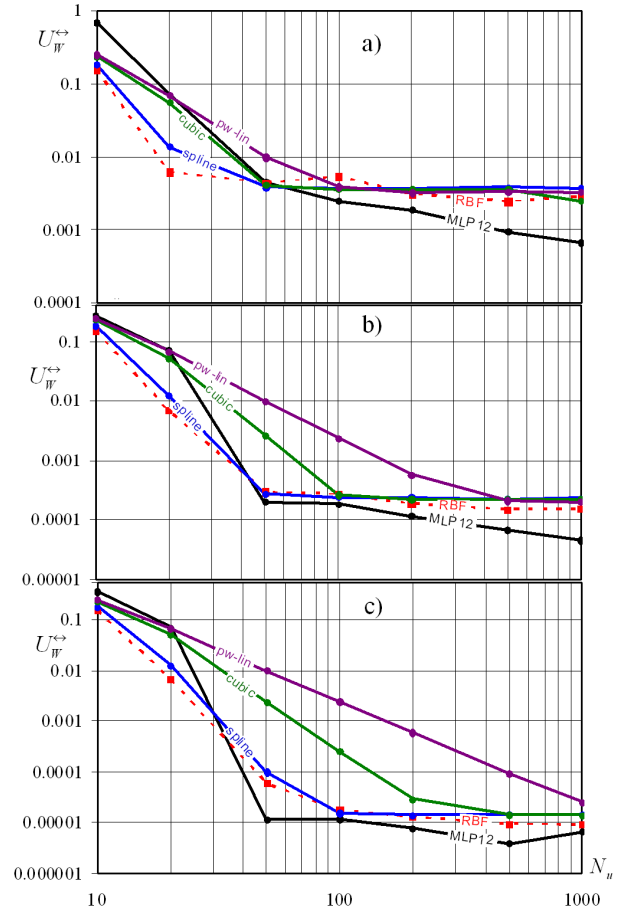


Fig. 3. The uncertainty interval width U_W^{\leftrightarrow} of impulse function (3) representation by interpolating methods and neural networks depending on the size of N_u training pattern sets and interpolation nodes whose values were quantized with the resolution of (a) 8 bits, (b) 12 bits, (c) 16 bits.

that when using the data with limited accuracy for network training, selection of the network structure due to the Kolmogorov theorem (3 hidden neurons) and the size of the training set estimated on the basis of the VCdim ($N_u = 120$) do not ensure obtainment of the best approximation results. In situations in which the learning data are quantized e.g. with a resolution of 12 bit, the MLP network with 12 hidden neurons performs the approximate task most accurately for the number of learning patterns not fewer than 50.

Furthermore, analyzing the results presented in Fig. 3, it can be seen that the abovementioned network also performs representation of function (3) more accurately than the conventional nonlinear interpolation methods, among which the spline interpolation showed the best convergence. Moreover, it can be noted that these methods, from a certain number of interpolation nodes N_u , allowed achieving approximately the same, impassable limit U_W^{\leftrightarrow} , expressing the interpolation inaccuracy.

References

- [1] S. Haykin, *Neural Networks: A Comprehensive Foundation*, 3rd ed., Prentice Hall, London 2008.
- [2] M. Gupta, N. Homma, L. Jin, *Static and Dynamic Neural Networks: From Fundamentals to Advanced Theory*, Wiley, New Jersey 2003.
- [3] X. He, S. Xu, *Process Neural Networks*, Springer-Verlag, Berlin 2009.
- [4] A.R. Barron, *IEEE Trans. Inf. Theory* **36**, 930 (1993).
- [5] K. Hornik, *Neural Networks* **2**, 251 (1991).
- [6] T. Chen, H. Chen, *IEEE Trans. Neural Networks* **4**, 904 (1995).
- [7] K. Hornik, M. Stinchcombe, H. White, *Neural Networks* **2**, 359 (1989).
- [8] J. Park, I.W. Sandberg, *Neural Comput.* **2**, 246 (1991).
- [9] S. Ablameyko, L. Goras, M. Goris, V. Piuri, *Neural Networks for Instrumentation, Measurement and Related Industrial Applications*, IOS Press, Oxford 2003.
- [10] W. Jakubik, M. Urbanczyk, E. Maciak, T. Pustelny, *Bull. Pol. Acad. Sci., Techn. Sci.* **56**, 133 (2008).
- [11] J. Ignac-Nowicka, T. Pustelny, Z. Opilski, E. Maciak, W. Jakubik, M. Urbanczyk, *Opt. Eng.* **42**, 2978 (2003).
- [12] T. Pustelny, J. Ignac-Nowicka, B. Jarzabek, A. Burian, *Opt. Appl.* **34**, 551 (2004).
- [13] A.P. Singh, T.S. Kamal, S. Kumar, *ISA Trans.* **3**, 319 (2006).
- [14] P. Kluk, R.Z. Morawski, in: *Proc. IEEE Instrum. Meas. Technol. Conf. — IMTC'96, Brussels (Belgium)*, 1996, p. 581.
- [15] V. Vapnik, *The Nature of Statistical Learning Theory. Information Science and Statistics*, Springer-Verlag, New York 2000.
- [16] D. Hush, B. Horne, *IEEE Sign. Proc. Mag.* **8**, (1993).
- [17] S.J. Khavinson, *Math. Notes* **4**, 406 (1994).
- [18] V. Kurkova, *Neural Networks* **5**, 501 (1992).
- [19] B.A. Smith, R.W. McClendon, G. Hoogenboom, *Int. J. Comput. Intellig.* **3**, 179 (2006).
- [20] A. Rabiatal, A. Zainal, *IIUM Eng. J., Spec. Iss. Sci. Ethics* **6**, 45 (2011).
- [21] S. Olyae, S. Hamedi, *J. Phys., Conf. Series* **276**, 1 (2011).
- [22] G.-B. Huang, Q.-Y. Zhu, Ch.-K. Siew, *IEEE Trans. Neural Networks* **4**, 863 (2006).
- [23] J. Jakubiec, P. Makowski, J. Roj, *IEEE Trans. Instrum. Measur.* **3**, 649 (2009).
- [24] J. Roj, *Przegląd Elektrotechniczny* **1a**, 84 (2013) (in Polish).
- [25] C. de Boor, *A Practical Guide to Splines*, Springer-Verlag, New York 2001.