Acoustic and Biomedical Engineering

Hardware Implementation of Artificial Neural Networks for Vibroacoustic Signals Classification

D. Dąbrowski*, E. Jamro and W. Cioch
Department of Mechanics and Vibroacoustics, Department Electronics Mining and Metallurgy Academy
al. Mickiewicza 30, 30-059 Kraków, Poland

This paper studies the architecture of a neural classifier designed to identify technical condition of machines, based on vibroacoustic signals. The designed neural network is optimized for implementation on Field Programmable Gate Arrays (FPGA) programmable devices. FPGA allows massive parallelism and thus real-time classification as each neuron can execute arithmetic operations simultaneously. The classifier of vibroacoustic signals was designed and tested for the self — organized neural network. The teaching vectors are based on estimates derived from processed vibroacoustic signals generated by rotary machines. The created classifier was applied for recognizing technical state of demonstrative toothed gear DMA1 in variable operating conditions.

PACS numbers: 45.80.+r, 46.40.–f

1. Introduction

1.1. Artificial neural networks for classification of machines technical condition

During the operation of different kinds of machines and devices it is crucial to efficiently and quickly evaluate their technical state. Changes in rotational speed or loading introduce ambiguities as early as in inference stage. Creating technical state vectors is necessary for identification of damage sources. These vectors consist of many numerical estimators, composing multidimensional vectors of state. It is possible to describe the state of an object at a given moment by the set of the object features (symptoms). Symptom is a general measure defined [1] as follows:

$$S(r,\theta) = E_t\{\Phi(s)\} = \frac{1}{T}\int_0^T \Phi\left[s(r,t,\theta)\right] \mathrm{d}t$$
$$= \Phi_0\left[\sum a_i(r)V_i(\theta) + c(r,\theta)\right] + n(r,\theta) \qquad (1)$$
$$V_i(\theta) = E_t\{V_i(t,\theta)\}, \quad i = 1,\ldots,n$$

where:
$E_t(*) = \frac{1}{T}\int_0^T (*)\,\mathrm{d}t$ — averaging operator after dynamic time t during observation time T,
$\Phi(*)$ — operation conducted on signal in order to get information,
$\Phi_0(*)$ — the functional relations of symptoms on indicated arguments with weight function $a(r)$,

Determining an object state involves subordinating an observation vector to a class distinguished in a process of grouping [1].

Separating such vectors with regard to class of technical state should be unambiguous. Moreover, the chosen characteristics shouldn't be sensitive to changes of working conditions. Artificial neural networks which, with their architecture and way they work, reflect the neural network in human brains are important tool of classification. There are ambiguities in mathematical description of technical state of machines. The application of a classifier based on neural networks enables the recognition of machines technical state when mathematical description is ambiguous [2–4]. Broad research on neural classifiers carried out recently indicates that artificial neural networks are promising alternative to conventional methods of classification [2, 3, 5, 6].

1.2. Symptoms selection for technical state classification of toothed gear

The purpose of the research is to build a neural classifier able to recognize a technical state of the toothed gear. The diagnostics of toothed gear, owing to their wide use in transmissions, is the object of interest of many industrial and scientific centers. The constructional variety of toothed-gear types inflicts that despite the existing inference procedures and algorithms of analysis of vibroacoustic signals, accuracy of diagnosis is in many cases insufficient [6, 7]. Artificial intelligence methods, which allow to model different kinds of nonlinearities contained in diagnostic signals, make up competitive tool compared to classic inference algorithms [2–4].

TABLE I
Vector of technical state of demonstrative DMG-1 toothed gear.

Vector of technical state	
I Estimator	Amplitude of the rotational frequency
II Estimator	Amplitude of the second harmonic of rotational frequency
III Estimator	Ratio of the second-harmonic amplitude to the reference frequency amplitude
IV Estimator	Amplitude of the third harmonic of rotational frequency
V Estimator	Ratio of the third-harmonic amplitude to the referenced rotational frequency amplitude

* corresponding author; e-mail: dabrowsk@agh.edu.pl

The researches were conducted on demonstrative toothed gear, DMG-1, enabling the introduction of certain changes in misalignment state and loading, as well as controlling rotational speed. The signals in the experiment were applied for two classes of technical state illustrating misalignment and its lack. Researches were carried out for two rotational speeds: 900 rpm and 1200 rpm and with and without load. The vector of technical state is based on spectrum of vibration velocity signals and is presented in Table I. The following symptoms we employed: rotation frequency amplitude and its harmonic amplitude, and their ratio.

The teaching set consisted of 12 vectors for each of the two classes comprising estimates for gear operation at two rotational speeds and for two different loads. The test set is composed of the same vectors — 8 for each class.

$$f = \frac{1}{\sqrt{n}} \tag{2}$$

$$s = f\sqrt{n - l^2} \tag{3}$$

$$X_i = f x_i \tag{4}$$

x_i — input vector containing n — elements before normalization, $i = 1, 2, 3 \ldots n$

X_i — input vector containing n — elements after normalization, $i = 1, 2, 3 \ldots n$

Input data are normalized before feeding the neural network. The Z-axis normalization method was selected, due to retaining the information about the absolute value [8].

2. The neural classifier of vibroacoustic signals

2.1. The architecture of LVQ neural networks

The LVQ (Learning Vector Quantization) was applied in conducted research. This model of neural network was introduced by T. Kohonen [9]. The LVQ network is made of three layers. First layer is the input layer. The second layer is the Kohonen-type self-organizing layer — it classifies the input vectors for categories detected during the grouping process of the teaching set. During learning process as a result of competition, Kohonen layer is subject to self-organizing according to the Kohonen's principle (5). This process involves correction of weights of vector which is the closest to teaching vector:

$$\hat{w}_m^t \hat{x} = \max(\hat{w}_i^t \hat{x}), \quad i = 1, \ldots, n \tag{5}$$

$$\hat{w}_m^{k+1} = \hat{w}_m^k + \alpha^k(\hat{x} - w_m^k) \tag{6}$$

$$\hat{w}_i^{k+1} = \hat{w}_i^k, \quad i \neq m \tag{7}$$

where:

$\hat{w}$ — Standardized vector of weight,

$\hat{x}$ — standardized input vector,

k — time moment,

m — digit of the winning neuron

In the model of LVQ network the neurons of Kohonen layer are assigned to individual classes, so that the equal number of neurons be assigned to a specific class, Fig. 1.

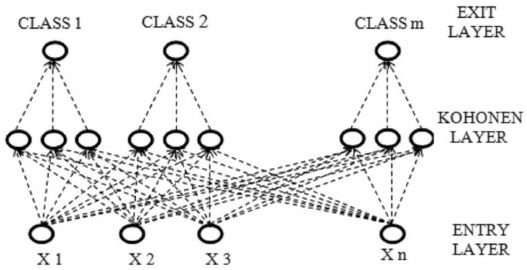

Fig. 1. Architecture of LVQ artificial neural network.

The purpose of the output layer is to assignment the input vector to one of several classes. The vector quantization network is an example of a strain network, which means that the individual neurons in a self-organized layer are connected with individual classes by a designer [10, 11].

2.2. Selection of neural classifier architecture

Apart from the above described model of the Kohonen neural network, there are also modified models. In case of too great number of winnings for one neuron during learning, that neuron is pushed aside from rivalry for some time. Such a model of network is called LVQ1. The model of neural network designed by authors is a modified LVQ1 network. The algorithm of pushing aside a winning neuron from rivalry was modified. Alteration involves counting the winnings for each neuron, in case when the number of winnings of a given neuron will reach certain values it is excluded from rivalry. This procedure is repeated for next realizations of learning network. When the number of winnings for every neuron will reach certain value it is reset. Such modification of model of vector quantization network enables to activate uniformly all the neurons in Kohonen's layer, thus the neural classifier has a much greater ability of learning.

Another modification is how the distance between the input vector and neural weights is calculated. The authors proposed to calculate the distance according to the (8b) i.e. sum of absolute difference between the neural weights and the input vector. A standard methods (8a) employs Euclidean distance:

$$\|w - x\| = \sqrt{\sum_{i=1}^{N}(w_i - x_i)^2} \tag{8a}$$

$$\|w - x\| = \sqrt{\sum_{i=1}^{N}|w_i - x_i|} \tag{8b}$$

where:

w — weight vector,

x — input vector,

The introduced method is particularly useful in case of hardware implementation of neural network, because it enables substitution of complex operations such as square and the square root calculation by a lot of simpler operation of absolute value calculation. This results in shorter

propagation time of network as well as reduced hardware area. The operation of fixed-point multiplication (similar to squaring) requires roughly n^2 Logic Elements (LE — a basic element in a FPGA), compared with operation of absolute value calculation, which occupies roughly n LEs; where n is the bit-width of data [9]. Further analyses will be carried out by PC with the algorithm written in C++ language, realizing designed artificial neural network.

3. The hardware implementation of neural classifier

3.1. Analysis of artificial neural network parameters

After selecting architecture of neural classifier, analysis of its parameters for hardware implementation on FPGA devices was conducted. At the first stage, the number of neurons in self-organizing layer (hidden layer) was studied according to the following procedure. The initial number of neurons equals to 2. The network was taught, and then the classification accuracy of network operation E, defined in (10), was evaluated. Then, in the next iteration, the procedures were repeated for the number of neurons increased by 2, and so on.

$$E_p = \frac{1}{n}\sum_{i=0}^{n-1}(t_{pj} - o_{pj})^2 \tag{9}$$

$$E = \frac{1}{m}\sum_{p=0}^{m-1} E_p \tag{10}$$

where:
E_p — epoch error,
E — network operation error,
n — number of input neurons,
t_{pj} — correct activation of neurons,
o_{pj} — actual activation of neurons,
m — number of introduction in epoch.

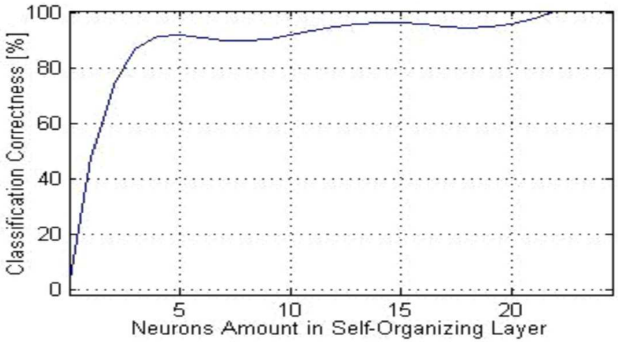

Fig. 2. Influence of number of neurons and iterations on classification correctness.

As demonstrated in Fig. 2, the sufficient classification efficiency can be achieved with as few as 4 neurons, however the network achieves a hundred-percent efficiency with the number of neurons equal to the number of teaching vectors. For the designed classifier 24 hidden neurons were selected.

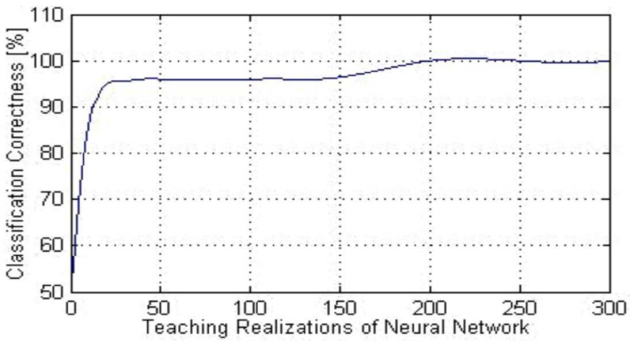

Fig. 3. Influence of number of neurons and iterations on classification correctness.

After selecting the number of hidden-layer neurons we searched for minimum number of teaching process iterations. Teaching process was carried out on a PC employing the floating-point data format. According to Fig. 3, after two hundred teaching iterations, the classifier shows a hundred-percent efficiency for teaching vectors. The network designed in this way was then fed with test vectors. Conducted research indicates almost a hundred-percent classification correctness of the test vectors different from the teaching set.

3.2. Implementation of neural classifier on FPGA

Implementation of the neural network on FPGA requires modification of data format from floating-point to fixed-point. In artificial neural networks data of low value may be represented with low accuracy, however data of significant value should be represented with greater accuracy; that is in according to fixed-point format.

Analyses on fixed-point network parameters are given in Fig. 4 and Table II. To obtain an accurate neural classifier, it is necessary to use 12-bits both for input data and neural weights. The designed classifier for vibroacoustic signals can be implemented on FPGA according to [8, 12, 13].

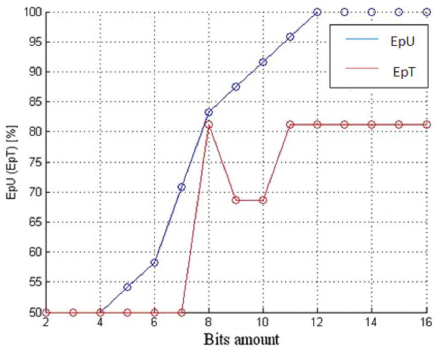

Fig. 4. Influence of weights and network input data bit-width on classification accuracy.

TABLE II

Influence of weights and input data bit-width on classification accuracy.

Bit	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
E_{pU}	50	50	50	54.1	58.3	70.8	83.3	87.5	91.6	95.8	100	100	100	100	100
E_{pT}	50	50	50	50	50	50	81.2	68.7	68.7	81.2	81.2	81.2	81.2	81.2	81.2

4. Conclusions

The present article studies selection of neural network architecture for classification of damages on DMA-1 toothed gear with the application of vibroacoustic signals. The new method of teaching selected neural network was proposed and analyzed. Based on conducted analyses, 24 hidden neurons and 200 teaching iteration of neural classifier were used. The main purpose of experiment was hardware implementation of neural network, with regard to operations conducted in parallel and the computation speed. For this reason, research on employing fixed-point instead of floating-point data format was conducted. In the proposed neural network 12-bits were used both for neural weights and input data.

References

[1] R. Tadeusiewicz, *Neural networks*, Akademicka Oficyna Wydawnicza RM, Warszawa 1993, (in Polish).

[2] J. Żurada, M. Barski, W. Jędruch, *Artificial neural networks*, Wydawnictwo Naukowe PWN, Warszawa, 1996, p. 240, (in Polish).

[3] T. Kohonen, *Neurocomputing* **21**, 1 (1998).

[4] E. Jamro, K. Wiatr, *Proc. of IEEE Design and Diagnostics of Electronics Circuits and Systems Workshop, Sopron, Apr. 2005*, p. 121.

[5] J. Adamczyk, W. Cioch, P. Krzyworzeka, *Int. Congress Diagnostyka 2000, Warszawa*, p.10.

[6] J. Dybała, S. Radkowski, *Diagnostyka* **31**, 59 (2004).

[7] P. Czech, B. Łazarz, *Diagnostyka* **2**, 42 (2007).

[8] W. Bartelmus, R. Zimroz, W. Sawicki, M. Maniak, Z. Woźniak, K. Furmaniak, *Górnictwo Geoinżynieria* **31**, 75 (2007).

[9] C. Cempel, *Vibroacoustics diagnostics*, Państwowe Wydawnictwo Naukowe, Warszawa 1989, p 23, (in Polish).

[10] G.P. Zhang, *IEEE Appl. Rev.* **30**, 451 (2000).

[11] K. Skahill, *VHDL, Projecting Programmable Logic Devices*, II Ed., WNT, Warszawa 2004, (in Polish).

[12] J.S. Shawe-Taylor, M. Anthony, W. Kern, *Neural Networks*, Vol. 5, 1992, p. 971.

[13] J.C. Gallagher, S.K. Boddhu, S. Vigraham, *Evolutionary Computation*, Vol.3, 2005, p. 2461.