Using the Elman neural network as an identity map in defect detection task

Volodymyr Khandetskyi[1[0000-0002-6386-4637]], Nadiia Karpenko[1[0000-0003-4700-6357]],

[1]Oles Honchar Dnipro National University, Gagarin Avenue, 72, Dnipro, 49010, Ukraine
v.khandetsky@gmail.com, karpenko_n@ffeks.dnulive.dp.ua

Abstract. This article presents the results of a study of the Elman neural network to identify defects (delamination) in composite materials and estimate their area. The registration zone is a square matrix consisting of 100 elements. For training, we used images with different defect spot area, moving along the matrix and additively mixed with Gaussian noise of various intensity. The network structure optimal by the criterion of testing error/training time is determined. Testing showed that when the defect spot area changes by more than 8 times and with a noise level of 27%, the test error reaches 6%. This error is significantly decreased when narrowing the range of the defect area changing and reducing the noise intensity. A block diagram of the corresponding intelligent system is proposed.

Keywords: delamination, Elman neural network, images, distortions, noise, identity map, intelligent system

1 Introduction

At present in a number of industries composite materials are widely used due to their high mechanical and thermodynamic properties at low density [1]. Most of them are laminated materials, so the most common defect in them is delamination. To identify and evaluate parameters of such defects in composites, ultrasound diagnostics of products are most often used. With two-sided access it is reasonable to use a shadow testing method with the radiating and sensor elements are located on opposite sides, for example, the product wall. Delamination of the material is a barrier in the path of ultrasound.

A composite material is heterogeneous in structure. The spatial arrangement of reinforcing elements, such as fiber boundless, is often not ideal. Furthermore, in the balk of the material and on its surface, especially at phase boundaries, there are accumulations of pores and various kinds of microdefects. Cracking of the composite matrix and fibers breaks are observed. The concentration of microdefects is especially high at the boundaries of delamination, i.e. in places where the material begins to disintegrate. All these structural imperfections distort the defect image fixed by the sensor element. Therefore, to identify the defect and estimate the true area of the delamination, it is necessary to process its noisy image.

The perspective of using neural networks for these purposes is based on the possibility of parallelizing the processing of information and their ability to learn, i.e. make generalizations. This makes it possible to identify images that have not been encountered in the learning process. In the problem of classification of defects images considered by us, the processing of information by a neural network in a number of key positions is close to the methods of non-parametric statistical training.

Recurrent neural network differs from the classical network of feed-forward by the presence of feedback loops from hidden or output neurons. Their presence has a direct impact on the ability of such networks to learn [2]. The Elman network contains recurrent links of hidden neurons with a layer of context units consisting of unit-delay elements. These elements save the outputs of hidden neurons for one-time step and then transmit them to the inputs of the neurons. This leads to the non-linear dynamic behavior of the network and allows implement a learning process that develops over time. Promising is the integration of a neural network with other information blocks into a single intellectual decision-making system [3].

2 Problem statement

The article aims to create an intelligent system with a core in the form of an Elman neural network to identify and estimate the area of distorted defect images moving within the control zone. In the process of solving this problem, we want to study the ability of recurrent neural networks to identify dynamic noisy images of various sizes defects in the process of scanning by the measuring transducer the product surface.

3 Literature review

Over the years, neural network image processing methods have been successfully used in the field of non-destructive testing, as well as in a number of other engineering fields.

In the article [4], the processes of teaching and testing of a back-propagation neural network to identify distorted defect signals have been analyzed. Using numerical simulation, including a network clustering mechanism, the authors found that clustering increases the probability of signal recognition.

The article [5] presents a deep-learning mechanism for classifying computer-generated images and photographic images. The proposed method accounts for a convolutional layer capable of automatically learning a correlation between neighboring pixels. The layer is designed to subdue the image's content and robustly learn the sensor pattern noise features as well as the statistical properties of images.

The authors of [6] analyze new trends in the digitization of complex engineering drawings. It includes symbol detection and symbol classification. The article [7] presents a symbol recognition method which applied an interactive learning strategy based on the recurrent training of a Hopfield neural network. This method was designed to find the most common symbols in the drawing, which were characterized by having a prototype pattern. The method recursively learns the features of the samples

to the detection and classification accuracy. However, the method can only identify symbols that are formed by a "prototype pattern", which means that irregular shapes cannot be addressed through this framework.

Elman network [8, 9] has been successfully applied in many fields, regarding prediction, modeling, pattern recognition, and control. In [10] Elman network is trained to predict the future value of the residual time series. Finally, the network is used to capture the relationship among the predicted value of the original time series and residual time series.

The standard back-propagation algorithm used in Elman neural network is called Elman's back-propagation algorithm. To increase the convergence, speed a new learning rate scheme is proposed [11].

Multilayer perceptron network (MLP) and Elman neural network were compared in four different time series prediction tasks [12]. Time series include load in an electric network series, which has a low-frequency trend, fluctuations in a far-infrared laser series, numerically generated series, and behavior of sunspots series. The time series is said to be stationary. MATLAB neural network toolbox training functions were used for training MLP and Elman network. Results show that the efficiency of the learning algorithm is a more important factor than the network model used. Elman network models load in an electric network series better than MLP. This network predicts the slope of the trend of the testing data more accurately than the MLP network. In other prediction tasks, it performs similarly to the MLP network.

One of the major problems facing researchers in the recurrent networks is the selection of the hidden neuron's numbers in neural network layers [13].

Jinchuan and Xinzhe [14] investigated a formula: $N_h = (N_{in} + \sqrt{N_p}) \cdot L$, where L is the number of hidden layers, N_{in} is the number of input neurons and N_p is the number of input images.

Kallan R. [15] proposed to calculate the number of neurons in the hidden layer according to the formula: $N_h = 2iK + 1$, where K is the number of network inputs, $i = 1,2,3...$.

Generalization performance varies over time as the network adapts during training. The necessary numbers of hidden neurons approximated in the hidden layer using a multilayer perceptron were found by Trenn [16]. The key points are simplicity, scalability, and adaptability. The number of hidden neurons is $N_h = n + \frac{(n_0 - 1)}{2}$, where n is the number of inputs and n_0 is the number of outputs.

Shibata and Ikeda [17] investigated the effect of learning stability and hidden neurons in neural networks. The simulation results show that the hidden output connection weight becomes small as a number of hidden neurons N_h becomes large. The formula of hidden nodes is $N_h = \sqrt{N_i \cdot N_0}$, where N_i is the number of input neurons and N_0 is the number of output neurons. A tradeoff is formed that if the number of hidden neurons becomes too large, the output of neurons becomes unstable, and if the

number of hidden neurons becomes too small the hidden neuron's output becomes unstable again.

Hunter et al. [18] developed a method in proper neural network architectures. The three networks: MLP, bridged MLP and fully connected cascades network are used. The implemented formula: as follows, $N_h = N + 1$ for the MLP network, $N_h = 2N + 1$ for bridged MLP network, were N is the number of input neurons.

The hidden layer nodes number was determined by using the empirical formula [19] $N_h \le 2N_i + 1$, where N_h is the maximum number of nodes in the hidden layer and N_i is the number of inputs.

In accordance with [20], the optimal number of hidden nodes is found out by trial – and – error approach.

4 Neural network design method

For preliminary processing of information received from the sensor element of the ultrasonic transducer, we used the Elman network, which is an example of a feedback network. The feedback has a profound impact on the learning capacity and the challenge in the design of a neural network is the fixation of hidden neurons with minimal error. The accuracy of training is determined by the parameters: neural network architecture, number of hidden neurons in hidden layers, activation function, inputs number, and weights updating procedure.

In our case, the control zone of the ultrasonic transducer is displayed in the form of a square matrix of 100×100 elements. The Elman network, which performs image processing, has 100 inputs and 100 outputs. In the process of learning the network, we used 8967 images. The results of calculations of the number of neurons in the hidden layer are shown in Table 1.

Table 1. The number of neurons in the hidden layer of neural networks

Source number	Formula for calculating	Number of neurons in the hidden layer
[14]	$N_h = (N_{in} + \sqrt{N_p}) \cdot L$, for $L = 1$	195
[15]	$N_h = 2iK + 1$, for $i = 1$	201
[16]	$N_h = n + \frac{(n_0 - 1)}{2}$	150
[17]	$N_h = \sqrt{N_i \cdot N_0}$	100
[18]	$N_h = N + 1$ $N_h = 2N + 1$	101 201
[19]	$N_h \le 2N_i + 1$	201

The Elman architecture employs a context layer which makes a copy of the hidden layer outputs in the previous time step. The dynamics of the change of hidden state neuron activations in Elman style recurrent networks [13] are given by

$$y_i(t) = f\left[\sum_{k=1}^{K} \upsilon_{ik} y_k(t-1) + \sum_{j=1}^{J} \omega_{ij} x_j(t-1)\right], \quad (1)$$

where y_k and x_j represent the output of the context state neuron and input neuron, respectively; υ_{ik} and ω_{ij} represent their corresponding weights; $f[\cdot]$ is the sigmoid transfer function (1).

Network training was conducted in sequential mode. This mode requires less internal memory for each synaptic link and is preferable for real-time processes. The initialization procedure was carried out without preliminary using the priory information. Herewith, the initial network parameters were set using a generator of uniformly distributed random numbers.

The number of examples for training the network should not be too large, as this can lead to overtraining of the network. In this case, the learning process can end only memorizing learning data. Overtraining network loses the ability to generalize. In accordance with Widrow's rule of thumb, the size of the training set necessary for a good generalization should be of the order of $N = O\left(\frac{W}{\varepsilon_0}\right)$, where W is the total number of free parameters, i.e. synaptic weights and thresholds, ε_0 is the permissible classification error. Let n_{in} be the number of input nodes of the network, and n_h be the number of neurons in the hidden layer. If the product $n_{in} \cdot n_h$ corresponds to the total number of free parameters W network [3], then N must have the order $\frac{n_{in} \cdot n_h}{\varepsilon_0}$. If we take $n_{in} = 100$, $n_h = 200$ and $\varepsilon_0 = 0.02$, then $N = 10^6$. We used 8967 clean and noisy images for training, but each of them is repeated 100 times within the same epoch. Thus, the actually used volume of the training set is 896700, which is close enough to N.

Hidden neurons of the Elman network, learned by the method of back propagation, play the role of feature detectors. Therefore, it is advisable to use this network as a replicator or identity map. At the same time, the input and output layers of our network have the same size $n_{in} = n_{out} = 100$ neurons. The network is fully connected.

A sigmoidal logistic function is used as the transfer function of neurons of the hidden layer. One of the important advantages for us of this function is the high enough speed of calculating the derivative. The network training time largely depends on this. In the process of network learning, we used the conjugate – gradient method. This was due to the need to increase the sufficiently low rate of convergence of the quickest descent method and avoid computational difficulties caused by operations with the Hessian matrix in Newton's method. The computational complexity of the quasi-

Newton methods is estimated as $O(W^2)$. In contrast, the computational complexity of the conjugate – gradient method is estimated as $O(W)$. Thus, in our case when $W = 2 \cdot 10^4$, conjugate – gradient method is more preferable.

5 Flaw detection process simulation

The image of the defect (delamination) has the form of a continuous black spot, which is formed by the n matrix cells with a color level $L = 5$ adjacent to each other. This spot is surrounded by a layer of gray cells with $L = 4$, the next layer of cells corresponds to $L = 3$ and so on to $L = 1$ (white color). White color characterizes the areas of the solid defect-free material that are completely transparent for an ultrasonic signal.

The defect image (pattern) No.1 is a black square with the size 2×2 cells having $L = 5$. The spot area of the defect is $S_1 = 4$. This is the target image for the first class of images. Moving this image on a matrix of 10×10 cells is carried out line by line. Checkpoint of this pattern, like all subsequent ones, is its lower left cell. In the process of moving, the coordinates of this cell for the first row $j = 1$ change from $i = 1$ to $i = 9$, then the same is repeated for $j = 2$ and so on to $j = 9$. Herewith the black spot is not distorted at the edges of the matrix. The number of images used for network training in this case is $N_{1c} = 81$. Here, the index c means that the displayed images are clean, i.e. not containing noise.

In patterns No.2 and No.3, the left and right upper cells of the black square (No.1) are replaced by gray cells with $L = 4$. The number of training images does not change $N_{2c} = N_{3c} = 81$.

Image No.4 is a black square with the size 3×3 cells having $L = 5$. This is the target image for the second class of images. During the scanning process, the checkpoint of this square was moved along the coordinates $i = 1...8$, $j = 1...8$. The number of patterns used for network training in this case is $N_{4c} = 64$. Patterns No.5, 6 and 7 are formed similarly to No.2 and 3. For patterns No. 8 and 9 in the upper layer of the black square, two cells on the left and two on the right are replaced by grey ones.

Image No.10 is a black square with the size 4×4 cells having $L = 5$. This is the target image for the third class of images. The scanning process is similar to that described above. The number of training images is $N_{10c} = 49$. Patterns No.11, 12 and 13 differ from the target pattern No.10 by the alternate replacement of each of the black corner cells with the exception of the checkpoint on the gray. Patterns No.14 and 15 are formed similarly to patterns No.8 and 9 only for the target image No.10.

Image No.16 is a black square of 5×5 cells with $L = 5$. This is the target image for the fifth class of images. The scan coordinates for the checkpoint are $i = 1...6$, $j = 1...6$. The number of training images is $N_{16c} = 36$. Patterns No.17, 18 and 19 are formed similarly to patterns No.11, 12 and 13. For images No. 20, 21, 22, 23, 24 and

25 a pair of cells of the outer black layer, shifting along the periphery of the spot, is replaced by gray ones.

The distortion coefficient of the target image is determined by the ratio of the total area of the cells with the changed color to the area of the corresponding black spot (defect image).

The defect images described above were noisy. The absolute noise level P was set by us as follows:

$$P = \sum_{i=1}^{M} \left| b_{in} - b_{ic} \right|, \tag{2}$$

where b_{ic} is the elements of a clear image matrix and b_{in} is the elements of the noisy image matrix.

Noise is additively mixed with a clean image. We used $P = 10, 30, 50, 70, 90$ levels (2). Considering that the full image contains cells with a gray color gradation from $L = 1$ (white color) to $L = 5$ (black color), the question arises about the distribution of the distortion magnitude $\Delta L = 1, 2, 3, 4$ by the number of distorted image elements. We assume that this distribution is Gaussian with a zero mean value and a standard deviation which is equal to 1. Then the probability density is

$$f(\Delta L) = \frac{1}{2\pi} e^{-\frac{(\Delta L)^2}{2}}. \tag{3}$$

Take into account that $\Delta L = 0$ corresponds to the absence of distortion of the image element, we get the following:

$$P = f(|\Delta L| = 1) + f(|\Delta L| = 2) + f(|\Delta L| = 3) + f(|\Delta L| = 4). \tag{4}$$

The corresponding distribution of ΔL by the number of distorted image elements q is shown in Table 2 (3), (4).

Table 2. The number of distorted image elements

q						γ, %
$\Delta L = -1$	$\Delta L = +1$	$\Delta L = -2$	$\Delta L = +2$	$\Delta L = -3$	$\Delta L = +3$	
4	4	1	1	-	-	10
12	12	3	3	-	-	17
20	20	5	5	-	-	22
28	28	7	7	-	-	27
36	36	8	8	1	1	31

The relative value of the noise intensity we calculated according to Table 2.

Herewith

$$\gamma = \frac{\sigma}{L_{\max}} \cdot k\,, \quad \sigma = \sqrt{\frac{1}{100}\sum_{1}^{100}(\Delta L_i)^2}\,, \quad L_{\max} = 5 \tag{5}$$

The value of the normalizing factor k was determined based on the following. If in all 100 cells of the matrix the black color with $L = 5$ (presence of a defect) is changed to white with $L = 1$ (defect-free material), then $\Delta L_i = 4$ and, in accordance with (5), $\sigma = 4$. Herewith $\frac{\sigma}{L_{\max}} = \frac{4}{5}$. Taking such a change in L in the entire matrix as a 100% error corresponding to $\gamma = 1$, we get $k = 5/4 = 1.25$. In the article, we used the following restrictions: if the total value of the image intensities and the noise in this cell of the matrix exceeds the maximum level $L_{\max} = 5$, then $L = 5$, if the summary value was below $L_{\min} = 1$, then $L = 1$ was kept.

6 Defect image capture

The outputs of the Elman neural network are connected to the inputs of a logical matrix connected in turn to a combinational circuit, which has four outputs by the number of classes of identifiable images. The functional scheme of the intelligent system for defect identification is shown in Figure 1.

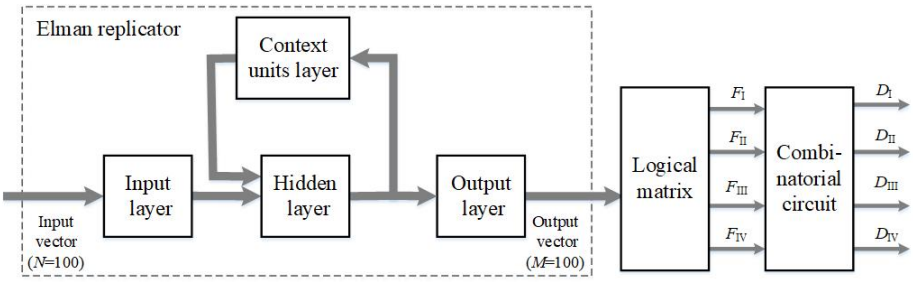

Fig. 1. Intelligent system for defect identification

To estimate the area of defect, we propose the following methodology: the matrix (10×10 elements) represents the control zone of an ultrasonic transducer operating in shadow mode and "sounding", for example, the product wall. The full image of this zone is captured by the sensor unit of the transducer.

During physical scanning of the product surface by the transducer, a noisy image of a defect in the form of a fuzzy dark spot may appear in the lateral part of the matrix. By a local displacement of the transducer, this spot must be moved to the central part of the matrix. The neural network allows you to significantly clear the image of the defect from the noise during its movement to the center of the matrix and make this image more distinct.

In the center of the matrix, there are positioned defect images capture zones, as shown in Figure 2. These zones correspond to images No.1 (2×2 elements), No.4 (3×3 elements), No.10 (4×4 elements), and No.16 (5×5 elements), which are targeted for the respected classes. The logic matrix has 100 inputs, corresponding to the number of outputs of the neural network that perform the function of a replicator. These inputs are numbered in accordance with the pair jk, where j denotes the row number and k denotes the column number of the matrix, as shown in Figure 2.

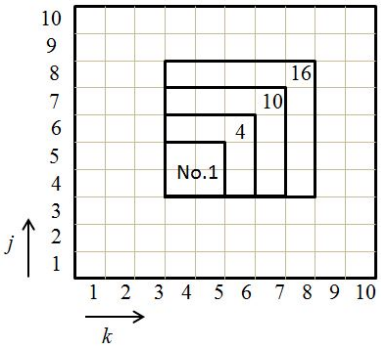

Fig. 2. Capture zones for defect images

The Boolean function F_{I}, implemented at the same output of the logic matrix, describes images of class I, to which images No. 1, 2, and 3 belong.

$$F_{\text{I}} = y_{44} \cdot y_{45} \cdot (y_{54} \vee y_{55}). \tag{6}$$

Herewith $F_{\text{I}} = 1$, if elements y_{44} and y_{45} have a level of $L = 5$ (black color), which corresponds to a logic-1 level, and at least one of the elements y_{54} or y_{55} has the same level (6).

The Boolean function F_{II} describes images of class II, to which images No. 4, 5, 6, 7, 8, and 9 belong.

$$F_{\text{II}} = y_{44} \cdot y_{45} \cdot y_{54} \cdot y_{55} \cdot y_{56} \cdot y_{65} (y_{64} \vee y_{66} \vee y_{46}). \tag{7}$$

Herewith $F_{\text{II}} = 1$, if all elements involved in conjunction have a level of $L = 5$, and at least one of the corner elements of the target image (No.4), i.e. one of the disjunctive members in brackets had the same level $L = 5$ (7).

The Boolean function F_{III} describes images of class III, to which images No.10, 11, 12, 13, 14, and 15 belong.

$$F_{\text{III}} = y_{44} \cdot y_{45} \cdot y_{46} \cdot y_{54} \cdot y_{55} \cdot y_{56} \cdot y_{57} \cdot y_{64} \cdot y_{65} \cdot y_{66} \cdot y_{67} \cdot y_{75} \cdot y_{76} \cdot (y_{47} \vee y_{74} \vee y_{77}). \tag{8}$$

Disjunctive terms of expression (8) correspond to the corner elements of the target image No. 10.

The Boolean function F_{IV} describes images of class IV, to which images No. 16 – No. 25 belong.

$$F_{IV} = y_{44} \cdot y_{45} \cdot y_{46} \cdot y_{54} \cdot y_{55} \cdot y_{56} \cdot y_{57} \cdot y_{64} \cdot y_{65} \cdot y_{66} \cdot y_{67} \cdot y_{68} \cdot y_{75} \cdot y_{76} \cdot y_{77} \times \\ \times \left(y_{74} \cdot y_{84} \cdot y_{85} \vee y_{87} \cdot y_{88} \cdot y_{78} \vee y_{47} \cdot y_{48} \cdot y_{58}\right). \quad (9)$$

When forming the $F_I - F_{IV}$ functions, we made assumptions about the optional presence of all corner elements of the target images in the defect image capture zones. This is consistent with control practice.

The analysis of expressions (6) – (9) shows that the patterns described by them are sequentially nested one into another, namely F_I is included in F_{II}, F_{II} — in F_{III}, F_{III} — in F_{IV}. Thus, when fixing an image of a senior class, for example No. 16, logical unit will be present not only at the output F_{IV} of the logical matrix, but also at outputs F_{III}, F_{II}, and F_I. The combinational circuit shown in Figure 3 helps to eliminate this effect. The circuit uses inverters and four–input conjunctors and provides separate fixation of each class of image. Herewith, the appearance of a logical unit at one of the outputs $D_I - D_{IV}$ of the combinational circuit indicates the capture of the defect image of the corresponding class.

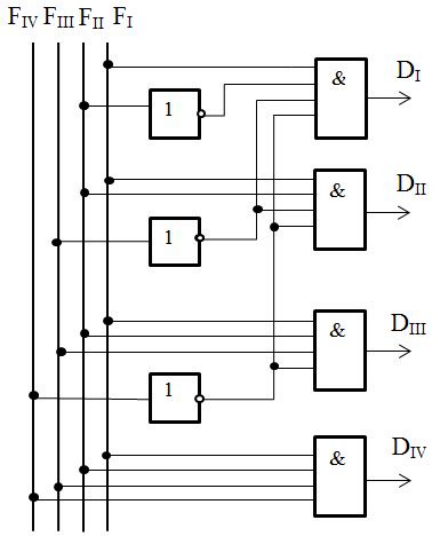

Fig. 3. The combinational circuit

The proposed system of the defect area estimation allows monitoring both in the manual scanning mode with indication of the defect magnitude by sound or light signals and automating the testing process. In the latter case, when scanning the product along parallel paths with a given step, it is easy to automatically register the area.

7 Experiments and results

To simulate the operation of a neural network, we developed the program functioning in the Matlab environment. Network training was carried out sequentially in three stages.

1. First, a clean image No. 1 located in the lower-left corner of the matrix is fed to the network input. This image then moves sequentially along the rows of the matrix within $i = 1 \ldots 9$, $j = 1 \ldots 9$. Here i and j are the horizontal and vertical coordinates of the matrix, respectively. The specified scan coordinates describe the movement of the lower-left cell of the image.
2. At the second stage, a clean image No.1 ($\gamma = 0$) was again fed to the network input and it was moved along the matrix as described in paragraph 1. Then, image No. 1, additively mixed with noise corresponding to $\gamma = 10\%$, was fed to the network input, and this noisy image was moved along the matrix. We repeated this process sequentially for all remaining noise levels, up to $\gamma = 31\%$. It should be noted that this scanning process was repeated 100 times for each value for the noise intensity.
3. After completing the second stage of training, in order to verify the absence of retraining effect, we repeated paragraph 1.

The network training process described above we fulfilled with each pattern from the training set. Table 3 shows the data on the image classes used for training.

Table 3. Image classes

Class number	Target image and its area S	Distorted images	The area of the undistorted part of the image	Distortion factor η,%	Number of training images
I	Image No. 1, S=4	No. 2, 3	3	25.0	170100
II	Image No. 4, S=9	No. 5, 6, 7	8	11.1	268800
		No. 8, 9	7	22.2	
III	Image No. 10, S=16	No. 11, 12, 13	15	6.25	205800
		No. 14, 15	14	12.5	
IV	Image No. 16, S=25	No. 17, 18, 19	24	4.0	252000
		No. 20, 21, 22, 23, 24, 25	23	8.0	

Figure 4 shows a graph of the dependence of the training time, relative to one pattern α on the area of the black spot S, reflecting the area of the defect. But at the same time, α characterized training on clean images, without noise. Training on noisy images is carried out after the training on clean images. And here the training time relative to one pattern β is significantly less. So, for images of the first type,

$\theta_1 = \frac{\beta}{\alpha} = 0.045$. For images of the second, third and fourth types, we have: $\theta_2 = 0.027$, $\theta_3 = 0.032$, and $\theta_4 = 0.021$.

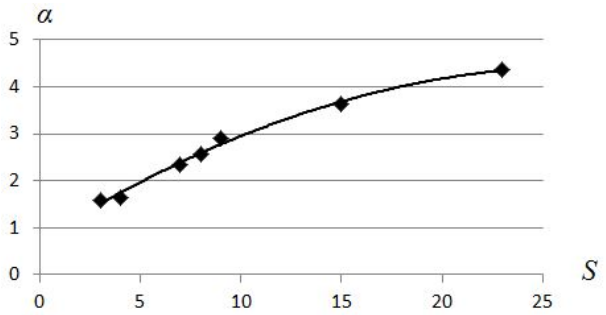

Fig. 4. Dependence of the training time for one pattern α on the area of the black spot S

Thus, we can state that the network training time at the second stage, assigned to one noisy image, is ten times less than the corresponding time spent at the first stage when training the network at the clean image.

The phenomenon of retraining the network was not observed.

Network testing was carried out by moving images from the set described above (No. 1 – No. 25) with noise levels from Table 2 along with the matrix. Testing results showed the following.

If testing was carried out after a full cycle of network training on patterns of one class using then a noisy image with $\gamma = 31\%$ from the same class, then the test error did not exceed 3%. This result is valid for all four classes.

When combining images of classes I and II, i.e. with the expansion of the defect spot area from $S = 3$ to $S = 9$, the testing error on noise patterns ($\gamma = 31\%$) increased to 8%. When combining all the images of I-IV classes in a single training set, the maximum testing error increased to 12%. And in the first and second cases, the parameter β increased significantly, its value became comparable with α. The reduction of the noise level during testing significantly decreases this error. So, with a reduction in γ from 31 to 27%, the test error decreases by 40-50%, from 31 to 22% – by 60-70% and so on.

The above results were obtained by using the Elman neural network as a replicator with $n_{in} = n_{out} = 100$, $n_h = 200$ and the number of layers $l = 1$. An increase in the number of neurons in the hidden layer to 400 on average by 25% reduces the testing error, however, the network training time increased by 30%. A network with two hidden layers, while reducing the test error by 20%, trained 4 times longer than a network with one hidden layer. Training a network with three hidden layers took 14 times more time than the single-layer network we used.

8 Conclusion

We investigated the Elman neural network as an identity map (replicator) for the task of identifying and estimating the area of a delamination in composite materials in the process of ultrasonic testing. Twenty-five images were used as training patterns. They were divided into four classes, according to the size of their defect area. Deterministic distortions caused by the nature of the testing object were introduced into the images of each class. Each image was moved step by step on a square matrix consisting of 100 cells, which represents the registration area of the ultrasound transducer. Images were additively mixed with white Gaussian noise, the intensity of which varied from 10 to 31%. At each scan point, the noise was superimposed 100 times.

The process of training the network began with presenting it with the clean (without any noise) patterns. In the process of training the network, it was found that the training time with clean patterns, assigned to one pattern fixed in a given position inside the matrix, increases with increasing the pattern area. Network training using noisy patterns was carried out after training with clean patterns. In this case, the training time, defined in the same way, turned out to be ten times shorter.

Network testing has provided the following. If testing was carried out after a full cycle of network training on patterns of one class using a test pattern with a noise level of 31%, then the test error did not exceed 3% for any class. When combining images of classes I and II, i.e. with the expansion of the defect spot area from $S = 3$ to $S = 9$, the testing error increased to 8%. When combining the images of all four classes in a single training set, i.e. with the expansion of the defect spot area from $S = 3$ to $S = 25$, the maximum testing error increased to 12%. The reduction of the noise level during testing significantly decreases this error. So, with a reduction in γ from 31 to 27%, the test error decreases by 40-50%, from 31 to 22% – by 60-70% and so on.

These results were obtained by modeling the Elman's neural network with the number of input and output neurons $n_{in} = n_{out} = 100$ and with the number of neurons in one hidden layer $n_h = 200$. This network structure turned out to be optimal by the criterion of error testing/training time.

To estimate the delamination area, four nested defect spot capture zones were implemented in the central part of the matrix. For this purpose, a cascade-connected logical matrix and the combinational circuit having four outputs by the number of classes of identifiable images are connected to the output of the neural network.

Estimation of the defect area serves as the basic for determining the residual strength of a given unit of a product and making a decision on the possibility of its further operation.

References

1. Composite Materials Handbook – 17: Polymer Matrix Composites, vol. 1-3. SAE International (2018)

2. Robinson, D.A.: Signal processing by neural networks in the control of eye movement. Computational Neuroscience Symposium, Indiana University – Purdue University at Indianapolis, pp. 73-78 (1992)
3. Naykin, S.: Neural Networks. A Comprehensive Foundation, second edition. Prentice Hall, New Jersey (2008)
4. Khandetskyi, V., Antonyuk, I.: Signal processing in defect detection using back-propagation neural networks. NDT&E International, 35, 483-488 (2002)
5. Chawla, C., Panwar, D., Anand, G.S., Bhatia, M.P.S.: Classification of computer generated images from photographic images using convolutional neural networks. International Journal of Computer and Informational Engineering, vol. 12, No. 10, pp. 822-827 (2018)
6. Moreno-Garcia, C.F., Elyan, E., Jayne, C.: New trends on digitalization of complex engineering drawings. Neural Computing and Applications, June, pp. 1-18 (2018)
7. Ablameyko, A.V., Uchida, S.: Recognition of engineering drawing entities: review of approaches. International Journal Image Graph 07(04): 709-733 (2007)
8. Elman, J.L.: Finding structure in time. Cognitive Science, vol. 14, pp. 179-211 (1990)
9. Elman, J.L., Bates, E.A., Johnson, M.H., Karmiloff-Smith, A., Parisi, D., Plunkett, K.: Rethinking Innateness: A Connectionist Perspective on Development. Cambridge, MA: MIT Press (1996)
10. Ardalani-Farsa, M., Zolfaghari, S.: Chaotic time series prediction with residual analysis method using hybrid Elman-NARH neural networks. Neurocomputing, vol. 73, Iss. 13-15, Aug., pp. 2540-2553 (2010)
11. Ren, G., Cao, Y., Wen, S., Huang, T., Zeng, Z.: A modified Elman neural network with a new learning rate scheme. Neurocomputing, vol. 286, Iss. 19, April, pp. 11-18 (2018)
12. Koskela, T., Lehtokangas, M., Saarinen, J., Kaski, K.: Time series prediction with multilayer perceptron, FIR and Elman neural network. Tampere university of technology, Electronics laboratory, FIN-33101, Tampere, Finland (2017)
13. Sheela, K.G., Deepa, S.N.: Review of methods to fix number of hidden neurons in neural networks. Mathematical Problems in Engineering, vol. 12, article ID 425740, pp. 1-11 (2013)
14. Jinchuan, K., Xinzhe, L.: Empirical analysis of optional hidden neurons in neural network modeling for stock prediction. In Proceedings of the Pacific-Asia Workshop on Computional Intelligence and Industrial Application, vol.2, pp. 828-832, Dec. (2008)
15. Kallan, R.: Main Conceptions of Neural Networks. Williams Publ., New York (2001)
16. Trenn, S.: Multilayer perceptrons: approximation order and necessary number of hidden units. IEEE Transactions on Neural Networks, vol. 19, No 5, pp. 836-844 (2008)
17. Shibata, K., Ikeda, Y.: Effect of number of hidden neurons on learning in large scale layered neural networks. In Proceedings of the ICCAS-SICE International Joint Conference (ICCAS-SICE'09), pp. 5008-5013, Aug. (2009)
18. Hunter, D., Yu, H., Pukish, M.S., Kolbusz, J., Wilamowski, B.M.: Selection of proper neural network sizes and architectures: a comparative study. IEEE Transactions on Industrial Informatics, vol.8, No 2, pp. 228-240 (2012)
19. Gavin, I., Holder, R., Graeme, C.: Input detection for neural network models in water resources applications, part 2. Case study: forecasting salinity in a river. J. Hydrol 301 (1-4): 93-107 (2011)
20. Devi, R., Rani, B.S., Prakash, V.: Role of hidden neurons in an Elman recurrent neural network in classification of cavitation signals. International Journal of Computer Applications, vol.37, no 7, pp. 9-13, (2012)