Chapter 12
Determination of Elastic and Dissipative Properties of Material Using Combination of FEM and Complex Artificial Neural Networks

A. N. Soloviev, N. D. T. Giang and S.-H. Chang

This paper describes the application of complex artificial neural networks (CANN) in the inverse identification problem of the elastic and dissipative properties of solids. Additional information for the inverse problem serves the components of the displacement vector measured on the body boundary, which performs harmonic oscillations at the first resonant frequency. The process of displacement measurement in this paper is simulated using calculation of finite element (FE) software ANSYS. In the shown numerical example, we focus on the accurate identification of elastic modulus and quality of material depending on the number of measurement points and their locations as well as on the architecture of neural network and time of the training process, which is conducted by using algorithms RProp, QuickProp.

A. N. Soloviev (✉) · N. D. T. Giang
Department of Strength of Materials, Don State Technical University, Rostov-on-Don, Russia 344000
e-mail: solovievarc@gmail.com

A. N. Soloviev
Vorovich Mechanics and Applied Mathematics Research Institute, Southern Federal University, Rostov-on-Don, Russia

A. N. Soloviev
Department of Mathematics, Mechanics and Computer Sciences, Southern Federal University, Rostov-on-Don, Russia

A. N. Soloviev
Department of Mechanics of Active Materials, Southern Scientific Centre of Russian Academy of Sciences, Rostov-on-Don, Russia

N. D. T. Giang
Department of Information Technology, Vietnam Maritime University, HaiPhong, Vietnam

S.-H. Chang
Department of Microelectronics Engineering, National Kaohsiung Marine University, Kaohsiung, Taiwan

S.-H. Chang et al. (eds.), *Advanced Materials*, Springer Proceedings in Physics 152,
DOI: 10.1007/978-3-319-03749-3_12,

12.1 Introduction

The artificial neural networks (ANN) [1] are widely used in different regions of science and industry. One of their applications in mechanics is solving the coefficient inverse problem of identifying elastic [2–6] and dissipative properties of solids [7–11]. In the early 1990s, the complex artificial neural networks (CANN) were proposed in the works of T. Nitta [12, 13], which then have been widely used for various applications [14–16]. The CANN parameters (weights, threshold values, inputs and outputs) are complex numbers, which used in various fields of modern technology, such as optoelectronics, filtering, imaging, speech synthesis, computer vision, remote sensing, quantum devices, spatial–temporal analysis of physiological neural devices and systems. Application of CANN in mechanics is a new research area, which has been developed over the last few years.

This paper describes the application of CANN in solving coefficient inverse problem of identifying elastic (Young's modulus) and dissipative (quality factor) properties of solids. Additional information for solving this inverse problem serves the components of the displacement vector measured at the boundary of the body (in a discrete set of points), which performs harmonic oscillations in the vicinity of the first resonant frequency. The process of displacement measurement in this paper is simulated using calculation of finite element (FE) software ANSYS. In the numerical example shown below, the accuracy of identifying mechanical properties of the material depends on the number of measurement points and their locations, as well as on the architecture of neural network and time of the training process (which is conducted by using algorithms CBP)

12.2 Formulation of Identification Inverse Problems for Mechanical Properties

In the plane problem of elasticity theory, the steady-state harmonic oscillations of a rectangular area ($a \times b$) with the angular frequency ω are considered (Fig. 12.1). The rectangle is fixed on the left side boundary S_u, the vertical force F_0 is applied to the lower right corner, the rest of the right boundary S_t is free from mechanical stresses. The mechanical properties of the material are described by Young's elastic modulus, E, Poisson's ratio, ν, and quality factor, Q.

The vibration equations of the solid have the form [18]:

$$\sigma_{kj,j} = -\rho(\omega^2 - i\omega\alpha)u_k; \quad k,j = 1,2;$$

$$\sigma_{kk} = c_{kj}(1 + i\omega\beta)\varepsilon_{jj},\ \sigma_{12} = c_{44}(1 + i\omega\beta)\varepsilon_{12},\ \varepsilon_{kj} = \frac{1}{2}\left(u_{k,j} + u_{j,k}\right). \tag{12.1}$$

The boundary conditions suppose the presence of a force at the point 2 (Fig. 12.1) and similar conditions in the rest of the boundary:

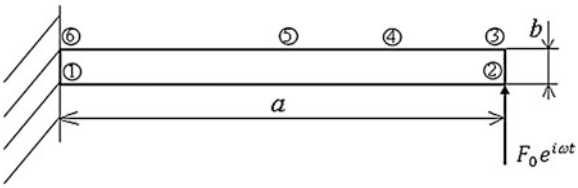


Fig. 12.1 Statement of the problem, the designations are interpreted into text

$$u_k\big|_{S_u} = 0, \quad \sigma_{kj} n_j\big|_{S_t} = 0. \tag{12.2}$$

The additional information to solve the inverse problem are the displacements, measured at the points 2–5 (Fig. 12.1):

$$u_1 = U_{mr} + iU_{mi}, \qquad u_2 = V_{mr} + iV_{mi}, \qquad m = 2, 3, 4, 5. \tag{12.3}$$

Here σ_{kj}, ε_{kj} are the stress and strain tensor components, respectively, ρ is the density. The elastic coefficients correspond to the isotropic body: $c_{11} = c_{22} = \lambda + 2\mu$, $c_{12} = c_{21} = \lambda$, $c_{44} = \mu$, λ, μ are the Lamé coefficients. The applied vibration frequency ω is matched with proper first resonance frequency, which does not take into account the dissipation of mechanical energy. Coefficients α, β characterize dissipation calculated according to the method [17].

12.3 Structure of CANN

Structure of CANN in solving coefficient inverse problem of identification of the elastic (Young's modulus) and dissipative (quality factor) properties of solids (Fig. 12.2).

This CANN is formed in three layers, called the input layer, hidden layer, and output layer. Each of layers consists of one or more nodes, represented in the diagram by the small circles, as shown in Fig. 12.2. The lines between the nodes indicate the flow of information from one node to the next (with weights W, $\breve{W}$).

When neural networks are trained, they works basically in the following way.

(1) The first step in training a network is to initialize the network connection weights by small random values (−1.0, 1.0).
(2) The input pattern ($X_1 = X_{1r} + iX_{1i}$, $X_2 = X_{2r} + iX_{2i}$, $X_n = X_{nr} + iX_{ni}$) on the base of which the network will be trained are presented at the input layer of the net and the net is run normally to see what output pattern it actually produces.
(3) The actual output pattern are compared to the desired output pattern for corresponding input pattern. The differences between actual and desired patterns form the error pattern:

$$Y_n = \sum_m W_{nm} X_m = X + iY = Z; \tag{12.4}$$

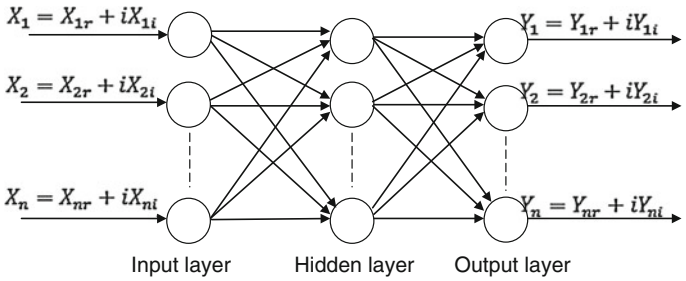


Fig. 12.2 Structure of CANN

$$O_n = F_c(z) = F(x) + iF(y) = \frac{1}{1+\exp(-X)} + i\frac{1}{1+\exp(-Y)}; \tag{12.5}$$

$$\begin{aligned} Ep &= \left(\frac{1}{N}\right)\sum\nolimits_{n=1}^{N} |T_n - O_n|^2 \\ &= \frac{1}{2}\sum\nolimits_{n=1}^{N} \left(|Re(T_n) - Re(O_n)|^2 + |Im(T_n) - Im(O_n)|^2\right), \end{aligned} \tag{12.6}$$

where W_{nm} is the (complex-valued) weight connecting neuron n and m, X_m is the (complex-valued) input signal from neuron m, F is the sigmoid function, Ep is the square error for the pattern p, O_n are the output values of the input neuron, T_n is the target pattern.

(4) The error pattern is used to adjust the weights on the output layer so that the error pattern reduced the next time if the same pattern was presented at the inputs:

$$W_{nm} = W_{nm} + \Delta W_{nm} = W_{nm} + \overline{H_m}\Delta\lambda_n; \tag{12.7}$$

$$\Delta\lambda_n = \varepsilon\{Re[\delta^n](1 - Re[O_n])Re[O_n] + iIm[\delta^n](1 - Im[O_n])Im[O_n]\}, \tag{12.8}$$

where λ_n is the threshold value of the output neuron n, $\delta^n = T_n - O_n$ is the difference (error) between the actual pattern and the target pattern, H_m are the output values of the hidden neuron.

(5) Then the weights of the hidden layer are adjusted similarly, it is stated in this time what they actually produce the neurons in the output layer to form the error pattern for the hidden layer:

$$\breve{W}_{lm} = \breve{W}_{lm} + \Delta\breve{W}_{lm} = \breve{W}_{lm} = \overline{\mathrm{I}_l}\Delta\theta_m; \tag{12.9}$$

$$\begin{aligned} \Delta\theta_m = \varepsilon\{&(1 - Re[H_m])Re[H_m]\sum\nolimits_n (Re[\delta^n](1 - Re[o_n])\,Re[W_{nm}] \\ &+ Im[\delta^n](1 - Im[o_n])Im[o_n]Im[W_{nm}])\} - i\varepsilon\{(1 \\ &- Im[Hm])Im[Hm]\sum\nolimits_n (Re[\delta^n](1 - Re[o_n])Re[o_n]Im[W_{nm}] \\ &- Im[\delta^n](1 - Im[o_n])Im[o_n]Re[W_{nm}])\} \end{aligned} \tag{12.10}$$

Table 12.1 Data from problem of modal and harmonic analysis using ANSYS

No	Input data		ω	Output data (m)			
	E (Pa)	Q		U_{2r}	U_{2i}	V_{2r}	V_{2i}
1	5×10^9	10	206.35	4.82×10^{-7}	-5.2×10^{-6}	1.09×10^{-6}	-7.3×10^{-5}
2	10×10^9	20	291.83	2.41×10^{-7}	-2.6×10^{-6}	5.46×10^{-7}	-3.6×10^{-5}
3	15×10^9	30	357.41	1.60×10^{-7}	-1.7×10^{-6}	3.64×10^{-7}	-2.4×10^{-5}
4	20×10^9	40	412.7	1.20×10^{-7}	-1.3×10^{-6}	2.73×10^{-7}	-1.8×10^{-5}
5	25×10^9	50	461.42	9.64×10^{-8}	-1×10^{-6}	2.18×10^{-7}	-1.5×10^{-5}
6	30×10^9	60	505.46	8.03×10^{-8}	-8.6×10^{-7}	1.82×10^{-7}	-1.2×10^{-5}
7	35×10^9	70	545.96	6.89×10^{-8}	-7.4×10^{-7}	1.56×10^{-7}	-1×10^{-5}
8	40×10^9	80	583.65	6.02×10^{-8}	-6.5×10^{-7}	1.36×10^{-7}	-9.1×10^{-6}
9	45×10^9	90	619.06	5.35×10^{-8}	-5.8×10^{-7}	1.21×10^{-7}	-8.1×10^{-6}
Etc.							

where $\breve{W}_{lm}$ is the weight between input neuron l and hidden neuron m, I_l are the output values of the input neuron l, θ_m is the threshold value of the hidden neuron m.

(6) The network is trained by presenting each input pattern in turn at the inputs and propagating forward and backward, followed by the next input pattern. Then this cycle is repeated many times. The training stops when any of these conditions occurs: the maximum number of epochs (repetitions) is reached or the error has an acceptable training error threshold.

After network training, a stable network structure is established. The network then can be used to predict output values from new input values, i.e. to solving coefficient inverse problem of identifying elastic and dissipative properties of solids.

12.4 Application of CANN to Solve Coefficient Inverse Problem for Identification of Elastic (Young's Modulus) and Dissipative (Quality Factor) Properties of Solids

In order to identify Young's modulus and quality factor as it was described above used CANN, in which they were output data. As input data, we used displacement amplitudes (12.3), measured on solid surfaces. By training the CANN with a set of input and output data, the first step of the computation process was calculation of natural resonance frequencies, and following definition of steady oscillations of the solid at these frequencies or in their neighborhood with the mechanical energy dissipation.

Figure 12.1 shows four points 2, 3, 4, 5, where displacement amplitudes were measured. The process of the displacement measurements is simulated using

Table 12.2 The results of training and testing for CANN with two and six input values at the different points (with the same number of epochs)

No	Number of data	CANN architecture	Epoch	Error	Accuracy (%)	Notes
1	2000	2-4-1	5000	0.01918	91.18	at the point 2
2	2000	2-4-1	5000	0.01856	93.52	at the point 3
3	**2000**	**2-4-1**	**5000**	**0.01853**	**93.88**	**at the point 4**
4	2000	2-4-1	5000	0.01872	92.95	at the point 5
5	2000	6-4-1	5000	0.01870	92.97	at the three points (x_1, y_1), (x_2, y_2), (x_3, y_3)
6	2000	6-4-1	5000	0.01883	93.64	for three frequencies (ω_{r-k}, ω_r, ω_{r+k}) at the point 2

Table 12.3 The results of training and testing for CANN with two input values at the point 4 (with different epochs)

No	Number of data	CANN Architecture	Epoch	Error	Accuracy (%)	Notes	Time (second)
1	2000	2-4-1	1000	0.02953	91.85	at the point 4	365.5
2	2000	2-4-1	5000	0.01853	93.88	at the point 4	1949.3
3	**2000**	**2-4-1**	**20000**	**0.01390**	**94.89**	**at the point 4**	**13735.5**

Table 12.4 The results of training and testing for CANN with two input values at the point 5 (with different numbers of hidden layers)

No	Number of data	CANN Architecture	Epoch	Error	Accuracy (%)	Notes	Time (second)
1	2000	2-2-1	1000	0.03012	89.72	at the point 5	269.5
2	**2000**	**2-4-1**	**1000**	**0.02653**	**91.85**	**at the point 5**	**365.5**
3	2000	2-6-1	1000	0.02955	89.50	at the point 5	476.5
4	2000	2-8-1	1000	0.02683	91.52	at the point 5	629.8
5	2000	2-10-1	1000	0.02761	91.20	at the point 5	739.1

Table 12.5 The results of training and testing for CANN with six input values at the points 3, 4, 5 (with different numbers of hidden layers)

No	Number of data	CANN architecture	Epoch	Error	Accuracy (%)	Notes	Time (second)
1	2000	6-2-1	1000	0.02485	91.17	at three points	364.6
2	2000	6-4-1	1000	0.02163	90.32	at three points	495.8
3	**2000**	**6-6-1**	**1000**	**0.02056**	**93.83**	**at three points**	**659.3**
4	2000	6-8-1	1000	0.02283	91.52	at three points	830.0
5	2000	6-10-1	1000	0.03763	85.89	at three points	3912.0

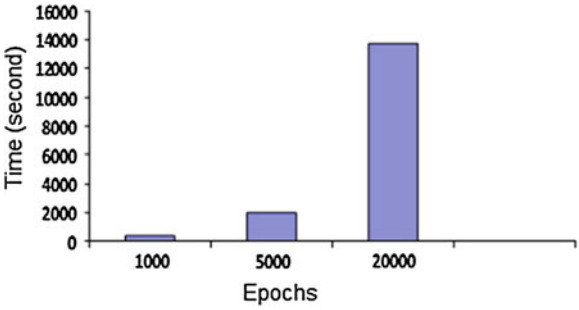


Fig. 12.3 Relationship between time of training and testing for CANN with different epochs

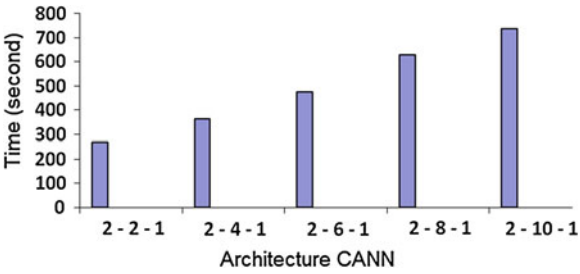


Fig. 12.4 Relationship between time of training and testing for CANN with different architecture

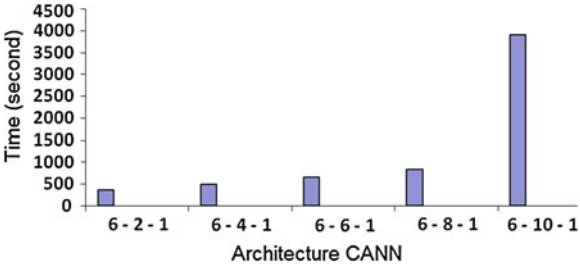


Fig. 12.5 Relationship between time of training and testing in CANN with different architecture

calculation of finite element software ANSYS. Table 12.1 presents the input and output data for the "measurement" at the point 2.

CANN has the following configuration in dependence on number of points and the frequency "measurement":

(i) at one measurement point and one frequency, this neural network consists of two neural input values: $\boldsymbol{U}_{2r} + \boldsymbol{iU}_{2i}$, $\boldsymbol{V}_{2r} + \boldsymbol{iV}_{2i}$ and one neural output value: $\boldsymbol{E} + \boldsymbol{iQ}$;

(ii) at one measurement point and three frequencies, this neural network consists of six neural input values: $\boldsymbol{U}_{1r} + \boldsymbol{iU}_{1i}, \boldsymbol{V}_{1r} + \boldsymbol{iV}_{1i}$, $\boldsymbol{U}_{2r} + \boldsymbol{iU}_{2i}, \boldsymbol{V}_{2r} + \boldsymbol{iV}_{2i}$, $\boldsymbol{U}_{3r} + \boldsymbol{iU}_{3i}, \boldsymbol{V}_{3r} + \boldsymbol{iV}_{3i}$ and one neural output value: $\boldsymbol{E} + \boldsymbol{iQ}$;

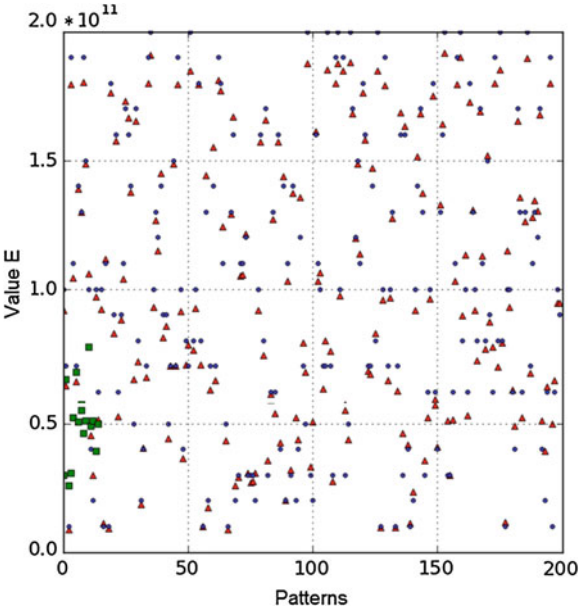


Fig. 12.6 Results of testing E at the threshold error of 20 %

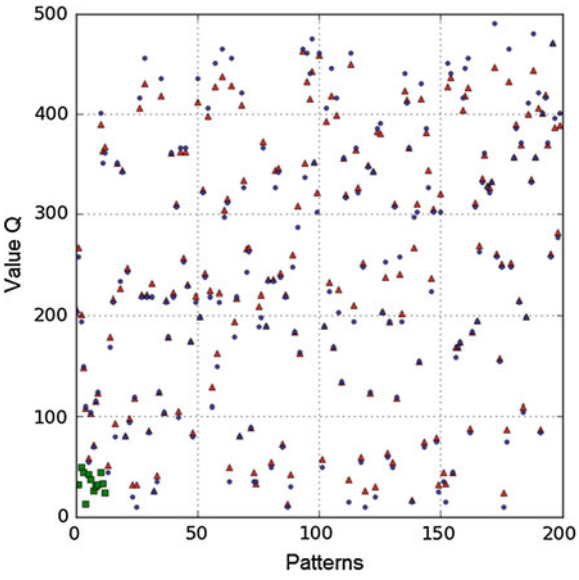


Fig. 12.7 Results of testing Q at the threshold error of 20 %

(iii) at three measurement points and one frequency, this neural network consists of six neural input values: $\boldsymbol{U}_{1r} + \boldsymbol{iU}_{1i}, \boldsymbol{V}_{1r} + \boldsymbol{iV}_{1i}$, $\boldsymbol{U}_{2r} + \boldsymbol{iU}_{2i}, \boldsymbol{V}_{2r} + \boldsymbol{iV}_{2i}$, $\boldsymbol{U}_{3r} + \boldsymbol{iU}_{3i}, \boldsymbol{V}_{3r} + \boldsymbol{iV}_{3i}$ and one neural output value: $\boldsymbol{E + iQ}$.

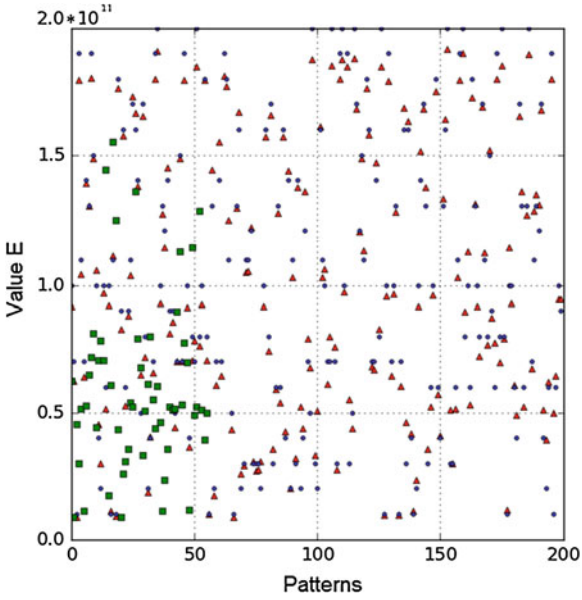


Fig. 12.8 Results of testing E at the threshold error of 10 %

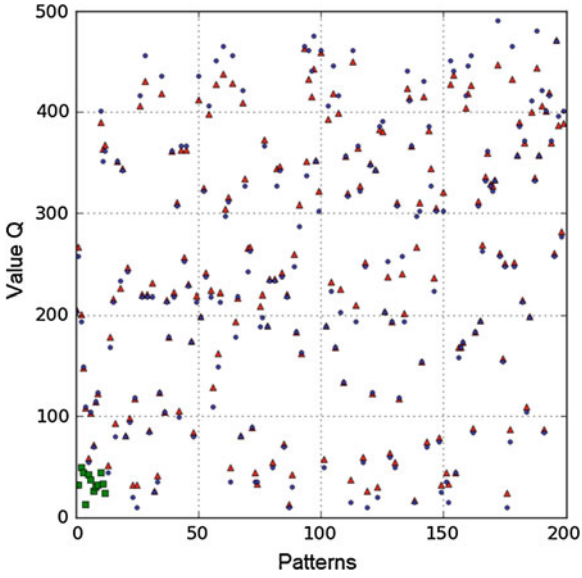


Fig. 12.9 Results of testing Q at the threshold error of 10 %

The results of the computer experiments, conducted by using CANN with 2000 patterns, in which, 1800 patterns were used for training and 200 for testing, are show in Tables 12.2, 12.3, 12.4 and 12.5.

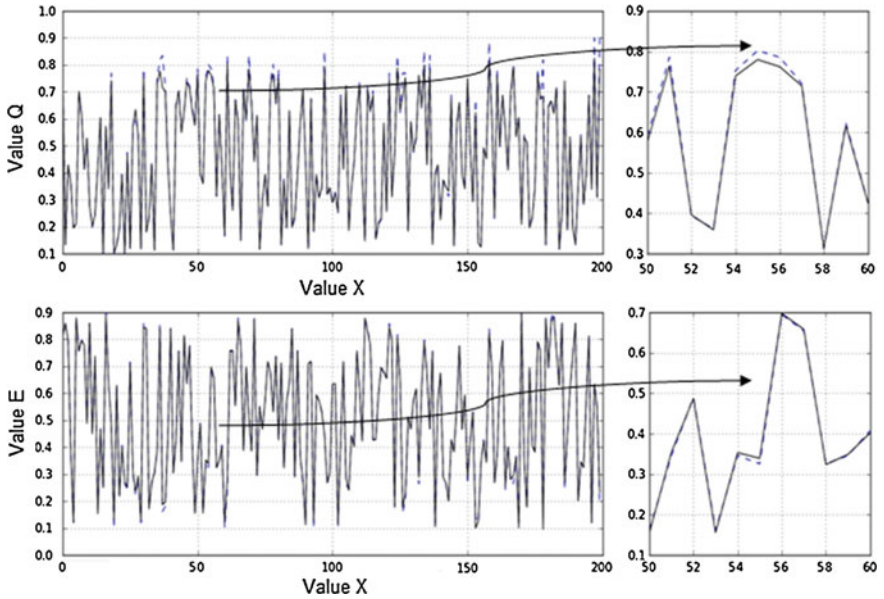
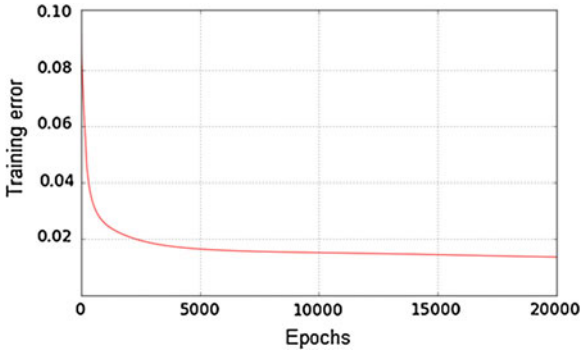


Fig. 12.10 Prediction of results obtained by using CANN

Fig. 12.11 Graph of training error



Time of training for CANN with different epochs (Table 12.3) is shown in Fig. 12.3. Time of training and testing for CANN with different number of neurons in hidden layer is shown in Fig. 12.4. The relationship between time of training and testing for CANN with different architecture is shown in Fig. 12.5 (data from Table 12.5).

Figures 12.6, 12.7, 12.8 and 12.9 show the results of testing with 200 patterns in CANN (with architecture 2-4-1, see Table 12.2, the row 3). The circles present the data of test E, triangles and squares describe the forecast data E_c, obtained by using

CANN. The relative errors δ_E, δ_Q are calculated in the forms: $\delta_E = |E_c - E|/E$, $\delta_Q = |Q_c - Q|/Q$. In this case, corresponding triangles and squares with the identification error no exceeding of 20 %, are shown in Figs. 12.6 and 12.7.

The similar results for threshold error of 10 % are shown in Figs. 12.8 and 12.9.

Figure 12.10 demonstrates the results of identification in dependence on number of test data in the form of two curves, the solid curves correspond to the required data, and the dashed lines define the forecast data obtained by using CANN. The graph of training error (12.6) with 20,000 epochs is shown in Fig. 12.11.

12.5 Conclusion

In this paper, we developed a method for determination of material elastic and dissipative properties by using data for harmonic oscillations on the resonance frequency based on a combination of the finite element method and CANN. The results of the experiments showed that CANN with one of the following architectures:

2 (input neurons)—4 (hidden neurons)—1 (output neuron)
or
6 (input neurons)—6 (hidden neurons)—1 (output neuron)

gives the best result of identification. The time expenses caused by the CANN training were also estimated in this paper. The developed method and computer program could be used to determine the dissipative properties on different frequencies (not only on the first one), as well as in anisotropic elastic solids.

Acknowledgments This work was partially supported by the European Framework Program (FP-7) "INNOPIPES" (Marie Curie Actions, People), grant # 318874, and Russian Foundation for Basic Research (grants # 13-01-00196_a, # 13-01-00943_a).

References

1. H. Simon, *Neural Network a Comprehensive Foundation*, 2nd edn. (Prentice Hall, New Jersy, 1998)
2. A.A. Krasnoschekov, B.V. Sobol, A.N. Soloviev, A.V. Cherpakov, Russ. J. Nondestruct. Test. **6**, 67 (2011)
3. S.W. Liu, J.H. Huang, J.C. Sung, C.C. Lee, Comput. Methods Appl. Mech. Eng. **191**(4), 2831 (2002)
4. Hasan Temurtas, Feyzullah Temurtas, Expert Syst. Appl. **38**(4), 3446 (2011)
5. V. Khandetsky, I. Antonyuk, NDT&E Int. **35**, 483 (2002)
6. Y.G. Xu, G.R. Liu, Z.P. Wu, X.M. Huang, Int. J. Solids Struct. **38**(8), 5625 (2001)
7. P. Korczak, H. Dyja, E. Łabuda, J. Mater. Process. Technol. **80–81**(8), 481 (1998)
8. T. Mira, S. Zoran, L. Uros, Polymer **48**(8), 5340 (2007)
9. J. Ghaisari, H. Jannesari, M. Vatani, Adv. Eng. Softw. **45**(3), 91 (2012)

10. P. Iztok, T. Milan, K. Goran, Expert Syst. Appl. **39**(4), 5634 (2012)
11. Y. Sun, W. Zeng, Y. Han, X. Ma, Y. Zhao, P. Guo, G. Wang, Comput. Mater. Sci. **60**(7), 239 (2012)
12. T. Nitta, in *Proceedingsof the International Joint Conference on Neural Networks*, IEEE, p. 1649. Nagoya, 1993
13. T. Nitta, Neural Network **10**, 1391 (1997)
14. A. Hirose (ed.), *Complex-Valued Neural Networks: Theories and Applications*. The Series on Innovative Intelligence. (World Scientific, Singapore 2003)
15. C. Li, X. Liao, J. Yu, J. Comput. Syst. Sci. **67**, 623 (2003)
16. T. Nitta, *Complex-Valued Neural Networks: Utilizing High-Dimensional Parameters* (Information Science Reference, New York, 2009)
17. A.V. Belokon, A.V. Nasedkin, A.N. Soloviev, Appl. Math. Mech. **66**(3), 491 (2005)
18. W. Nowacki, *Theory of Elasticity* (MIR Publishers, Moscow, 1976). (in Russian)