Computer Assisted Mechanics and Engineering Sciences, 6: 449–467, 1999.

Beam yielding load identification by neural networks

Bartosz Miller, Grzegorz Piątkowski and Leonard Ziemiański
Department of Structural Mechanics, Rzeszów University of Technology
PL-35959 Rzeszów, ul. W. Pola 2, Poland

(Received March 8, 1999)

The paper presents the application of Artificial Neural Networks for the identification of the load causing a partial yielding in the cross-section of a simple supported beam. The identification of the load was based on a change of the dynamic parameters (eigenfrequencies) of the partially yielding structure. On this basis and using neural networks a tool for the location and evaluation of the load causing the deformation was built. The optimum network architecture, learning algorithm, number of epochs, and the minimum number of eigenfrequencies have been found. In order to come to the final conclusions, a wide variety of network architectures (from simple networks with four neurons in one hidden layer to complex networks consisting of two or three simple networks), learning algorithms and different numbers of learning epochs have been tested.

1. Introduction

Plastic deformation of a structure may arise as a result of external actions like exceeding the yielding load or as a result of section defects [2] (microdefects). The determination of the section yielding index enables the evaluation of the safety margin of the structure [6]. The yielding index can be calculated if the load effecting the structure is known. Thus load identification may provide information on the structure state. The identification can be performed by a modal analysis of the dynamic response of the object and by reckoning of the dynamic characteristics of the structure.

The term dynamic characteristics of a structure refers to the eigenfrequencies of the mathematical model which represent the modal parameters of the structure. By measuring the changes of the dynamic response of a structure we can estimate the structure state and identify the kind and value of the load. The state changes, e.g. the development of yielding zones, lead to changes of the structure stiffness and further to modal parameters changes. Information on the dynamic characteristics changes during the yielding process is required for the structure state evaluation.

Recently Artificial Neural Networks (ANN) have been applied in numerous fields of civil engineering, e.g. damage identification, structure parameters identification, nondestructive testing, structure state estimation, and others [12]. ANN proved to be a very useful and effective tool.

This paper presents the application of ANN in the load identification based on modal properties changes. The assessed problem is of the Load Simulation type [11] in which the response and features of a Mechanical Structure (MS) are given and the identification of the load is the expected result. A simple supported beam with loads exceeding the yielding load has been analysed. The load location and value have been calculated on basis of changes of the beam eigenfrequencies. ANNs recognising the load parameters have been constructed. The minimum number of eigenfrequencies for proper network operation and the optimum network architecture have been determined.

2. Description of the problem

Relative changes of the first ten eigenfrequencies of the numeric model have been considered in the solution of the problem performed with the finite element method (FEM) system ADINA [10].

The dynamic characteristic changes have been caused by exceeding the yielding load and further development of the yielding effect.

Backpropagation ANNs have been applied. The application of such networks has been decided basing on the publications concerning ANNs in civil engineering, especially in structure mechanics [11, 13].

The eigenfrequencies and eigenvectors of the structure have been calculated from the generalized eigenproblem [1]:

$$\mathbf{K\Phi} = \mathbf{M\Phi\Omega}^2, \tag{1}$$

where **K** and **M** are the stiffness and mass matrices. The columns of the **Φ** matrix are eigenvectors and **Ω** is a diagonal matrix with eigenvalues adequate to the vectors. The eigenvectors have the following properties:

$$\mathbf{K}\phi_i = \omega_i^2 \mathbf{M}\phi_i\,; \qquad i = 1, \ldots, n, \tag{2}$$

$$\phi_i^T \mathbf{M}\phi_j = \delta_{ij}\,, \tag{3}$$

$$\phi_i^T \mathbf{K}\phi_j = \omega_i^2 \delta_{ij}\,. \tag{4}$$

In this paper, the subspace iteration method has been applied in the solution of the eigenproblem described by Eq. (1) [1].

A simple supported beam (Fig. 1) has been analysed assuming on elastic-ideal plastic material.

The calculations have been performed for a series of load profiles causing exceeding of the yield point of the beam section. The structure has been modeled with eight-node plain stress finite elements. Basing on the previous research of the authors [8], the cross-section has been divided into six layers in the height direction. In the other direction the beam has been divided into 30 finite elements (Fig. 2).

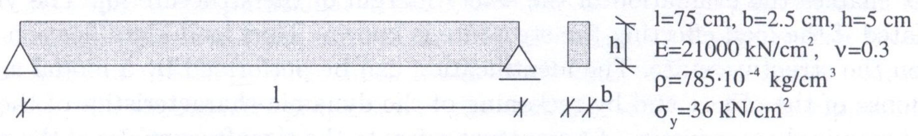

Fig. 1. Simple supported beam, dimensions and the data

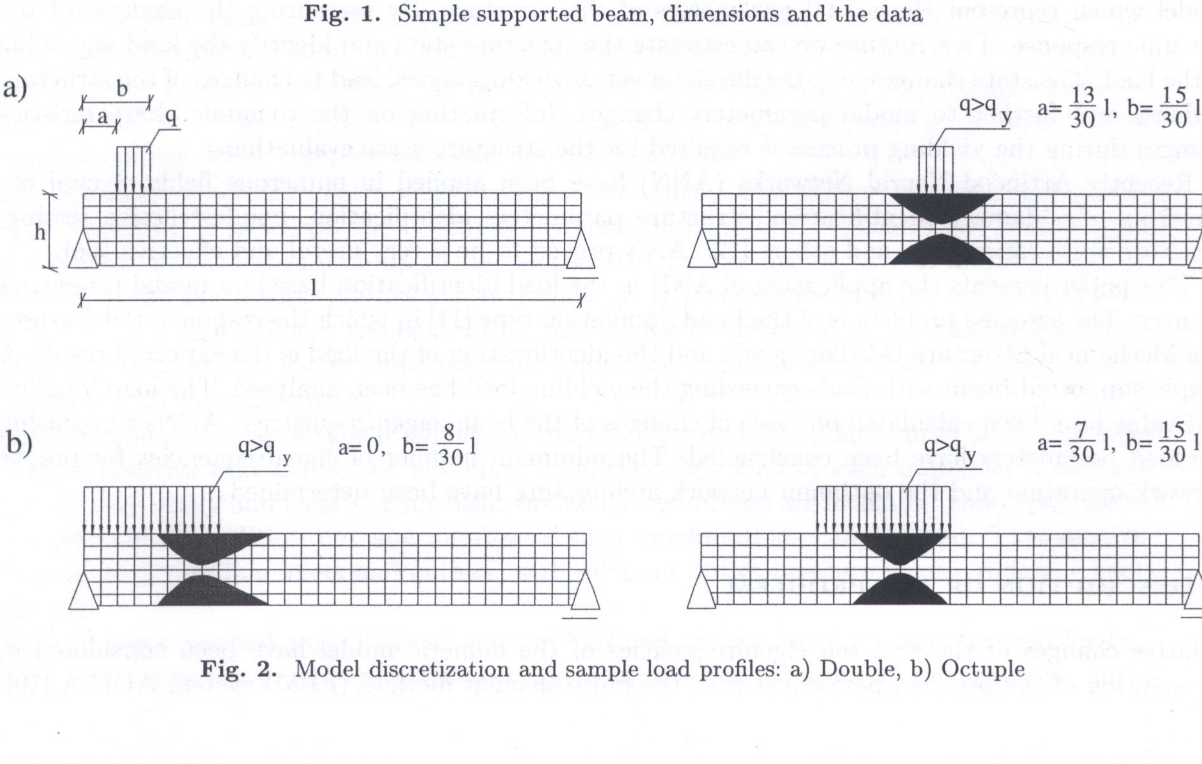

Fig. 2. Model discretization and sample load profiles: a) Double, b) Octuple

The load classification comprises of 5 profiles. The distinguishing parameter was the width of the load action zone: 1/30, 2/30, 4/30, 8/30 and 12/30 of the beam length. Due to the discretization method used the loads have been modeled as operating on 1, 2, 4, 8 or 12 contignous finite elements (Fig. 2). Further in this paper, the following terms are used:

- Single Load – load operating on 1/30 of the beam,
- Double Load – load operating on 2/30 of the beam,
- Quadruple Load – load operating on 4/30 of the beam,
- Octuple Load – load operating on 8/30 of the beam,
- Twelvefold Load – load operating on 12/30 of the beam.

The load location within a profile may vary; for example 14 locations of a Double Load and 8 locations of a Octuple Load have been considered. This approach makes it possible to calculate 53 different load profiles.

Two load profiles are shown in Fig. 2 (Double and Octuple Loads). Figure 2 shows also examples of yielding zones caused by the yielding load exceeding.

The beam load has been increased incrementally. The most interesting phase of the load application is this one which causes a plastic strain and leads to changes of the dynamic characteristics. The quantity of the load has been increased till a nearly complete yielding of the material in the height of the cross-section has been achieved. The quantity of the load has been increased so that after the first increment a maximum elastic stress has been reached and the next increments have been selected to obtain the same number of them in the plastic range for all the profiles.

The calculations have been performed in two stages. At first, the beam stress distribution has been qualified and for a certain state of the yielding effect, the dynamic characteristics have been reckoned. The results of these computations are sets of data for each of the load profiles that have been used for the ANN learning and testing. The number of patterns is as follows:

- Single 56 learning and 10 testing patterns,
- Double 75 learning and 11 testing patterns,
- Quadruple 59 learning and 9 testing patterns,
- Octuple 44 learning and 20 testing patterns,
- Twelvefold 29 learning and 13 testing patterns.

Together: 263 learning and 63 testing patterns.

The division of the patterns into the learning and testing ones is related to the time of their creation. The first group of the patterns has been qualified as the learning ones, then the testing patterns have been calculated to cover the whole area defined by the learning patterns.

3. Neural networks for separate load profiles

3.1. Assumptions for the ANN formulation

In the construction of ANNs for solving the problem, the following assumptions have been made:

1. In the first stage, five separate networks have been considered (one for each of the load profiles). Two learning algorithms have been applied: the Backpropagation with Momentum term (BPM) [9] and the Resilient Backpropagation (Rprop) [7]. Each network had ten ins and two or three linear outs (Section 3.2).

2. In the second stage (also five networks, one for each of load profiles), networks with ten ins and sixteen binary outs have been applied. The Rprop algorithm has been used for learning (Section 3.3).
3. In the third stage (complex networks, the same for all load profiles), networks with different number of inputs and outputs and the Rprop algorithm have been applied (Section 4.1).
4. In the fourth stage, simple networks with the Rprop algorithm have been applied (Section 4.2).

The input of each network has been fed with the relative changes of the beam eigenfrequencies after a partial yielding:

$$\Delta\omega_i = 0.9\frac{\omega_i^e - \omega_i^y}{\omega_i^e}; \qquad i = 1, \dots, n \tag{5}$$

where:

ω_i^e – i-indexed beam eigenfrequency before yielding (elastic range),

ω_i^y – i-indexed beam eigenfrequency after yielding (beyond the elastic range).

For all networks the full neighbouring layer neuron connection has been used. For all the neural network operations the SNNS [9] system has been used.

3.2. Neural networks with linear outputs

3.2.1. Inputs and outputs

The BPM and Rprop learning algorithms have been used in the first stage. The networks had ten ins and two or three outs. The output data were the ordinates of the starting points a and in some cases the ending b points of the load operating on the beam (given nondimensionally within the range 0–1) and the load quantity q normalized in the 0–1 range (0 – maximum value of load located next to the support and causing complete yielding in the cross-section), see Fig. 2.

Because in this stage separate networks have been built for each of the load profiles and the load operating width was known, two-output networks have been constructed without specifying the end point of the load operation.

3.2.2. Learning parameters and the network architecture

The network building was started with setting the learning parameters. Two learning algorithms have been applied: BPM and Rprop. The BPM has four parameters:

- η: learning parameter,
- μ: Momentum term,
- c: flat spot elimination value,
- $d_{\max}$: the maximum difference between the input learning value and the output value.

Nine different parameter sets have been tested for four different architectures with one, two or three hidden layers. For the smallest mean square error (MSE) calculated from Eq. (6) and the best learning process stability the following parameters have been chosen: $\eta = 0.3$, $\mu = 0.005$, $c = 0.001$ and $d_{\max} = 0.0$. These have been in most cases the optimum parameters although in some cases $\eta = 0.2$, $\mu = 0.2$, $c = 0.001$ and $d_{\max} = 0.0$ produced better results. The same parameters have been chosen for all the load profiles.

The Rprop algorithm has three parameters:

- Δ_0: initial update value,
- Δ_{max}: limit for the maximum step size,
- α: weight-decay exponent.

The following set of parameters has been selected from among the tested ones: $\Delta_0 = 30.0$, $\Delta_{max} = 0.5$ and $\alpha = 0.0$. In some cases the following values seemed to be better: $\Delta_0 = 30.0$, $\Delta_{max} = 2.0$ and $\alpha = 0.0$.

The mean square error has been used as the measure of the precision of the learning and testing processes [7]:

$$MSE = \frac{\sum\limits_{p \in \text{patterns}} \; \sum\limits_{j \in \text{outputs}} (t_{pj} - o_{pj})^2}{n} \tag{6}$$

where:

t_{pj} – the output value expected of the j neuron for the p pattern,
o_{pj} – the output value,
n – number of patterns.

After the learning algorithm parameters were selected, the next step was the selection of the architecture. The following network architectures have been tested for each of the load profiles:

- two one-hidden layer networks: 10-10-3 and 10-14-3,
- three two-hidden layer networks: 10-10-6-3, 10-10-10-3 and 10-14-10-3,
- one three-hidden layer network: 10-14-10-6-3.

The most precise and stable network is the 10-14-10-3 one. In some cases the 10-14-10-6-3 (three hidden layers) network produces better results but is less stable. The one-hidden layer networks tested for the Single, Double and Quadruple loads produced results of lower precision than the two hidden layer networks.

Figures 3 and 4 show the relation of the learning error to the number of epochs for Single and Octuple Loads and both the learning algorithms. The error is much smaller for the Rprop algorithm. The latter algorithm requires also less learning epochs. Further the learning process is interrupted after 10000 epochs.

3.2.3. Three-output network results

The 10-14-10-3 network has been trained to recognize loads operating on the structure and causing a yielding effect in the cross-section. The network has been tested with patterns which had not been used in the learning process. Each load profile has been tested separately. The calculations have been performed using both BPM and Rprop algorithms.

The results for the two chosen profiles are shown in Figs. 5 and 6. The horizontal axis represents expected values calculated with the FEM, while the vertical axis represents the results from the network. The $y = x$ line shows the desired location of the points, i.e. location without error. The distance of a point from the $y = x$ line represents the precision of the parameter (location or value) determination. Figs. 5 and 6 show that the selected network identifies properly the load operating on the beam on basis of the dynamic characteristics. In the case of the BPM algorithm, the error of the load edge point location remains within ±5% of the beam length limits which is equal to a one finite element size error. The Rprop networks are even more precise.

The 5% error limit is exceeded (BPM networks) for a Single load for learning patterns referring to the first phase of the yielding. Relative dynamic characteristic changes are small for these states. Therefore, the network was unable to interpret them correctly [4].

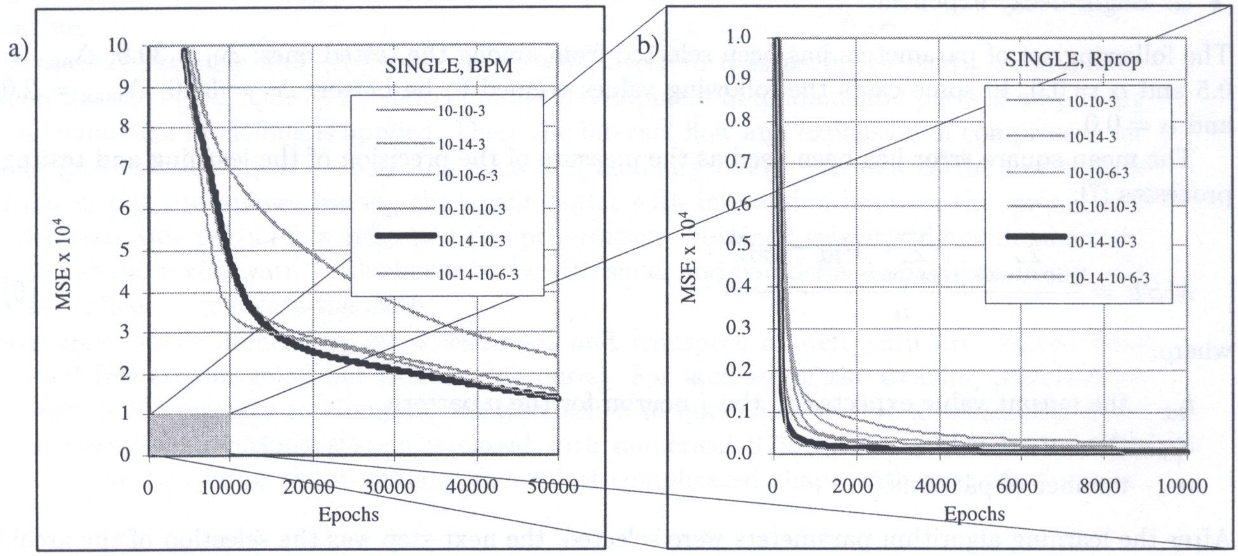

Fig. 3. Network learning for the Single Load: a) BPM algorithm b) Rprop algorithm

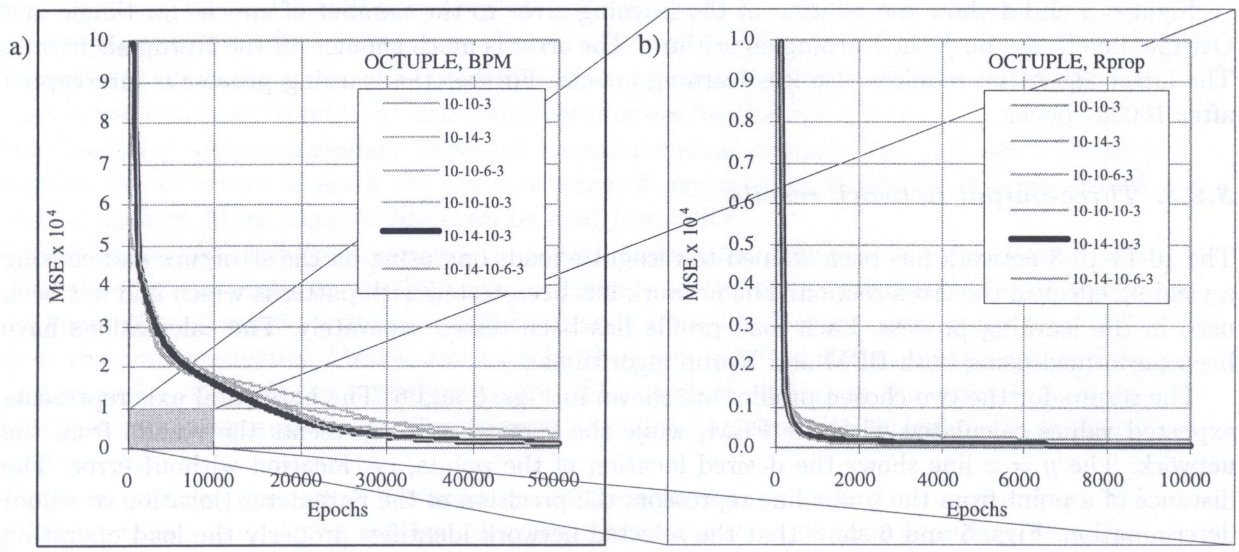

Fig. 4. Network learning for the Octuple Load: a) BPM algorithm b) Rprop algorithm

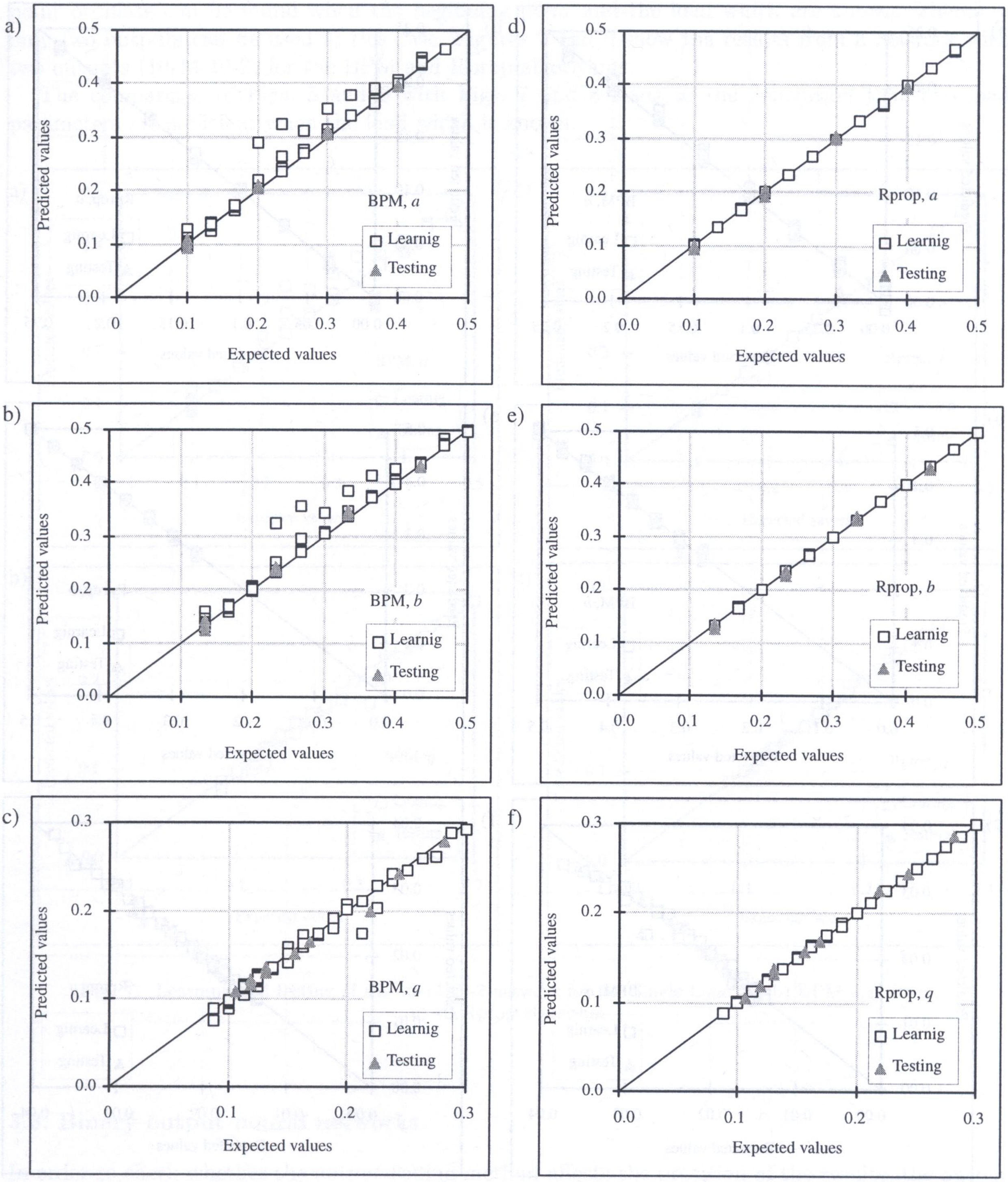

Fig. 5. Learning and testing of the 10-14-10-3 network for the Single Load; a)–c) BPM algorithm, d)–f) Rprop algorithm

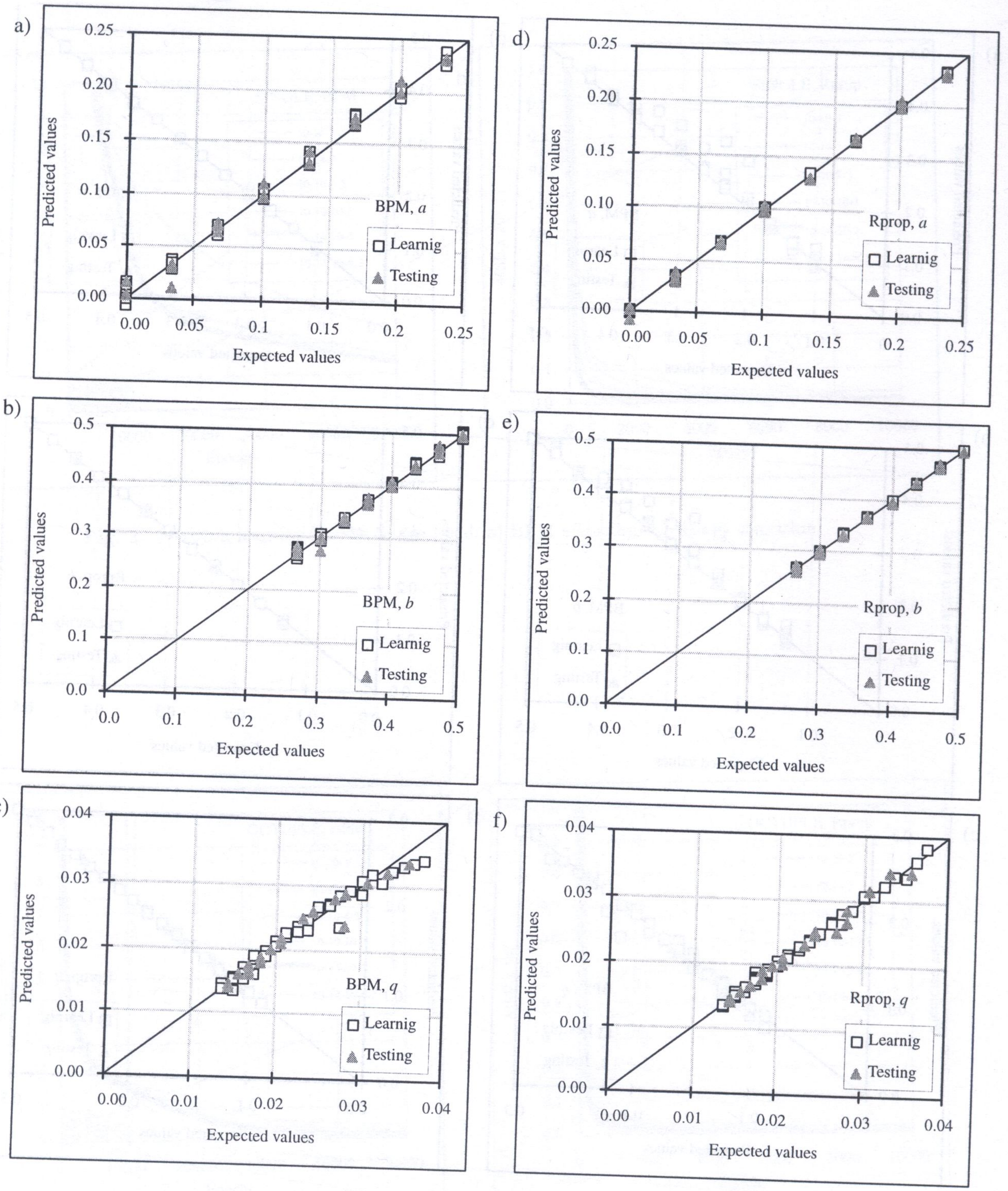

Fig. 6. Learning and testing of the 10-14-10-3 network for the Octuple Load; a)–c) BPM algorithm, d)–f) Rprop algorithm

3.2.4. Two-output network results

As stated previously, at this stage separate networks have been used for each load profile. The load is described by two parameters: the ordinate of the load beginning point and its value. The end point ordinate can be found when the beginning point and the load width are known. Therefore only two outputs can be used in this case. Figures 7 and 8 show the results from a network with two outputs (10-14-10-2) for the BPM and Rprop algorithms.

The comparison of Figs. 5 and 6 with Figs. 7 and 8 leads to the conclusion that two load parameters are sufficient when the load width is known.

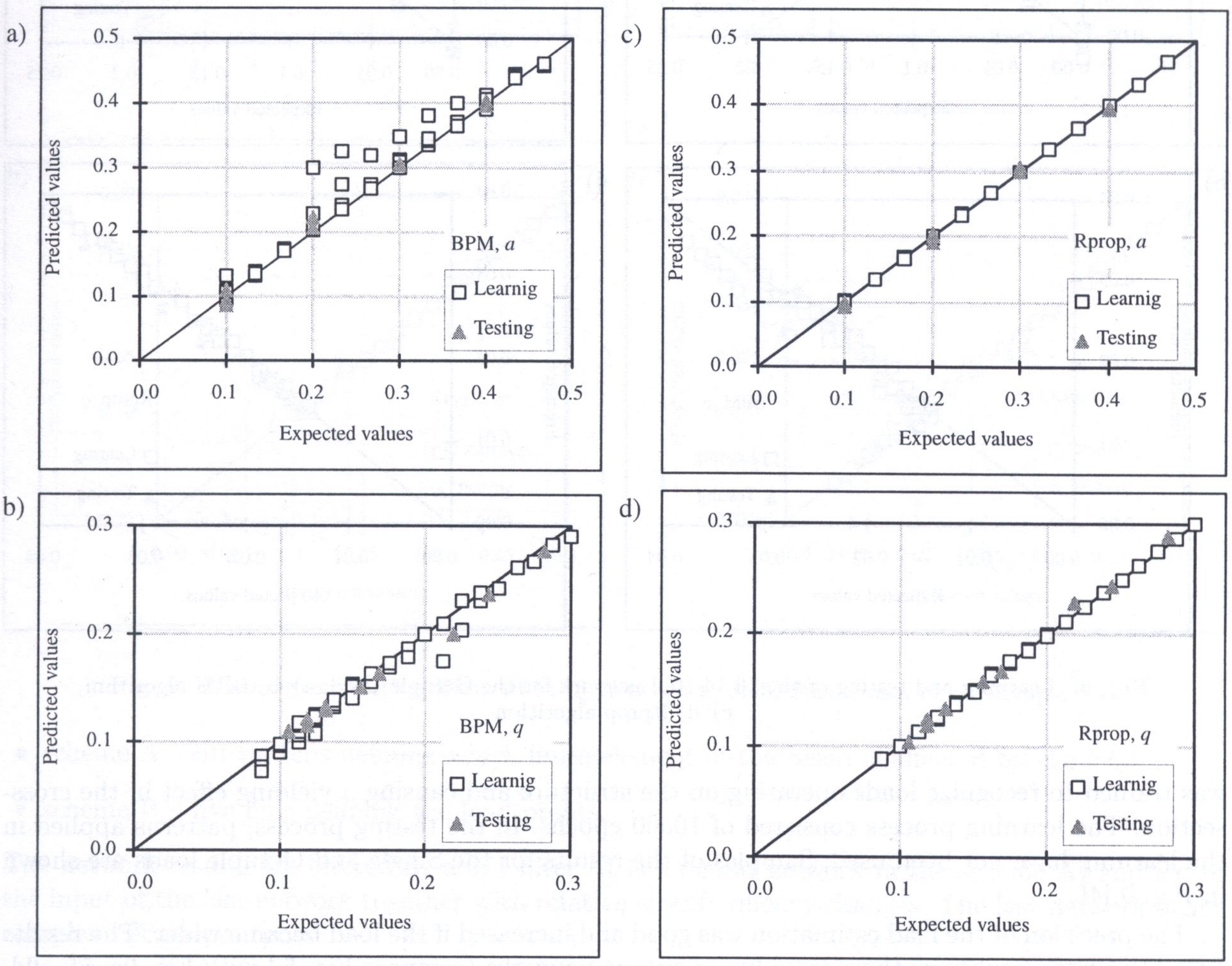

Fig. 7. Learning and testing of the 10-14-10-2 network for the Single Load; a)–b) BPM algorithm, c)–d) Rprop algorithm

3.3. Binary output neural networks

In order to check whether the output coding method affects the precision of the results, the output parameters have been changed [3]. The inputs were fed with the relative changes of the beam eigenfrequencies calculated from Eq. (5), just like in the first stage. The network output was defined as follows: fifteen binary numbers indicating if the given finite element is borne with the load (1 – yes, 0 – no) and the normalized quantity (q) of the load ranging from 0–1. After two, three and four hidden layer networks were tested, it was decided to apply one network 10-14-20-16 (10 ins, 14 neurons in the first hidden layer, 20 in the second and 16 outs) for all the loads. The network

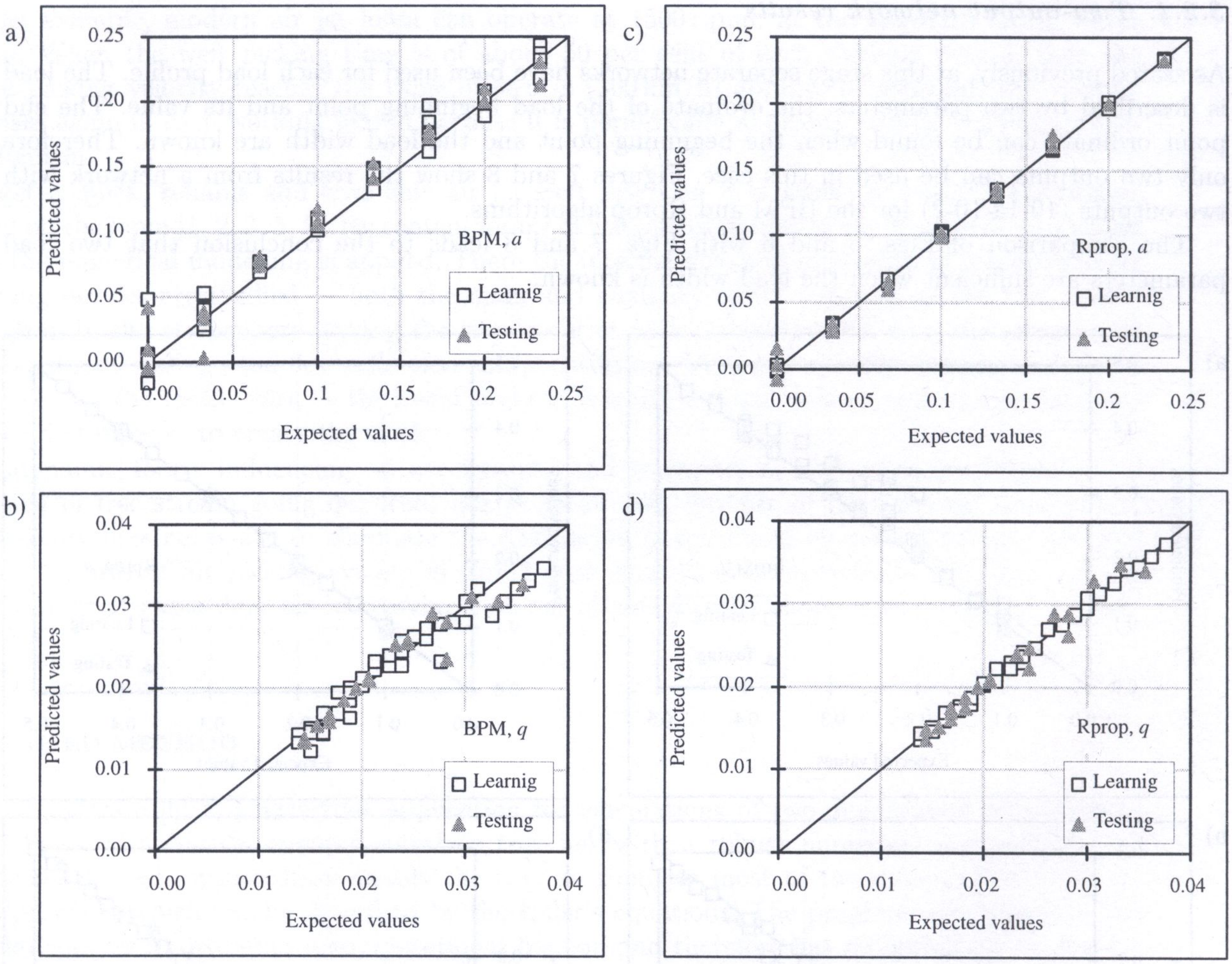

Fig. 8. Learning and testing of the 10-14-10-2 network for the Octuple Load; a)–b) BPM algorithm, c)–d) Rprop algorithm

was trained to recognize loads operating on the structure and causing a yielding effect in the cross-section. The learning process consisted of 10000 epochs. In the testing process, patterns applied in the learning have not been used. Samples of the results for the Single and Octuple loads are shown in Fig. 9 [4].

The precision of the load estimation was good and increased if the load became wider. The results are of the same quality as those from linear output networks (compare Fig. 5d with Fig. 9a, 5f – 9d, 6a – 9c, 6f – 9d).

4. Neural networks for the all load profiles analysis

4.1. Complex networks

At this stage, a universal load was considered without division of the load type. The results from individual networks trained for the load identification without the load division have not been satisfactory; therefore several complex networks, consisting of simple networks have been built. The complex networks are shown in Fig. 10. In both cases three-stage architectures are applied. The front network has ten relative changes of the eigenfrequencies on the inputs. The outputs from the network are:

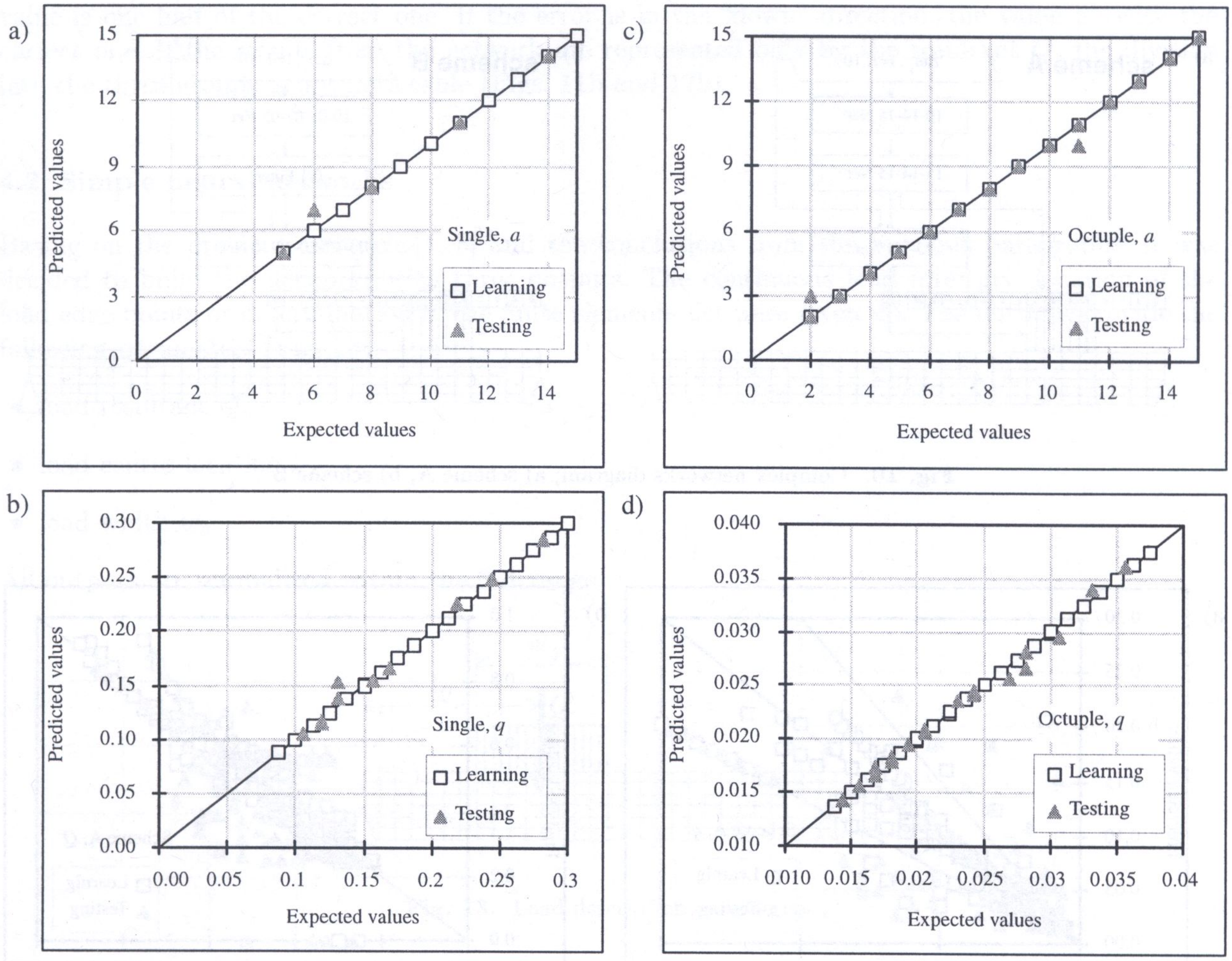

Fig. 9. Learning and testing of the 10-14-20-16 network; a)–b) Single load, c)–d) Octuple load

- scheme A – fifteen bins defining which finite element of the beam is affected by the load,
- scheme B – five bins defining the load profile.

The network results are filtered by a 0/1 filter (it is a neural network in the scheme A) and fed to the input of the last network together with relative eigenfrequency changes. The last network of the complex network outputs:

- scheme A – the q load intensity operating on the beam,
- scheme B – fifteen bins identifying the load location and one number identifying the intensity q.

Other complex networks have been tested as well, but none of them produced results better than those shown in Fig. 10.

The results are shown in Figs. 11 and 12. The charts show the q value (intensity) obtained directly from the network and the Q value of the resultant load calculated as follows: $Q = q(b - a)$ (see Fig. 2a).

The points in Figs. 11a and 12a are concentrated in the vicinity of three lines. The points placed close the middle line ($y = x$) represent a proper identification done by the network, while the points placed nearly the boundary lines represent identification cases in which the load value is twice too high or too low. This is due to the fact that the neighbouring load profiles are characterized by a half or a double width of the "lower" or "higher" profile (except the Twelvefold ones). If the load profile identification is wrong in the "up" direction (e.g. Double instead of Single), the resulting

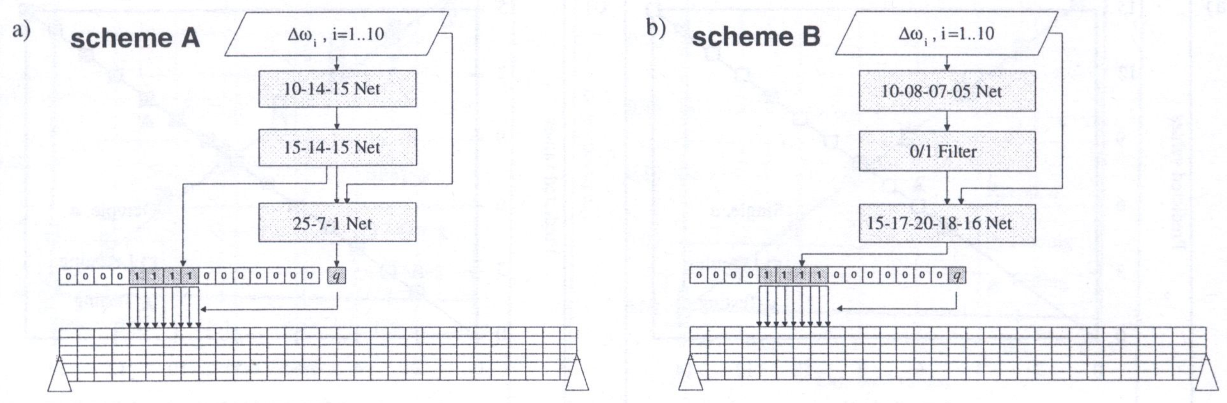

Fig. 10. Complex networks diagram; a) scheme A, b) scheme B

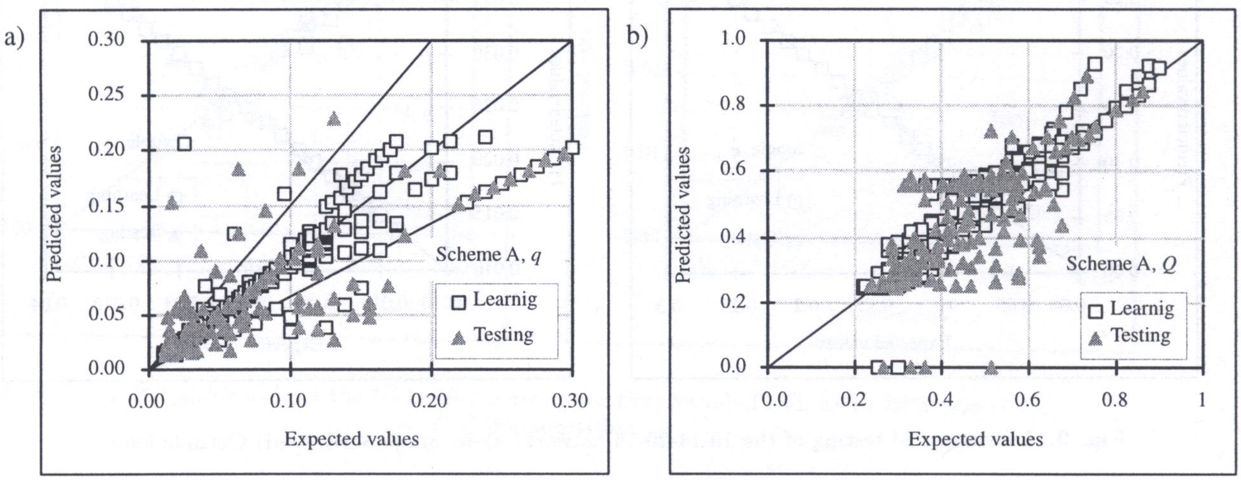

Fig. 11. Results from the complex network (scheme A – Fig. 10a); a) intensity q, b) resultant Q

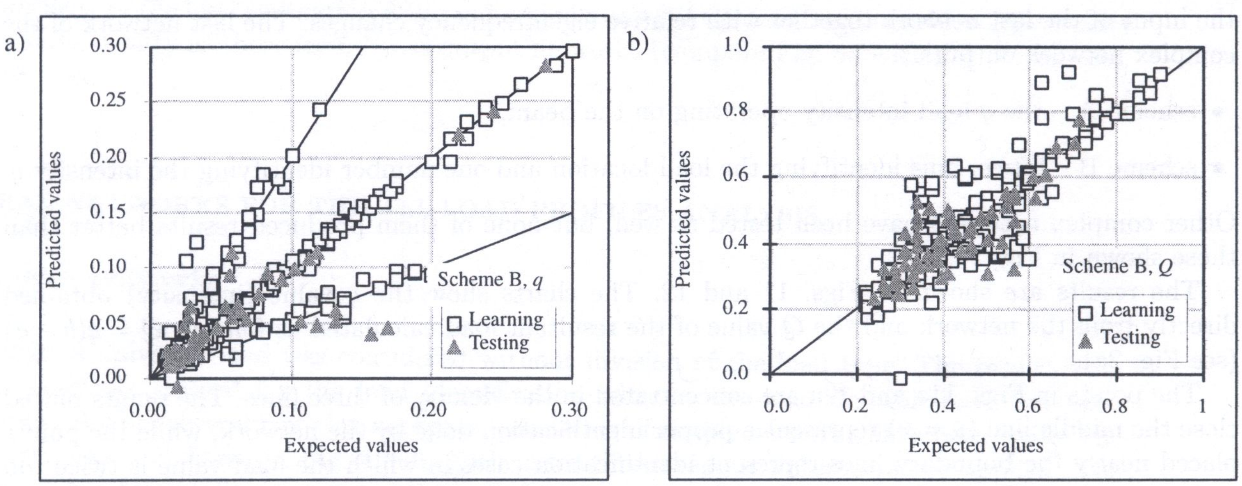

Fig. 12. Results from the complex network (scheme B – Fig. 10b); a) intensity q, b) resultant Q

value is one half of the correct one. If the error is in the "down" direction, the value is twice the correct one. If the results from the network are represented only by the resultant Q, the division into the three groups is not noticeable (Figs. 11b and 12b).

4.2. Simple neural networks

Basing on the previous research [4, 5] and the conclusions from the previous paragraphs, it was decided to build the networks with three outputs. The continuous load intensity, location of the load edge points or determination of the finite elements list were given up. The outputs provide the following parameters (Fig. 13):

- load resultant Q,
- load centre location l_Q,
- load width w_q.

All outputs are normalized within the 0–1 range.

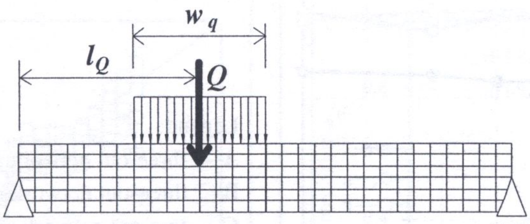

Fig. 13. Load description parameters

The first calculations proved that the change of the output definition allows to simplify the network structure. In order to select the best architecture for each kind of the problem, 78 different networks have been built. Each of them has 3, 4, 6, 7, 8 or 10 ins, one or two hidden layers with 4 to 16 neurons and 3 outs. The inputs are relative eigenfrequencies changes. In order to find the minimum number of frequencies necessary for a correct load identification, the network has been fed with 3, 4, 6, 7, 8 or 10 relative changes of the first eigenfrequencies. The changes of eigenfrequencies were scaled in the [0.15 ; 0.85] or [0 ; 1] range for all the frequencies together or in the [0;1] range for separate frequencies. All networks have been trained to recognize the load operating on the beam. 666 patterns (the old set of patterns plus patterns not fitting in the previously defined load profiles — load operating on three, six or ten finite elements) were used. 46 of them were chosen at random as the testing patterns, the remaning ones were the learning set. Basing on our previous experiences, the Rprop algorithm has been used and the learning process consisted of 5000 epochs. A measure of network precision is the linear correlation coefficient of two n-element sets X and Y (learning and testing sets) defined by the formula:

$$r = \frac{\sum(x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum(x_i - \overline{x})^2 \sum(y_i - \overline{y})^2}} \tag{7}$$

where:

x_i, y_i – elements of X and Y sets,
$\overline{x}$, $\overline{y}$ – arithmetic mean of X and Y sets.

Figure 14 shows the coefficient of the linear correlation r of the expected value with the network value in relation to the number of inputs.

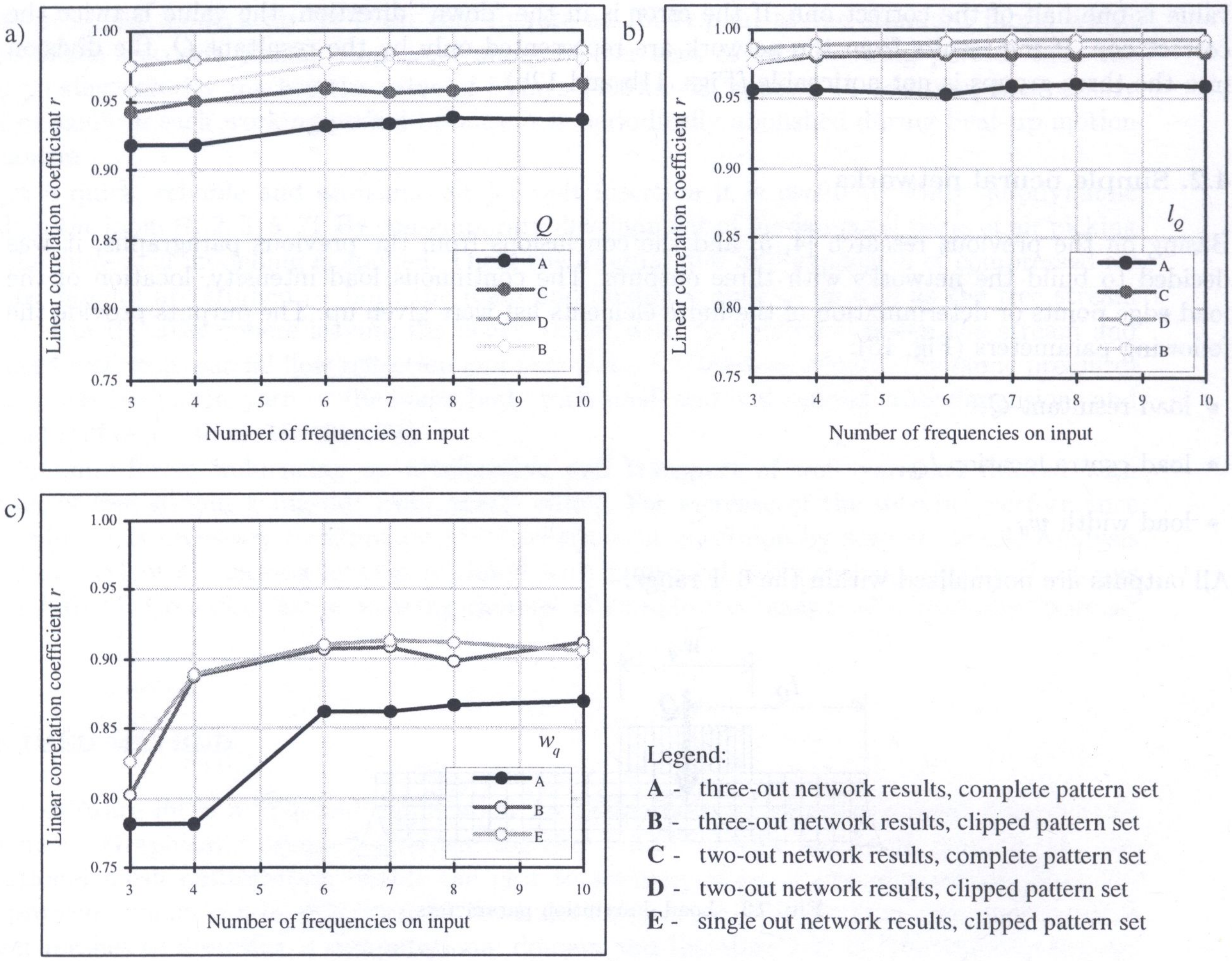

Fig. 14. Linear correlation coefficient value in relation to the number of network input frequencies; a) load resultant Q, b) load centre location l_Q, c) load width w_q

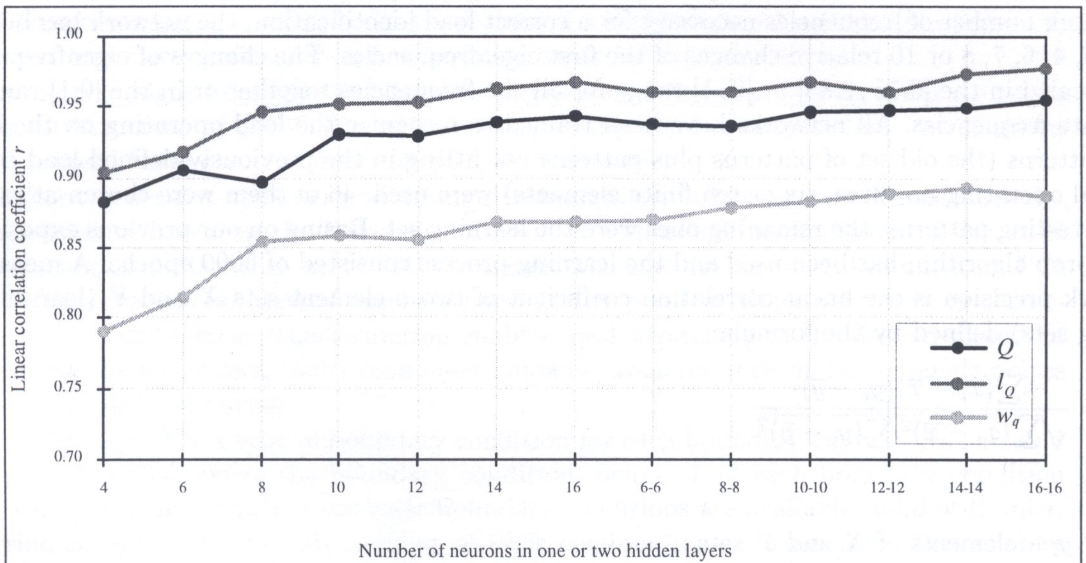

Fig. 15. Linear correlation coefficient value r in relation to the number of hidden layer neurons of the six-input network

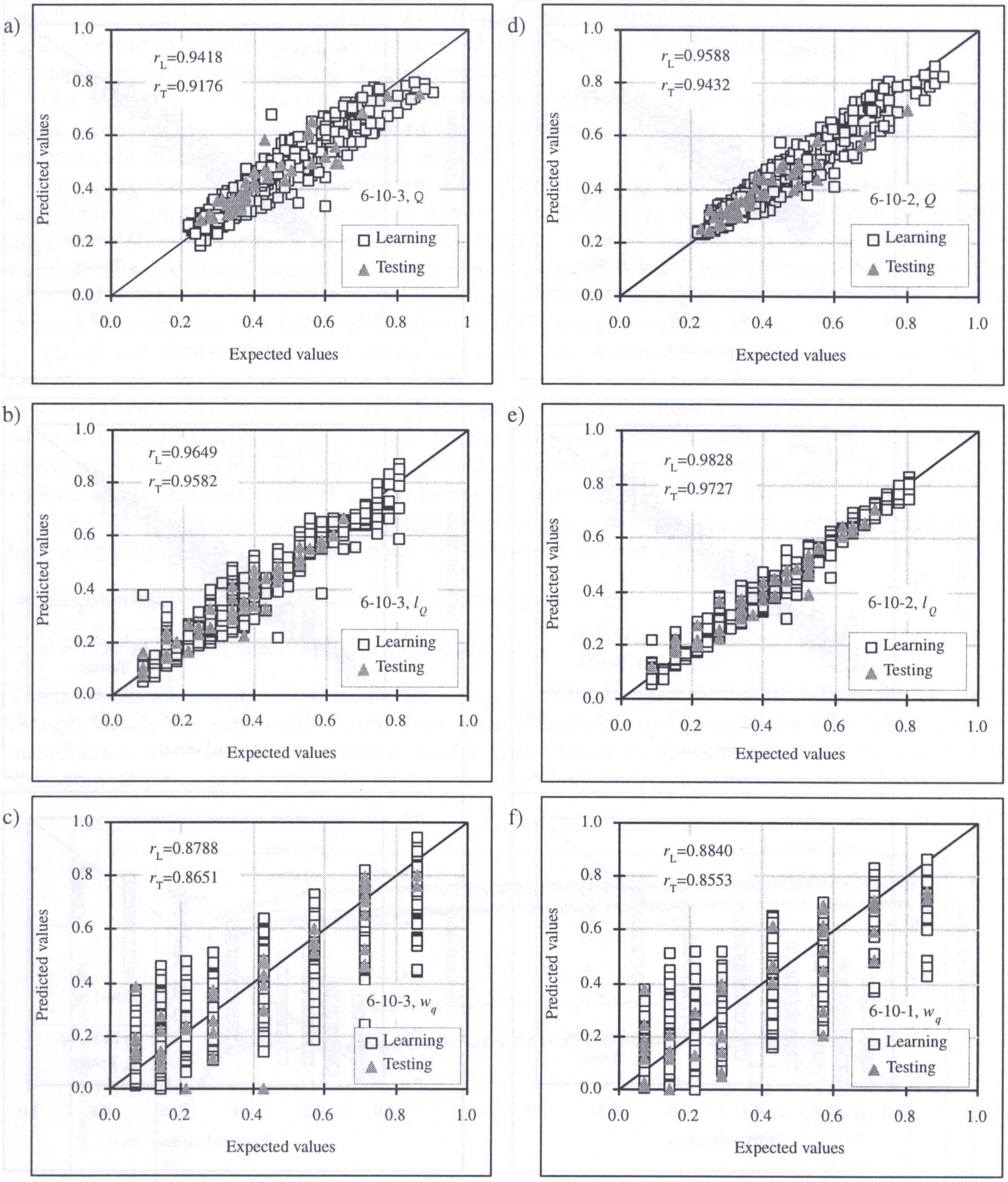

Fig. 16. One-, two- and three-output network results (complete pattern set); a)–c) three-out network, d)–e) two-out network, f) one-out network

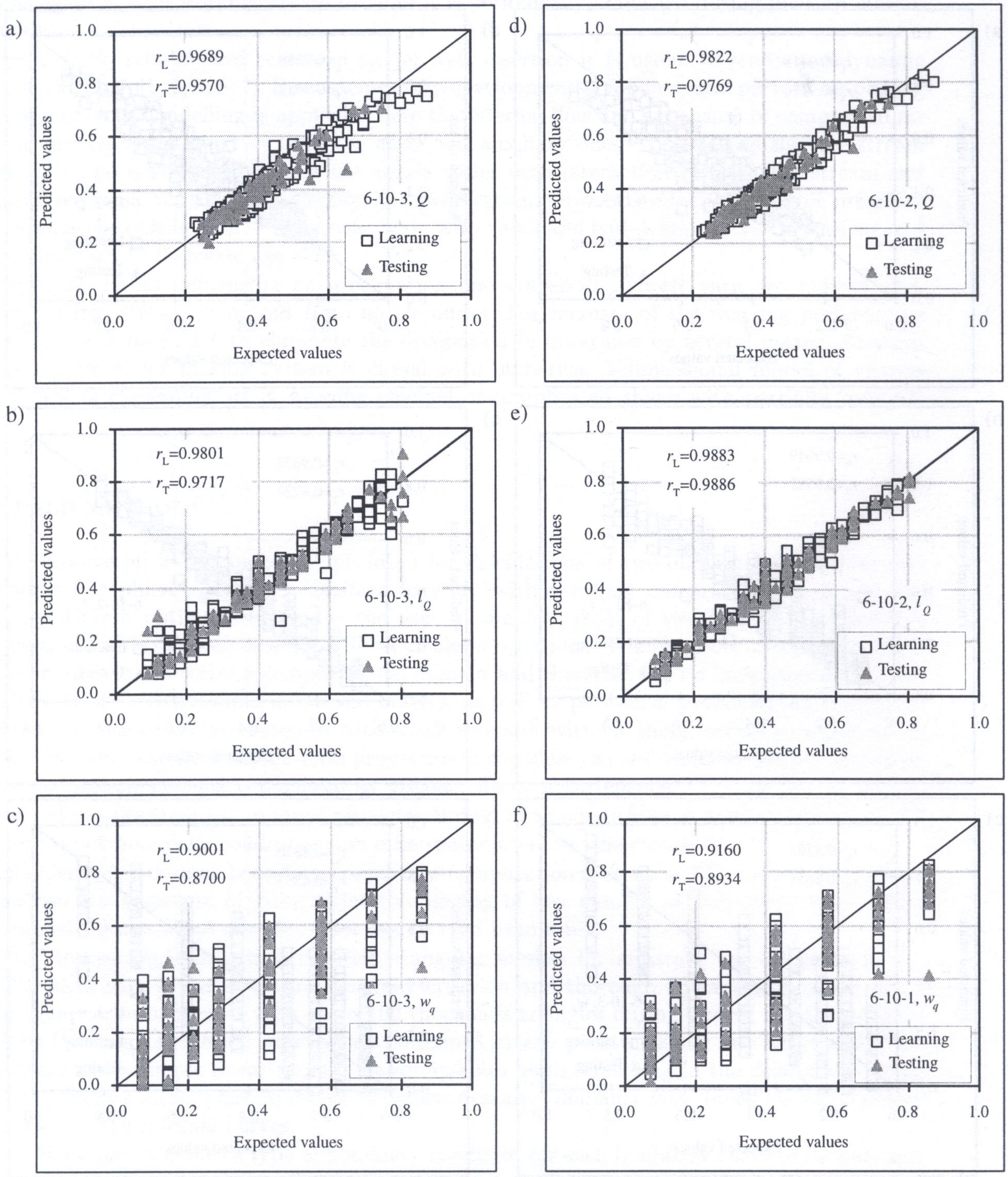

Fig. 17. One-, two- and three-output network results (clipped pattern set); a)–c) three-out network, d)–e) two-out network, f) one-out network

The l_Q load centre location and the Q load resultant are identified correctly; the precision of the network with three frequencies on the input, described by the r coefficient, does not different significantly from the precision of the network with ten frequencies on the input. The load width determination is less precise than load resultant or the centre location. The precision resulting is significantly worse in case of networks with four or less eigenfrequencies on the input.

The performed calculations have not resulted in a conclusion about the best input scaling method; all the methods produced results of similar precision.

The next step has been the network architecture selection. Figure 15 shows the value of the linear correlation coefficient in relation to the network complexity (only six-input network — other networks are characterized by the same relation).

Increasing the number of neurons from 4 to 10 in one hidden layer networks brings a significant improvement. A further increase of the number of neurons (up to 32 in two hidden layers) still improves the network quality though the improvement is not so high. It may be concluded that the simplest network identifying the load with a good precision should have 10 neurons in one hidden layer and relative changes of five or six eigenfrequencies on the input. The results from the 6-10-3 network (the simplest one fulfilling the above requirements) are shown in Fig. 16.

The Q load resultant value and the l_Q load centre are identified by the network with satisfactory precision, while the load width bears a significant error. To check the influence of the worse output, a network with only two outputs (the load resultant and the load centre) has been tested. The network proved to produce much better results (compare series A with C on Fig. 14 and series a with d, b with e on Fig. 16). No improvement was observed in the networks with only one output — load width (compare series B with E in Fig. 14 and series c with f in Fig. 16).

From Fig. 16 one can learn that some of the parameter calculations (the resultant or load width) have a significant error. This takes place in the following situations:

- the load lightly exceeds the maximum elastic stress,
- the load is located close to the beam support.

The networks are not able to estimate the output parameters correctly due to the relative frequency changes nearly the zero value or the data with a significant numerical error (the beam support introduces a distortion in the numeric model of the beam in the assumed finite element mesh). Therefore all patterns for which the relative changes of all the ten eigenfrequencies are less than 2%,

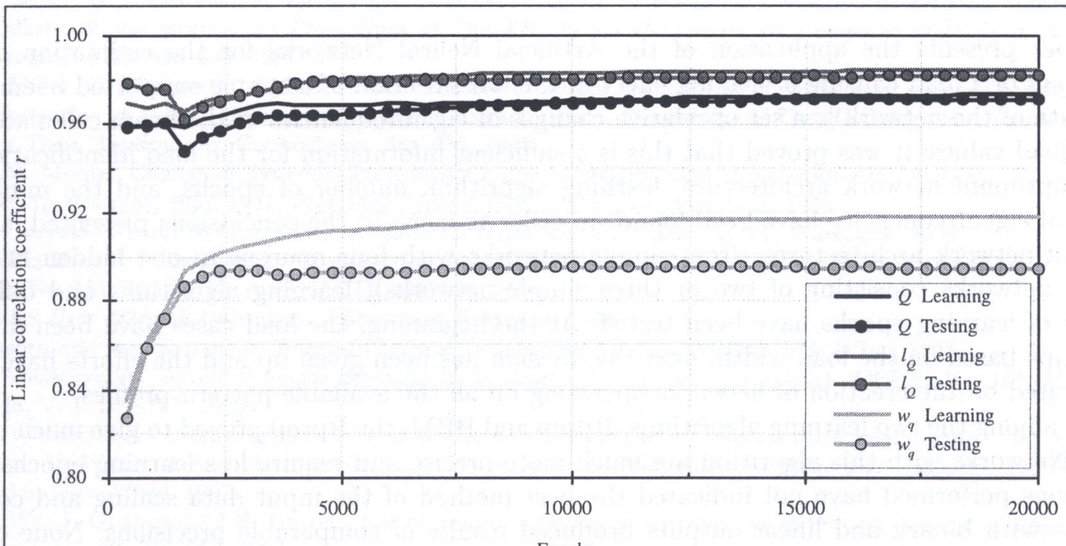

Fig. 18. The r linear correlation coefficient during 6-10-3 network learning

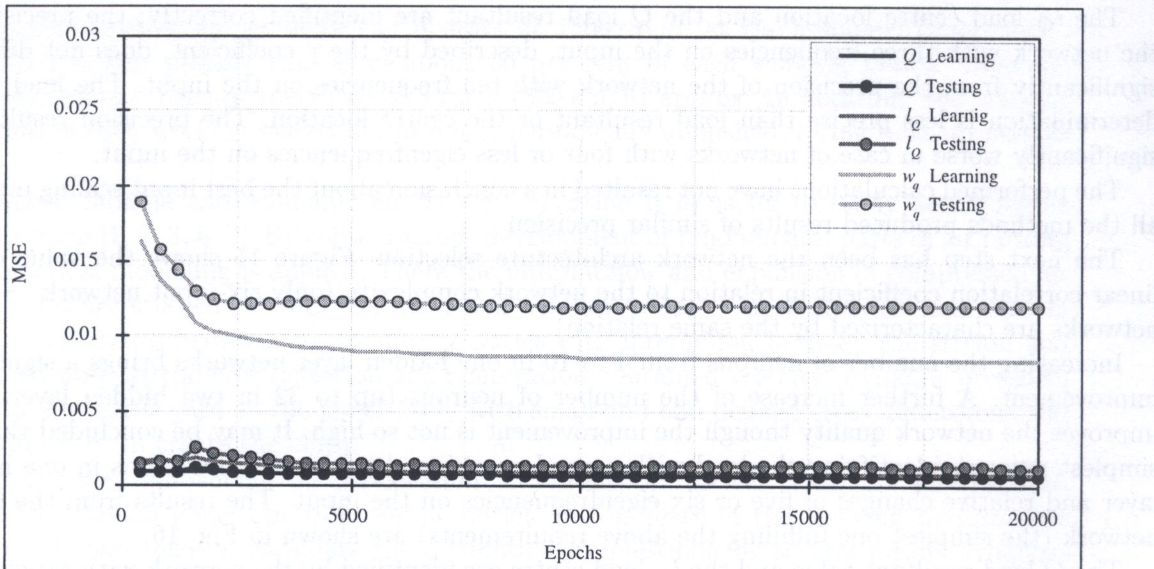

Fig. 19. MSE during 6-10-3 network learning

and patterns with load operating on the first or second finite element neighbouring the support, have been eliminated. The total of 136 out of 666 patterns have been given up. The results from networks trained on the clipped pattern sets are shown in Fig. 14 (series B, D and E). Figures 16 and 17 and also Fig. 14 (compare series A with D and C with D) show the improvement in comparison to the networks trained on the full 666 pattern set.

The last task is to set the optimum number of learning epochs. Figs. 18 and 19 show the relation of the r coefficient and the MSE (calculated individually for each of the outputs) to the number of learning epochs (for the clipped pattern set).

An extension of the number of epochs above 5000 hardly improves the network precision, and in case of the load resultant and load centre (first two outputs) identification 3000 epochs seems to be sufficient.

5. Final remarks

The paper presents the application of the Artificial Neural Networks for the estimation of the parameters of a load causing a yielding effect in the cross-section of a simple supported beam. The input data of the network is a set of relative changes of eigenfrequencies of the beam calculated for various load values; it was proved that this is a sufficient information for the load identification.

The optimum network architecture, learning algorithm, number of epochs, and the minimum number of eigenfrequencies have been found. In order to come to the conclusions presented, a wide variety of network architectures (from simple networks with four neurons in one hidden layer to complex networks consisting of two or three simple networks), learning algorithms and different numbers of learning epochs have been tested. At the beginning, the load cases have been divided into groups based on the load width, next the division has been given up and the efforts have been concentrated on the creation of networks operating on all the available pattern profiles.

From among the two learning algorithms: Rprop and BPM, the Rprop proved to give much better results. Networks with this algorithm are much more precise and require less learning epochs. The calculations performed have not indicated the best method of the input data scaling and coding. Networks with binary and linear outputs produced results of comparable precisions. None of the input scaling methods has produced better results than the other two ones. It is unnecessary to input more than five or six first eigenfrequencies changes. The increase of the number of the frequencies does not improve the precision of the calculation and lengthens the calculation time.

First four eigenfrequencies make it possible to identify correctly the load resultant and the location, while the precision of the width identification bears a much larger error. Less than four eigenfrequencies on the input lead to a situation in that the network is unable to recognize any load parameter. From among the three parameters used for the load description, two (location and value resultant) are defined correctly by the network. The third parameter (load width) has the biggest error — the input eigenfrequencies changes do not carry the information sufficient for the width identification with expected precision.

In the case of parameters defined in the last stage of the work (load location, resultant and operational width), even networks with one hidden layer with ten neurons produce satisfactory results. Further network complexity increase does not improve the quality of the results. The precision of the load location and the resultant identification may be improved by the elimination of the third parameter. The same approach applied to the width identification (a network identifying only this parameter with the resultant and location eliminated) does not improve the precision above that precision of the parallel identification of all three parameters.

Because the load width identification is very unprecise, the efforts to build a complex network identifying the load width in its first stage are ineffective. The error of the width identification is transferred to the further identification process. After the load parameters definition change was introduced the width identification did not influence the other parameters which results in a significant improvement of the precision in the third stage of the research. A further precision improvement may be obtained not by increasing the number of input eigenfrequencies but by providing supplementary input data, e.g. selected coordinates associated with the eigenvector frequencies.

ACKNOWLEDGMENTS

Financial support by the Polish Committee for Scientific Research Grant No. 7T07A00313 *"The Use of Neural Networks in Dynamics of Structures"* is gratefully acknowledged.

REFERENCES

[1] K.-J. Bathe. *Finite Element Procedures.* Prentice Hall, Englewood Cliffs, New Jersey, 1996.

[2] W.F. Chen, D.J. Han. *Plasticity for Structuaral Engineers.* Springer-Verlag, New York/Berlin/Heidelberg, 1988.

[3] T. Masters. *Practical Neural Networks Recipes in C++.* Academic Press Inc, 1993.

[4] B. Miller, G. Piątkowski, L. Ziemiański. The use of neural networks for prediction of dynamic characteristics of elastic-plastic structures. *Proceedings of The XIII Polish Conference on Computer Methods in Mechanics*, Poznań, V. **III**: 893–900, 1997.

[5] B. Miller, G. Piątkowski, L. Ziemiański. The use of neural networks for identification of loads causing plastic deformation of a structure based on dynamic characteristics. *Proceedings of The III Conference Neural Networks and Their Applications.* Częstochowa, 381–386, 1997.

[6] M.M. Moran. Change of dynamic characteristics due to plastification. *Computational Plasticity. Fundamentals and Applications*, 1967–1997, Pineridge Press–CIMNE, Swansea–Barcelona, 1995.

[7] S. Osowski. *Neural Networks in Algorythmical Depiction* (in Polish). WNT, Warszawa, 1996.

[8] G. Piątkowski, L. Ziemiański. Cross-section yielding influence on dynamic parameters of a beam (in Polish). ZN PRz No. 151, Mechanika z. 48, V. **III**: 137–144, 1996.

[9] *SNNS User Manual.* University of Stuttgart, Stuttgart, 1995.

[10] *Theory and modeling guide*, Adina-In Manual, Adina-Plot Manual. ADINA R & D Inc, 1992

[11] Z. Waszczyszyn. Artificial neural networks in dynamics of structures (in Polish). ZN PRz No. 151, Mechanika z. 48, V. **I**: 107–127, 1996.

[12] Z. Waszczyszyn. Some applications of artificial neural networks in structural engineering. In: K. Meskouris, U. Witek, A.A. Balkema, eds., *Aspects in Modern Computational Structural Analysis*, 369–383. Rotterdam/Brookfield. 1997.

[13] X. Wu, J. Ghaboussi, J.H. Garret. Use of neural networks in detection of structural damage. *Computers and Structures.* **42**: 649–659, 1992.