Computer Assisted Methods in Engineering and Science, **19**: 339–349, 2012.
Copyright © 2012 by Institute of Fundamental Technological Research, Polish Academy of Sciences

Neural network prediction of load capacity for eccentrically loaded reinforced concrete columns

Magdalena Jakubek
Institute for Computational Civil Engineering
Cracow University of Technology
Warszawska 24, 31-155 Cracow, Poland
e-mail: mj@L5.pk.edu.pl

This paper presents neural networks prediction of load capacity for eccentrically loaded reinforced concrete (RC) columns. The direct modelling of the load capacity of RC columns by means of the finite element method presents several difficulties, mainly in geometry representation and handling of several nonlinearities. Properly trained neural network can provide a useful surrogate model for such columns. The paper discusses architecture and training methods of the both multi-layer perceptron (MLP) and fuzzy weights neural networks (FWNN) for this application. It also presents the performance analysis of the networks trained on data from three independent databases available in the literature.

Keywords: concrete reinforced columns, load capacity, neural networks, fuzzification.

1. Introduction

The analysis of reinforced concrete columns with Finite Element Method is not an easy task. First of all, one have to overcome the difficulties related to building a geometric model and its discretization. Further, the analysis requires some skills in selecting appropriate formulation, element type, handling material and geometric nonlinearities. Finally, from the point of view of design purposes, the results obtained must be further postprocessed to reduce the number of data and to allow their meaningful interpretation.

This paper presents an analysis of the load capacity of eccentrically loaded reinforced concrete columns with multi-layer perceptron (MLP) and fuzzy weight neural network (FWNN). The analysis of such columns under critical load is a complex problem and is mainly based on empirical modeling. Most of the design recommendations have phenomenological origins which can be linked with difficulties in building and handling theoretical models as well as with the need to build formulas suitable for designing such structures. In practice, one has to account for many factors influencing the performance of columns, which makes the problem complex. The author investigates the idea of application of a neural network system for the prediction of columns critical load, including such issues as the number of inputs, selection of training patterns, restrictions of the value of load eccentricity.

2. Performance of reinforced concrete columns

Columns are elements carrying compressive load from the upper parts of a structure onto the lower part or the foundation [6–8]. They can be precast or made directly on construction site. The column cross-section most of the time has the shape of square, rectangle, polygon, T-shape, I-shape or circle. Two main categories of reinforced columns can be distinguished:

– tied columns,

– spirally reinforced columns.

Experimental evidence proves that all elements carrying compressive load should be treated as eccentrically loaded. This is the basis for the assumption present in the Polish Standards that compressive force is applied with some initial eccentricity. There are several other factors influencing the performance of a column, including:

- slenderness of the column;
- way of application of load force;
- way of supporting the column;
- column cross-section;
- distribution of reinforcement;
- material properties of concrete and reinforcement.

Handing all the input parameters in a consistent manner and producing suitable design formulas is difficult in the analytical way. The application of neural network to this task allows us to hope that the resulting formula, while empirical, will capture all the essential influences of the input parameters. Obviously, the quality of neural network approximator depends strongly on the amount and the quality of training data. On the other hand, the neural network framework can be convenient in dealing with noised or inconsistent data.

3. Experimental analysis of reinforced concrete columns under compressive loading

In the presented research data from three independent sources were used. One is the Pacific Earthquake Engineering Research Center Structural Performance Database (PEER database) created at Berkeley University [4]. This database is continuously updated and its additional advantage is precise description of the gathered data.

The second data source is Chudyba's Ph.D. dissertation [2]. The third one is a report published by Cranston in 1972 [3]. Table 1 shows the number of available patterns and the number of patterns excluded from the analysis on various grounds.

Table 1. Number of experimental test cases in the databases.

No.	Database	No of patterns	No of excluded patterns
1	PEER	296	231
2	Chudyba	36	9
3	Cranston	336	0

To make a comparison of the results originating from different experiments possible, the data cases present in the databases are normalized in respect to a cantilever column of the length L_{eqv} shown in Fig. 2. In the databases the following types of column supports are present:

- cantilever columns,
- double-curvature,

- double-ended,
- flexible-base,
- hammerhead.

For each configuration the equivalent length L_{eqv} is defined as the distance between the level of transverse displacement measurement and the base level. For most of the columns the equivalent length L_{eqv} is equal to the nominal length L. In other words, the transverse column displacement was measured at the same level at which the transverse force was applied.

3.1. PEER database

PEER database [4] was created in order to facilitate access to the data concerning structural elements performance under the action of a seismic load. This database is a compliance of results published by various researchers during the period of several years. For each test case the database provides a reference to the original data source, data describing force-displacement relation for the column endpoint, material specification, and column geometry description. The database consists of 296 cases of rectangular cross-section, simply reinforced columns and 160 cases of columns of various cross-sections with spiral reinforcement.

For each column the database provides geometrical parameters as indicated in Table 2. All the gathered test cases can be divided into various categories using two criteria: a) column support type, shown in Fig. 2, b) column reinforcement type.

Table 2. Geometric parameters.

Overall	H or D	Column depth
Column	B	Column width
Dimensions	Area	Cross-sectional area of column
	L	Length of equivalent cantilever
	Total Bars	Number of longitudinal reinforcing bars
Longitudinal	Bar Dia.	Diameter of longitudinal reinforcing bars
reinforcement	Bar Dia. Corner	Diameter of longitudinal corner bars
	Bar Dia. Interm.	Diameter of longitudinal intermediate bars
	Reinf. Ratio	Longitudinal reinforcement ratio

3.2. Chudyba's database

Tests made by Chudyba [2] concerned 36 columns split into four series by nine elements. The columns in each series were made of a different kind of concrete. One of the series was loaded eccentrically (with eccentricity $e = 2.5$ cm), others were loaded axially. All the cross-sections were square with dimensions 15x15 cm. In each series there were three types of columns of the height of 60 cm, 120 cm, 180 cm, respectively. All the columns were reinforced with four reinforcement bars placed at the corners. For each column length the reinforcement bars were of the diameter 10, 14 and 18 mm. Thus the reinforcement ratio was ρ_1=1.40%, 2.74% and 4.52%, respectively. The main reinforcement bars were made of steel A-III (34GS) and transverse bars of steel AO (6 mm in diameter). All the columns were supported as shown in Fig. 2.

3.3. Cranston's database

The Cranston's database is presented in the report [3]. In this report Cranston collected data corresponding to 381 test cases of square and circular columns. The columns were classified according to the type of support as:

– pined columns, equivalent to case c) shown in Fig. 2

– columns as part of frame systems, equivalent to case d) shown in Fig. 2

– columns with both ends fixed, equivalent to case b) shown in Fig. 2.

4. Formulation of fuzzy weights neural networks (FWNNs)

In this section a brief overview of the algorithm for formulation of a fuzzy NN is given. The detailed description of this algorithm can be found in [1, 9], so here only the main points are highlighted.

The formulation of fuzzy NN is done in three stages as shown in Fig. 1. In Stage I the network is trained on the assumed set of training MLP patterns. This results in the initial values of NN weights

$$\mathbf{W}^0 = \{ w_i^0 \,\big|\, i = 1, \ldots, W \}, \tag{1}$$

where W – number of NN parameters (synaptic weights and biases). In Stage II the network is trained as many times as the number of training patterns in Stage I, separately for each input pattern in the training patterns set. After the training a set of weights is completed as the matrix

$$\mathcal{W} = \{\mathbf{W}_i\}_{(W\times L)} = [w_i^{(p)} \,\Big|\, i = 1, \ldots, W;\; p = 1, \ldots, L]. \tag{2}$$

The matrix from Stage II is the basis for computation of the membership functions for each weight of the NN. The particular way of calculating the membership functions depends on their assumed shape.

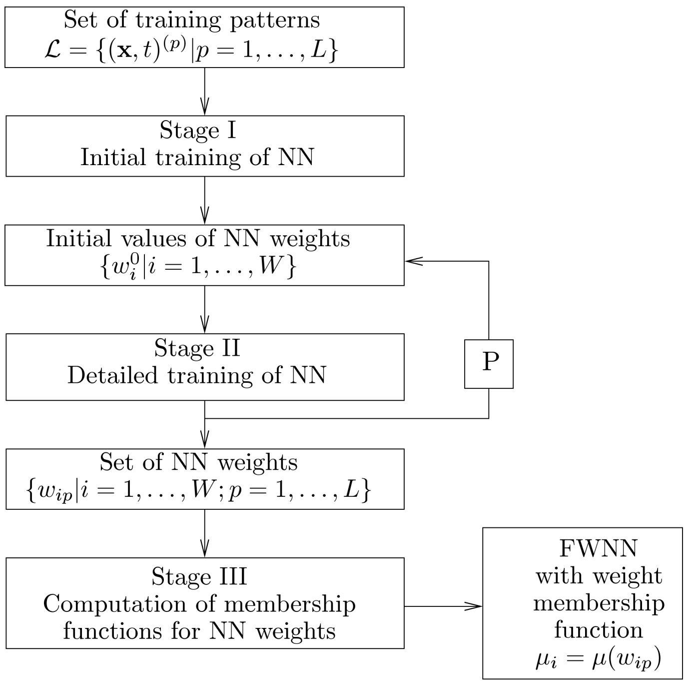


Fig. 1. Schematic algorithm for a Fuzzy Weight Neural Network (FWNN) formulation.

5. Data preprocessing – formulation of a homogenised experimental database

Table 3 shows the number of training and testing patterns selected from the databases described above. In case of Chudyba's database series III of tests was carried out for large eccentricity and thus were excluded from testing in cases **A**, **B**, **C**, **D** and **E**.

Table 3. Number of learning (L) and testing (T) patterns selected for the neural network analysis.

Case	Database	L	T
A	PEER	65	–
	Chudyba	–	27
B	PEER	64	–
	Chudyba	–	27
C	PEER	65	–
	Chudyba	–	27
D	Cranston	296	–
	PEER + Chudyba	–	101
E	Cranston	79	–
	PEER + Chudyba	–	92

PEER database practically does not contain data related to eccentric loading. The eccentricity of load for particular test can be classified as spurious. From the PEER databases only the cases for cross-section with four corner reinforcement bars and non-zero axial force F were selected. Eventually, 65 and 27 test cases from PEER and Chudyba's databases were selected, respectively.

Cranston's database is the richest one. It contains both test cases for large eccentricity as well as test cases for axial load. Neural simulations were thus performed both the whole database (cases **D**) as well as after removing the cases of large eccentricity.

6. Methodology of critical load prediction

The formulation of a neural network for critical load prediction of the reinforced columns requires consideration of the following issues:

- geometric and material parameters,
- training, testing and validation patterns,
- network architecture and training method.

The main factor influencing the decisions related to the issues above is the amount and quality of available data.

For the presented analysis six inputs and one output parameters were selected. The input and output vectors have the form:

$$\mathbf{x} = \{B, H, L, \rho, f_c, f_y\}, \quad y = \{F\}, \tag{3}$$

where B, H – cross-section dimensions, L – equivalent column length, ρ – percentage of reinforcement, f_c – compressive concrete strength, f_y – yield stress for reinforcement steel, F – critical force.

The input parameters were scaled to the range [0.1, 0.9] and output critical force was normalised by the maximum value of the encountered critical force F_s. MLP with simple hidden layer was used. The training method was based on Levenberg-Marquardt algorithm. With the three distinct databases it was decided that the best way of testing generalization properties of neural network is to use one database exclusively for the purpose of validation.

6.1. Case A – training on PEER database, testing on Chudyba's database

The data cases selected from PEER database (65 items) were randomly split into training patterns $L = 65$ and validation patterns $T = 10$. The number of neurons in the hidden layer was set to 3 giving the architecture 6-3-1 and 25 free parameters. Selection of the network architecture was based on the ground of equilibrating the training and validation results. The trained network was then tested on 27 test patterns. The results are presented in the first row of Table 4.

6.2. Case B – training on modified PEER database, testing on Chudyba's database

This case is a modification of case A in which the quality of training patterns was analyzed. On the basis of leave-one-out method a training pattern with the worst validation error was pruned from the training set. It has turned out that the pruned pattern is the one with minimal critical load. The results obtained are presented in the second row of Table 4.

6.3. Case C – PCA analysis, training on PEER database, testing on Chudyba's database

The size of the input vector influences the size of network and the performance of the resulting approximator. Generally, it is beneficial to reduce the size of input vector applaying the Principal Component Analysis (PCA). The analysis of correlation matrix of input parameters gives the following eigenvalues:

$$\boldsymbol{\lambda} = \{12.86,\ 12.27,\ 10.22,\ 7.99,\ 6.98,\ 4.40\}. \tag{4}$$

In terms of the contribution of each of the input parameters to the global data variance the results are

$$\mathrm{m} = \{29.9,\ 27.3,\ 18.9,\ 11.6,\ 8.8,\ 3.5\}\ \%. \tag{5}$$

Looking at the above results one cannot really justify pruning of the input vector, but skipping the component of the smallest contribution, which is 3.5%, should not make the results drastically worse.

The same simulations as in cases **A** and **B** repeated for the input vector with five elements confirm the above statement – the errors are slightly worse but the correlation coefficients are satisfactory. The results are shown in the third row of Table 4.

6.4. Case D – training on Cranston database, testing on merged PEER and Chudyba's databases

Cases **A-C** considered only axially loaded columns. It is obvious that this is merely idealization of a real world situation where, inevitably, some eccentricity is likely to appear. Cranston's database is the biggest one among the three considered bases with 29% test cases. The load eccentricity ranges from 0 mm to 229 mm. However, the quality of this database is not satisfactory, mainly due to poor test cases description, especially with respect to load description.

The same training strategy as in the previous case was applied. The difference was in the size of the input vector, which was extended with eccentricity parameter

$$\mathbf{x} = \{B, H, L, \rho, e, f_c, f_y\}. \tag{6}$$

The corresponding neural network architecture was 7-3-1 with 28 free parameters. The results of testing of this network are shown in the fourth row of Table 4.

Table 4. Summary of training and testing errors for the considered analysis cases.

Case	Critical force $[F_{\min}, F_{\max}]$ [kN]	L	T	Statistical measures			
				St ε_{L} [kN]	r_{L}	St ε_{T} [kN]	r_{T}
A	95–2176	65	27	141.42	0.930	45.23	0.853
B	160–2176	64	27	131.21	0.851	87.03	0.878
C	95–2176	65	27	163.78	0.823	199.01	0.881
D	61–2211	296	101	189.01	0.771	280.03	0.627
E	95–2211	79	92	133.17	0.891	162.91	0.917

6.5. Case E – training on modified Cranston's database, testing on merged PEER and Chudyba's database

The results of case **D** were unsatisfactory because the neural network was trained on the whole Cranston's database, including also the test cases with large eccentricity. The critical force for large eccentricity turned out to be much smaller than for other cases. What is more, the critical force turns out to be very sensible to the eccentricity value, resulting in abrupt changes in force for the change in eccentricity.

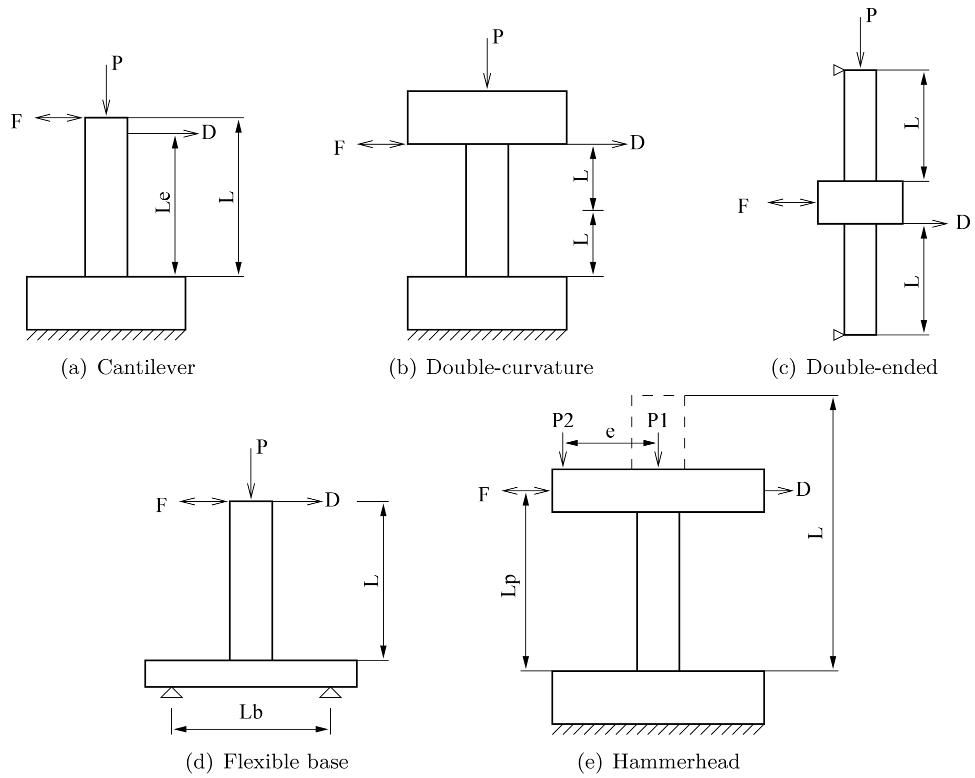


Fig. 2. Column test configuration in PEER database.

Unfortunately, the number of test cases with large eccentricity was too small for the network to accurately "recognize" and "learn" the relation of force to eccentricity for the whole range of the latter. To remedy this, it was decided to prune the database with test cases of eccentricity larger than 5 mm. The threshold value of 5 mm was taken arbitrarily assuming that it is close to a spurious eccentricity of otherwise axially loaded columns. The resulting training set was thus restricted to 79 training patterns. The analysis was carried out for the network of architecture 7-3-1 and the results are presented in the last row of Table 4.

7. Application of FWNN

After analyzing the results of the cases mentioned above, two of them, namely **A** and **E** were selected and the analysis with a fuzzy weights neural network (FWNN) was carried on them. The FWNN analysis started from the configuration parameters (weights and biases) of the classical MLP used in the previous analysis. These parameters were used to initialize the training process of FWNN. The training process was continued further using the learning patterns (for case **A** $L = 65$ patterns and for case **E** $L = 79$ patterns). In the fuzzification of the neural network two types of membership function were used, triangular and nonlinear ones.

Table 5 presents average relative errors, RMSE and correlation coefficients calculated for the normalized output values $\hat{y}_{\mathrm{FWNN}}$ and $\hat{y}_{\mathrm{exp}}$. The normalization means that the results are divided by the maximal value in the experimental data. All values listed in Table 5 pertain to $\alpha = 1.0$ cut.

Table 5. Learning and testing errors for FWNN for $\alpha = 1.0$.

Case	No. patterns	avr *ep* [%]	St ε	r	RMSE
A	$L = 65$	27.12	111.83	0.949	0.0546
	$T = 27$	7.33	60.39	0.852	0.0712
E	$L = 65$	27.22	141.28	0.887	0.0922
	$T = 92$	27.15	118.98	0.962	0.0877

The relation between experimental data and FWNN analysis results for $\alpha = 1.0$ cut is shown in Fig. 3. The points on the graphs cluster around the 20 % error cone around the diagonal, which indicates that no gross error was made.

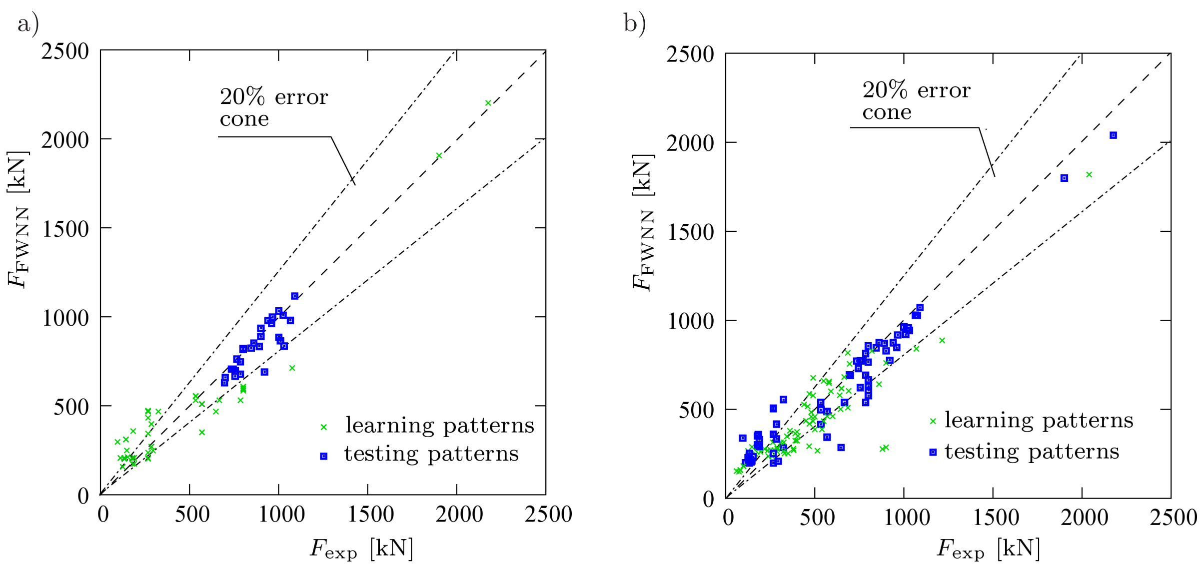


Fig. 3. Comparison of NN simulation results versus experimental data for: a) case **A**, b) case **E**.

Another visualization of the results can be done by constructing cumulative distribution function curves (ogives) (Fig. 4) calculated for the difference between experimental data and FWNN analysis results. The ogive curves allow one to easily find the probability such that the relative error of subsequent calculations is smaller than the assumed value. In the analyzed cases the probability of the error *ep* being smaller or equal 20 % is approximately 0.7.

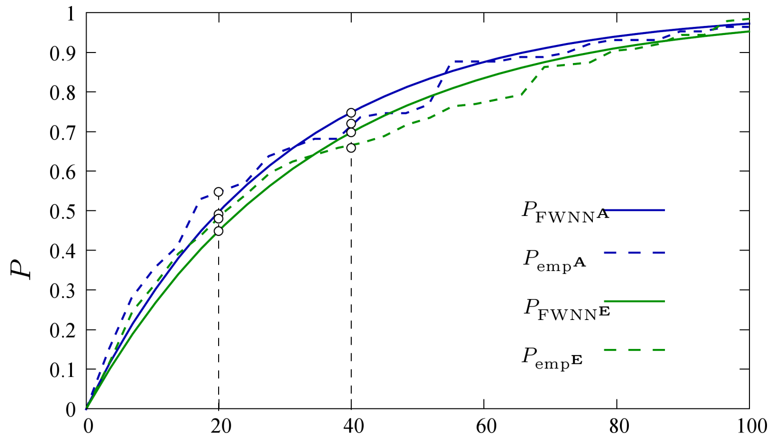
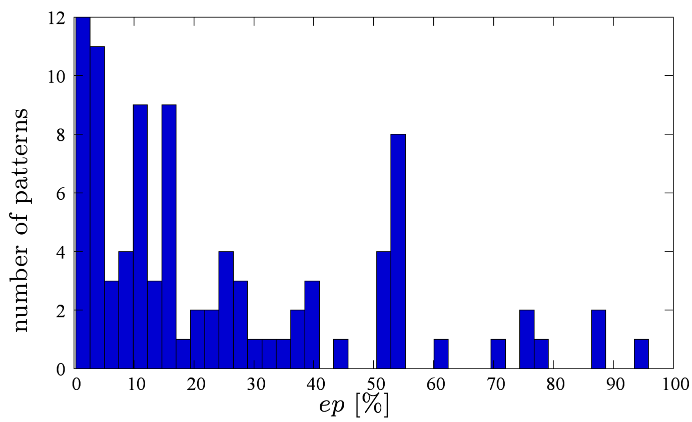


Fig. 4. Cases **A**, **E**. Cumulative distribution function curves (ogives) for empirical and FWNN results for the α-cut of $\alpha = 1$.

In addition to the above, Figs. 5a, 5b show the histograms of relative errors of the neural analysis and simulation data generated for cases **A** and **E**. From these figures it can be seen that the neural

a)



b)

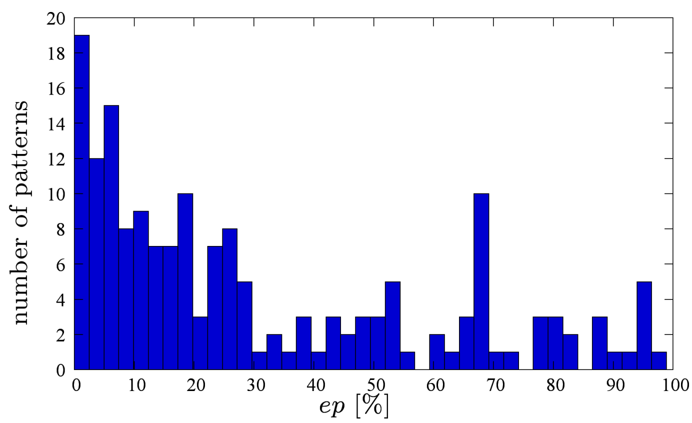


Fig. 5. Cases **A** a) and **E** b) – histograms of relative error *ep* for the FWNN for α-cut of $\alpha = 1.0$.

prediction is relatively good (without distinguishing training and testing patterns) because most of the data have relative average error smaller than 20%.

Figures 6a and 6b show a comparison of the critical load F for two α-cuts for $\alpha = 0.9, 0.75$. It can be seen in these figures that even for $\alpha = 0.75$ the majority of the intervals are located in 20% error cone.

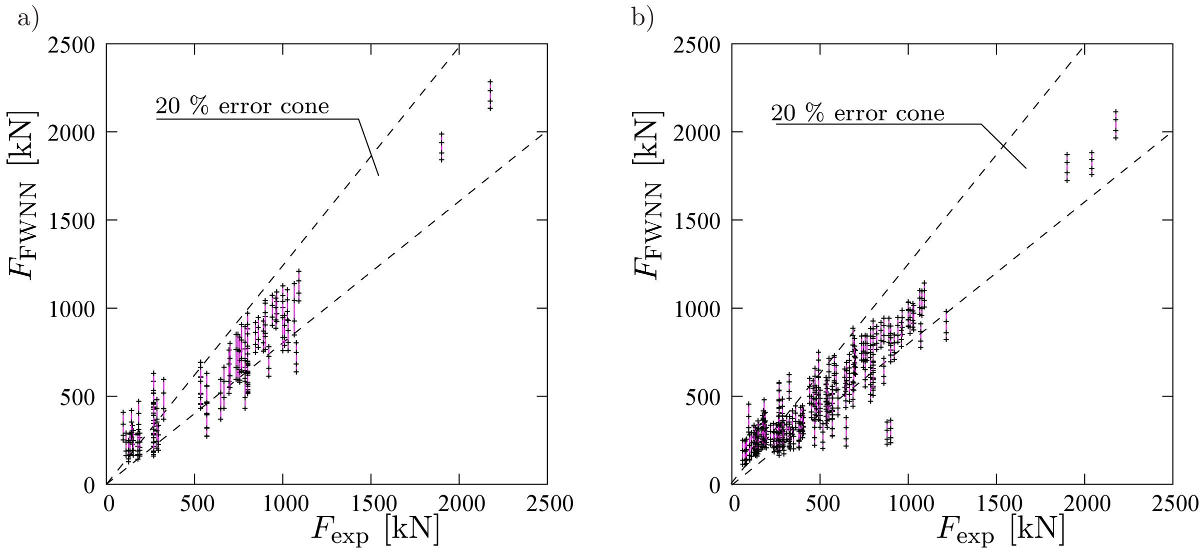


Fig. 6. Cases **A** a) and **E** b). Comparison of experimental data F_{exp} and FWNN results F_{FWNN} for the α-cuts for $\alpha = 0.9,\ 0.75$.

8. Estimation of critical load according to Polish standards

The results obtained by the neural network approximator for columns described in Cranston's and Chudyba's databases were compared with those values resulting from the Polish Standards design recommendations. Figure 7a shows the comparison of the critical load obtained from FWNN analysis and the experimental values from Chudyba's database. Figure 7b shows a similar comparison

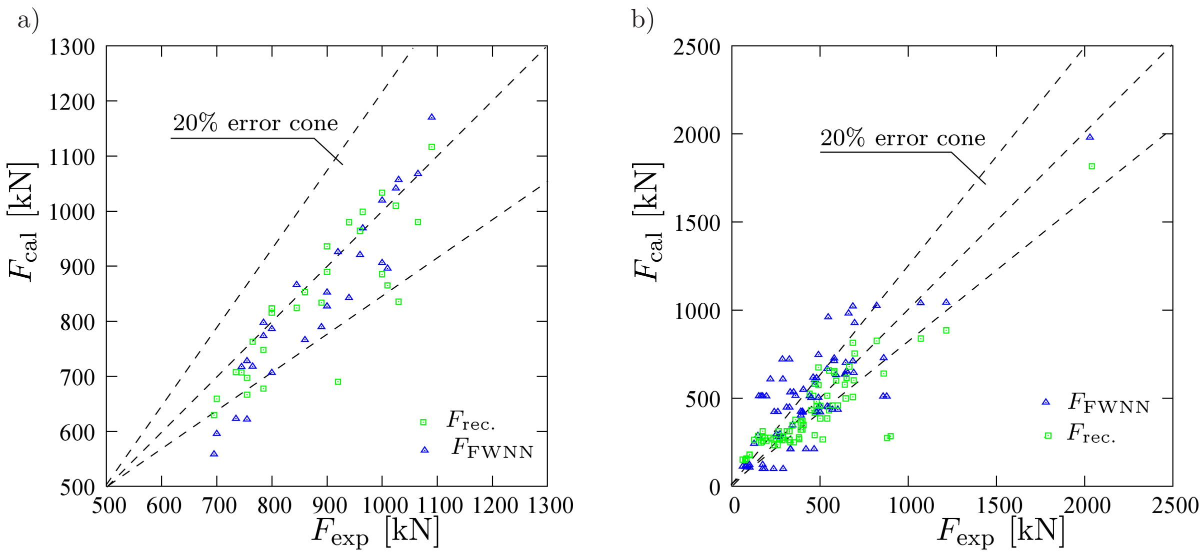


Fig. 7. Comparison of both the critical load calculated according to design recommendation [5] and calculated by FWNN for α-cross-section 1.0 against the experimental results from a) Chudyba's database, b) Cranston's database.

but for Cranston's database. Unfortunately, for this case the relative errors between values resulting from the design recommendations and experimental ones are even five times greater than respective errors for Chudyba's database. This discrepancy is caused first of all by the values of parameters taken for design recommendation calculations. In the case of Chudyba's database the true average concrete strength is known. In the case of Cranston's database we have only the computed approximate values.

9. Conclusion

This paper presents results of application of neural network approximator to the problem of prediction of the critical load for eccentrically loaded concrete columns. These results allow us to draw the conclusions pertaining to both the formulation of a neural network and the quality of experimental data used in the analyses.

- The total number of 674 patterns was used to formulate the neural network approximator. Assuming 6-dimensional input space and uniform density regular hyper-grid, we get $\sqrt[6]{674} \approx 2.9$ grid points per dimension on average. This is enough to capture linear or even quadratic behaviour in respect to each input variable but might not be enough to resolve a more complex critical load 'landscape'. This concerns especially a finer resolution and higher accuracy requirements. On the other hand, the neural network was formulated for patterns taken from three independent databases and such an approach can give more confidence in the NN generalization properties.
- Having three different databases at hand, different configuration of selection of training and testing patterns were applied. The main assumption was that patterns for training and, respectively, testing are selected from a single database. Random selection of patterns from all three databases would destroy their main advantage, which is their independence.
- The testing patterns were not used in the training phase so the testing results are an objective measure for the network generalization properties.
- The best results were obtained for cases **A** and **E**. In both cases the correlation coefficients between experimental and simulated values were in the range 0.850–0.900.

References

[1] M. Jakubek. *Application of artificial neural networks to selected problems of experimental mechanics of materials and structures*, (in Polish). Ph.D. thesis, Cracow University of Technology, 2008.

[2] K. Chudyba. *The influence of material and geometric characteristics on the compressive load capacity of reinforced concrete columns*, (in Polish). Ph.D. thesis, Cracow University of Technology, 1999.

[3] W.B. Cranston. Analysis and design of reinforced concrete columns. Research report 20, Cement and Concrete Association, 1972.

[4] http://www.ce.washington.edu/ peera1/, 2007.

[5] Concrete and reinforced concrete structures. Static calculations and design Polish Standards PN-B-03264:2002, Polish Normalisation Committee, Warszawa, 2002.

[6] M. Bradford. Short-time and time dependent design rules for slender RC columns of circular sections. *Magazine of Concrete Research*, 159: 74–79, 1992.

[7] Design of concrete structures. Part 1: General rules and rules for buildings. Technical report ENV 1992-1-1, 1991.

[8] A. Lapko, B.C. Jansen. *Reinforced concrete structures – design algorithms and recommendations.* Arkady, Warszawa, 2006.

[9] M. Jakubek. Fuzzy weight neural network in the analysis of concrete specimens and R/C column buckling tests. *Computer Assisted Mechanics and Engineering Sciences*, **18**(4): 243:254, 2011.