Computer Assisted Mechanics and Engineering Sciences, **18**: 243–254, 2011.

Fuzzy weight neural network in the analysis of concrete specimens and R/C column buckling tests

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

The paper describes the applications of back propagation neural networks with the ability to process input and output variables expressed as fuzzy numbers. The presentation of an algorithm for finding fuzzy neural network weights is followed by three examples of applications of this technique to the problems of implicit modelling of material and structure behaviour. The following problems are considered: prediction of concrete fatigue failure, high performance concrete strength prediction, and prediction of critical axial load for eccentrically loaded reinforced concrete columns.

Keywords: neural networks, fuzzy weight neural network, strength of high performance concrete, buckling of reinforced columns.

1. Introduction

The majority of mathematical models used in engineering and science is formulated for state variables and model parameters expressed in crisp numbers. This leads to several mathematical idealisations, for instance the perfect circle. However, it is common experience that the real world is far from ideal. Even a seemingly simple question of what the diameter of a pipe is, can lead to surprisingly complex answer. This is due to the presence of imperfections and the fact that the real world answers are fuzzy, probabilistic in nature. Such observations and a similar line of thought motivated the introduction of fuzzy logic, pioneered by L. Zadeh [1].

Development of the fuzzy set theory opened the door to the theory of fuzzy numbers. Unlike real numbers, fuzzy numbers can better capture uncertainties of real world variables. Fuzzy numbers are, for instance, a very convenient tool to express uncertainties related to measurement data. Analysis of measurement data is in turn a ground where fuzzy numbers can meet another very convenient tool, which are neural networks. Some of the prominent applications neural networks are regression problems in which implicit modelling gives relations between input and output parameters. By endowing neural networks with the ability to process fuzzy input and output variables one obtains a convenient framework in which surrogate modelling techniques and handling of uncertain or stochastic data can be used at the same time.

The article presents a way to combine neural networks and fuzzy numbers analysis and shows how this new tool can be applied to three distinct problems of building implicit models for materials and structures behaviour [4]. In all three cases the implicit models were built on experimental data sets, with all the ensuing consequences, that is an insufficient number of data points, unknown coverage of the domain space, noised and erroneous data.

2. Basic properties of fuzzy numbers

The neural networks described previously work on the basis of sets of real numbers which will be called crisp sets and crisp numbers, respectively. The crisp set is defined as:

$$A = \{x|x \in \mathcal{X}\} \ , \tag{1}$$

where x – an element belonging to an assumed domain $\mathcal{X}$. In most cases the domain $\mathcal{X}$ is assumed to be the set of integer numbers or real numbers.

The use of real numbers for quantisation of physical parameters is a sort of idealisation, because in real life we have to deal with a fixed resolution of the measuring devices, uncertainties in mathematical model, not to mention the most fundamental uncertainties of quantum mechanics.

Most of the shortcomings of modelling with real numbers can be remedied by the use of fuzzy sets and fuzzy logic introduced by L. Zadeh [1]. An in-depth introduction to this topic can be found for instance in [2, 3]. At this point only the basic idea and notation necessary for further presentation will be introduced.

Fuzzy set is defined with respect to a pair:

$$A = \{(x, \mu_A(x))|x \in \mathcal{X}\} \ , \tag{2}$$

where $\mu_A(x) \in [0, 1]$ – membership function (MF) of element x to the set A.

For each element $x \in \mathbf{X}$ the membership function assigns a grade of the statement that the element belongs to the set A. One can distinguish three distinct states of element membership:

1. $\mu_A(x) = 1$ – full membership to the set A;
2. $\mu_A(x) = 0$ – no membership of x to the set A, thus $x \notin A$;
3. $0 < \mu_A(x) < 1$ – partial membership of x to the set A.

There are many possible types of membership function. Below three common types are presented, which differ in the number of function parameters. These functions are also illustrated in Fig. 1.

a) singleton (crisp value/discrete value)

$$\mu_A(x; c) = \delta(x - c) = \begin{cases} 1 & \text{for} \quad x = c, \\ 0 & \text{for} \quad x \neq c; \end{cases} \tag{3}$$

b) Gaussian shape exponential function:

$$\mu_A(x; c, \sigma) = \exp\left[-\frac{(x - c)^2}{2\sigma^2}\right]; \tag{4}$$

c) triangular shape function

$$\mu_A(x; a, b, c) = \begin{cases} 0 & \text{for} \quad x \leq a, \\ (x - a)/(c - a) & \text{for} \quad a \leq x \leq c, \\ (b - x)/(b - c) & \text{for} \quad c \leq x \leq b, \\ 0 & \text{for} \quad x \geq b. \end{cases} \tag{5}$$

One of the most important technical concepts introduced by the fuzzy set theory is the so-called α-cut, defined as follows:

$$A_\alpha = \{x|\mu_A(x) \geq \alpha, \ x \in \mathcal{X}\} \quad \text{for} \quad \alpha \in [0,1]. \tag{6}$$

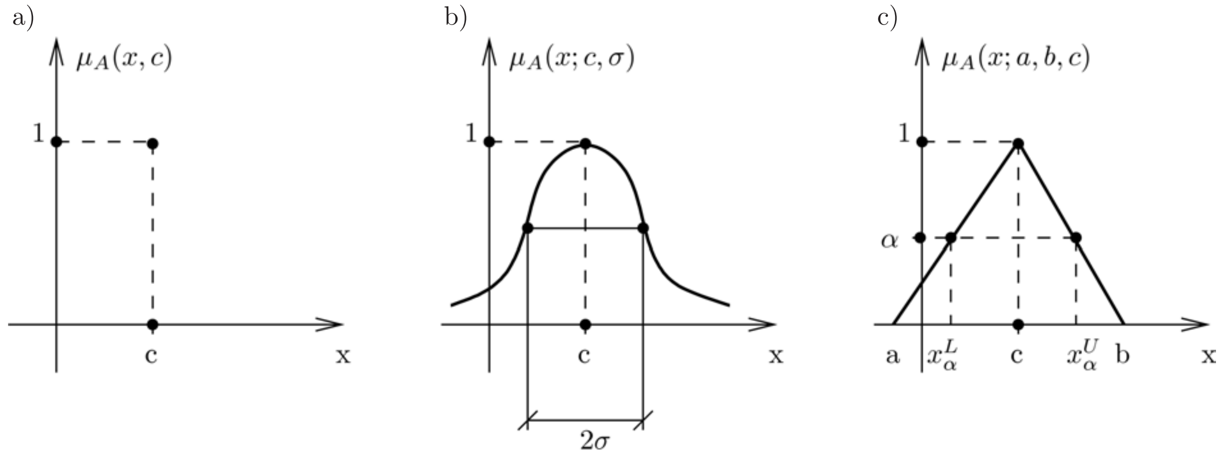

Fig. 1. Membership functions a) singleton, b) Gaussian, c) triangular.

A α-cut can be associated with the range given by the formula:

$$A_\alpha = [x^{\mathrm{L}}, x^{\mathrm{U}}]_\alpha = [x^{\mathrm{L}}_\alpha, x^{\mathrm{U}}_\alpha], \tag{7}$$

where $x^{\mathrm{L}}_\alpha, x^{\mathrm{U}}_\alpha$ are the minimum and maximum function value in the range. An illustration of a α-cut is given in Fig. 1c with assumed triangular membership function.

The concept of α-cut allows us to define arithmetic operations (addition, subtraction and multiplication) with the help of interval arithmetic:

$$A_\alpha + B_\alpha = [a,b] + [c,d] = [a+c,\ b+d], \tag{8}$$

$$A_\alpha - B_\alpha = [a,b] - [c,d] = [a-d,\ b-c], \tag{9}$$

$$A_\alpha \times B_\alpha = [a,b] \times [c,d] = [\min(ac,ad,bc,bd),\ \max(ac,ad,bc,bd)]. \tag{10}$$

By defining the above operations it is possible to introduce the concept of fuzzy numbers and treat them as an extension of crisp real numbers. Fuzzy numbers in turn are a very convenient setting to express the uncertainties of the modelled phenomena mentioned above.

3. Formulation of fuzzy weights neural networks (FWNNs)

A method for formulation of a fuzzy NN is shown in Fig. 2 where a schematic algorithm is presented. It is related to a standard multilayer feed-forward neural network, called for short MLP. The formulation assumes the training set of parameters:

$$\mathcal{L} = \{(\mathbf{x}, t)^{(p)} | p = 1, \ldots, L\}. \tag{11}$$

with $\mathbf{x}$ being the network input vector and t a single network output value.

Let us assume that the network was designed using a corresponding cross-validation procedure using subsets selected from (11). The formulated network is then trained on the whole set at Stage I of the algorithm shown in Fig. 2. A set of NN weights is collected as a set of initial value weights

$$\mathbf{W}^0 = \{\, w^0_i \big|\, i = 1, \ldots, W\}, \tag{12}$$

where W – number of NN parameters (synaptic weights and biases).

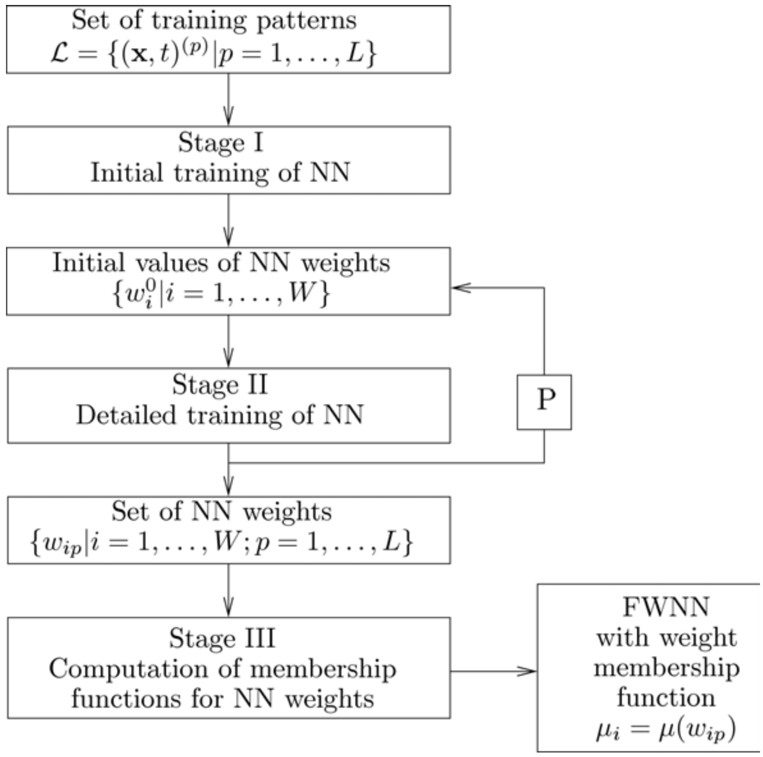

Fig. 2. Schematic algorithm of the Fuzzy Weight Neural Network (FWNN) formulation.

The weights w_i^0 are adopted as initial weights to learn weights corresponding to each pattern of the training set. At Stage II the network is trained L times for a sequence of single patterns $p = 1, \ldots, L$. After the training a set of weights is completed as the matrix

$$\mathcal{W} = \{\mathbf{W}_i\}_{(W \times L)} = [w_i^{(p)} \Big| \, i = 1, \ldots, W; \; p = 1, \ldots, L]. \tag{13}$$

The membership functions for the NN weights $\mu_i = \mu(W_i)$ are computed at Stage III.

In Fig. 3a the triangular shape of MF (t) is shown for the weight w. The distances $3\sigma_L$ and $3\sigma_U$ are measured from the mean value $\overline{w}$, where σ_L, σ_U – standard errors of patterns p that are smaller or greater than $\overline{w}$, respectively. The interval values of the α-cut $[w^L, w^U]_\alpha$ are shown in Fig. 3a as w_α^L, w_α^U.

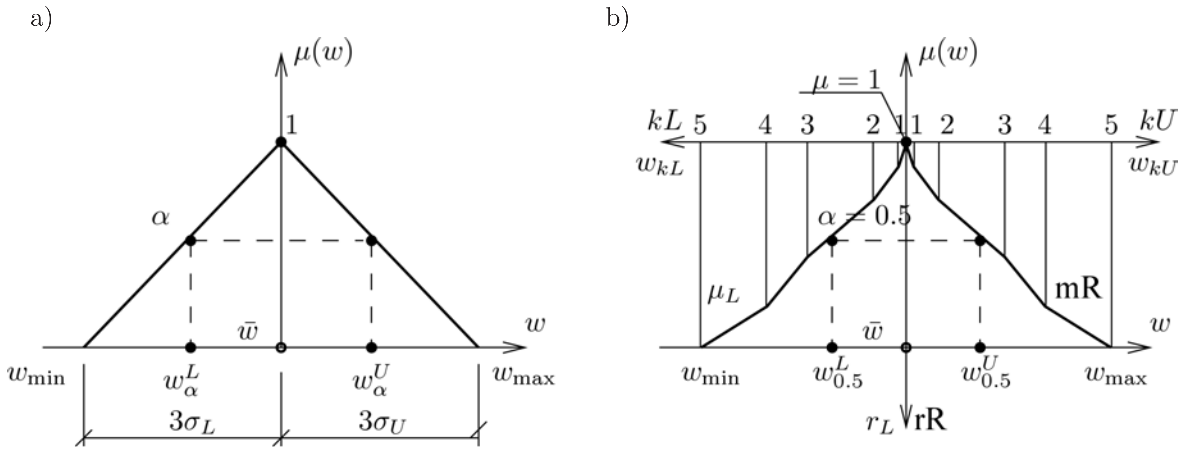

Fig. 3. Shapes of membership functions for FWNN weights a) triangular (t), b) nonlinear (n).

The order of formulation of a nonlinear MF (function (n)) is shown in Fig. 3b. The method shown was proposed in [5, 11]. The idea lies in the computation of discrete cumulative functions for the ranges $w_{\min} \leq w_{kL} < \overline{w}$ and $\overline{w} \leq w_{kU} \leq w_{\max}$, where $kL = 1, \cdots NL$; $kU = 1, \cdots NU$ – numbers of weights on the left-hand side L and right-side U, measured from the mean weight values $\overline{w}$. The ratio of cumulative error is calculated for the left- and right- hand sides:

$$rS = \left(\sum_{k=1}^{NS} kS \right) / NS = 1 - \mu S \quad \text{for} \quad S = L, U, \tag{14}$$

where μS – values of membership function for the side $S = L$, U, respectively.

The formulated MF is composed of two linear piece-wise branches μL, $\mu U \in [0, 1]$ as shown in Fig. 3b. In case of the α-cut the interval values w_α^L and w_α^U can be computed by means of linear interpolation (in Fig. 3b the cut $\alpha = 0.5$ is shown for $NL = NU = 5$).

After the membership functions are formulated for each NN weight the fuzzy network is ready for operation. The network can be used for interval values of inputs $[x_j^L \, x_j^U]_\alpha$ for both fuzzy type variables, i.e. $(x_j^L \neq x_j^U)_\alpha$, and crisp inputs, i.e. $x_j^L = x_j^U$ for each α-cut. In general, the outputs are computed as intervals $[y_m^L, y_m^U]_\alpha$ for both fuzzy and crisp inputs. They are computed by means of interval arithmetic manipulations for fixed α-cuts.

4. Applications of FWNN

4.1. Interval prediction of concrete specimens fatigue failure

The analysis and prediction of concrete fatigue strength is a very complex task. This is because of several processes taking place in the concrete material at different resolution scales. The dependency of these processes on several material parameters and insufficient experimental data base to accurately capture and calibrate these dependencies must also be taken into account. Several of the concrete parameters exhibit fuzzy nature, and this is exactly the reason why fuzzy weights neural networks seem to be a suitable tool to analyse the dependency of the concrete fatigue strength on these parameters.

To verify the last claim a fuzzy neural prediction of concrete fatigue strength was performed [14, 15] and the results compared with empirical formula presented in [12]. The basis for the neural prediction were the result of tests on concrete specimens subjected to compressive loading cycles, that are collected in [12] and reproduced in Table 1.

The data shown in Table 1 were split into two groups. Group I consists of results for which the crisp (not interval) value of concrete fatigue strength f_c was measured. Group II, on the other hand, is characterised by the interval for $f_c \in [f_{c\min}, f_{c\max})]$. Figure 4 shows a comparison of laboratory tests and simulation results for Group I concrete specimen, with case a) showing the neural prediction of the number of fatigue cycles, and case b) showing the prediction according to formula [12].

A better illustration of the results obtained, especially from the point of view of interpretation of results obtained by fuzzy network, is presented in Fig. 5a. This figure provides clear interpretation of the value of α-cut, and allows us to assess quickly the qualitative agreement between the predictions obtained by all three methods.

The quantitative comparison of standard neural prediction, fuzzy one and empirical formula is presented in Table 2 taken from [4].

The same analysis as for data from Group I was done for Group II and the results obtained are shown in Fig. 5b. From the simulations and obtained results one can draw several conclusions. First of all, it can be noticed that fuzzy neural networks give results closer to experimental data than those obtained by empirical formula. One can also observe that neural networks that were trained only on the data from Group I can give false results. This obviously comes

Table 1. Experimental input data collected in [12].

DB No.	Fig. No.	Data source [12]	$R = \sigma_{\min}/\sigma_{\max}$	f [Hz]	f_c [MPa]	Specimen dimensions [cm]	No. of spec.
1	7	37	0.025	16.7	28.0	ϕ 7.6×15.2	18
2	8	38	0.15, 0.38 0.60, 0.88	150.0	41.0	ϕ 5.1×10.2	9
3	9	39	0.44	0.025	28.0	10.2×10.2×30.5	9
4	10	40	0	5.0	[20.0, 30.0]	7.0×7.0×21.0	62
5	11	41	0.14, 0.75	7.5	[14.8, 32.7]	13.0×13.0×40.0	25
6	12	42	0	20.0	[20.0, 45.0]	10.2×10.2×50.8	30
7	13	43	0	20.0	[33.1, 44.8]	10.2×10.2×50.8	7
8	14	19, 20	0.044, 0.75	7.5	[14.8, 32.7]	13.0×13.0×40.0	40
9	15	44	0.05	16.7	25.5, 42.7	ϕ 7.6×15.2	31
10	16	45	0.05	1.167	24.8, 33.1	15.2×15.2×162.6	28
11	17	21	0	[5.0,16.7]	[20.0, 30.0]	different	33
12	18	46	0.074, 0.253	10.0	45.2	ϕ 5.0×10.0	63
13	19	47	0	0.25	20.7	10.2×13.0×82.7	13
14	20	–	0	6.67,15.0	26.2	15.0×15.0×15.0	10

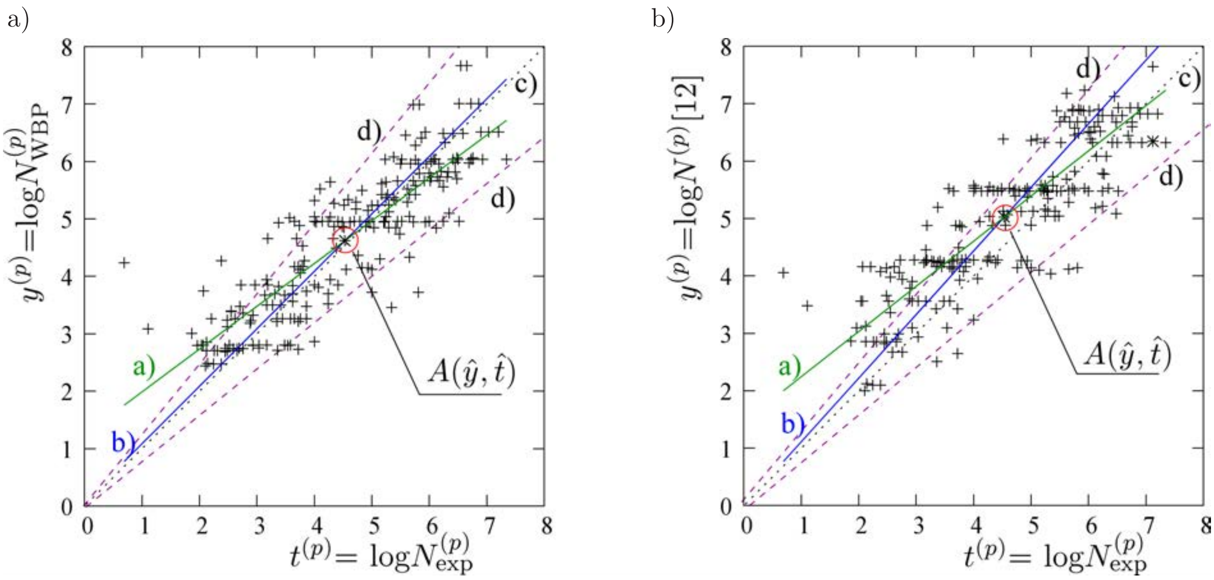

Fig. 4. Comparison of laboratory tests and simulation results for the Group I concrete specimens.

from the nature of the data – crisp values of the fatigue strength as well as other parameters are of just averages and due to the variability of material parameters or simple measurements error can be far from true values. Fuzzy values of the fatigue strength, in turn, catch the influence of several experimental conditions and leave the neural network more freedom in fitting its parameters. The mechanism of α-cuts allows us to easily express our confidence in experimental results.

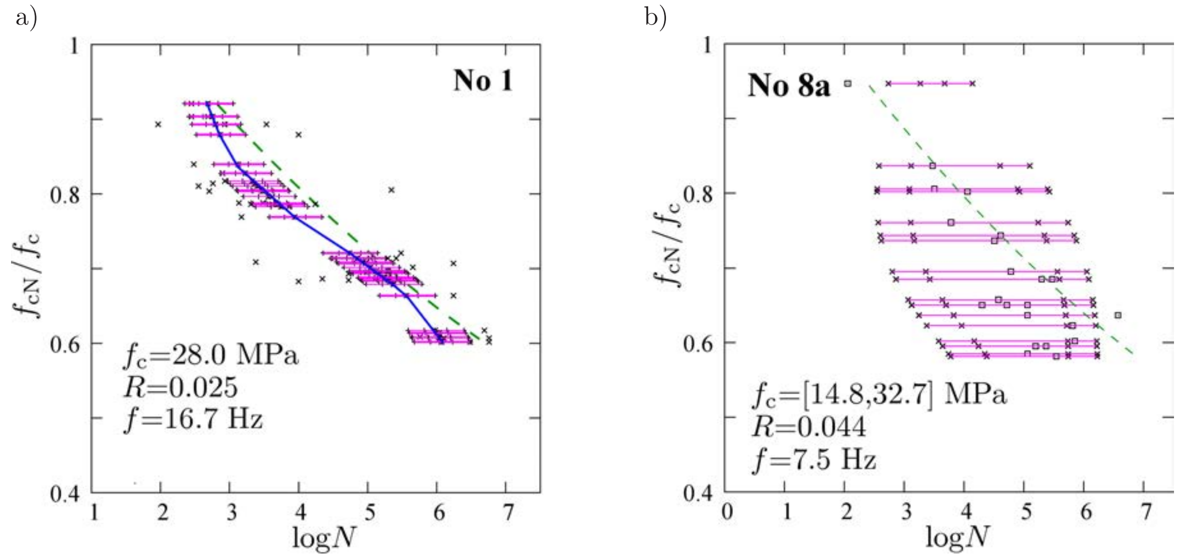

Fig. 5. Mapping f_c–log$N$ obtained by NN (—) and as empirical formula (- -). The α-cuts for $\alpha = 1, 0.9, 0.75$ based on the experimental results from: a) Antrim and Mc Laughlin, 1959, b) Graff and Brenner, 1934.

Table 2. Comparison of error measures and statistical parameters.

Network architecture 4-5-4-1		avr ep [%]		r			St ε		
		L	T	P	L	T	P	L	T
MLA	[13]	12.3	13.3	0.854	–	–	0.76	–	–
MLA	[4]	12.6	15.5	0.863	0.871	0.855	0.736	0.701	0.777
FWNN $\alpha = 1$		13.0	14.7	0.870	0.879	0.861	0.735	0.700	0.772

4.2. Application of a neuro-fuzzy network to HPC strength prediction

The HPC (High Performance Concrete) strength and other performance parameters strongly depend on the components of the concrete mixture. The new technological methods of HPC manufacture have a large number of decisive parameters (up to 30). Despite enormous advances in material modelling, formulating the relations between concrete mixture parameters like C – cement, W – water, S – silica, Su – superplasticiser, FA – fine aggregate, CA – coarse aggregate, etc. and the performance of the final material is very difficult. That is why one has to resort to experimental methods and search for empirical formulas. However, due to the high number of decisive parameters the number of laboratory tests required to obtain meaningful results is huge. Such data can be collected only by gathering results obtained by different laboratories and over several years. Unfortunately, such data are very noisy and their analysis by traditional methods is difficult. This is why neural networks have been proposed [9, 10] for the analysis of HPC strength. Neural networks are able to deal with very noisy or incomplete data. This work is based on paper [10] where the HPC experimental data were analysed with Fuzzy ART-MAP network.

The authors of paper [10] applied their network to a data base consisting of about 340 records. The concrete mixture space was defined by six parameters: C – cement, W – water, S – silica, Su – superplasticiser, FA – fine aggregate, CA – coarse aggregate. Two variants of tests were carried out – with data divided evenly into training and testing sets and with data split arbitrarily into sets of 200 training patterns and 140 testing patterns. In both cases, unexpectedly high correlation coefficients between the predicted and actual strength values were obtained: $r = 0.870$ for the former variant and 0.784 for the latter one.

Table 3 shows the range of HPC mixture parameters and concrete strength. HPC is characterised with a high value of compressive strength and it can be seen that in the data base there were concrete mixtures with relatively low compressive strength. However, because of the high value of the W/C ratio, the lack of superplasticiser and silica it was decided that they would not affect negatively neural network predictions.

Table 3. Minimal and maximal values of concrete components densities and HPC strength f_c.

	C [kg/m^3]	W [kg/m^3]	S [kg/m^3]	Su [kg/m^3]	FA [kg/m^3]	CA [kg/m^3]	f_c [MPa]
min	94	106	0	0	0	0	2
max	1586	540	298	39	1761	1444	136

One of the goals of the performed simulations was the analysis of the influence of partitioning data patterns into learning and testing sets. The following four cases were analysed:

A – random partitioning into L=231 and T=115 learning and testing patterns, respectively;

B – arbitrary partitioning: first L=231 patterns in data base as learning patterns and remaining T=115 as testing patterns;

C – random partitioning into equal sets L=T=173;

D – partitioning into equal sets L=T=169 with the constraint that the patterns from one source were either testing or learning.

Table 4 shows the learning and testing errors for the above listed cases.

Table 4. Comparison of learning and testing errors for the analysed cases.

Case	L	T	learning err.			tesing err.		
			avr ep [%]	St ε [MPa]	r	avr ep [%]	St ε [MPa]	r
A	231	115	13.95	9.33	0.937	21.07	11.80	0.902
B	231	115	12.80	8.89	0.927	19.80	16.45	0.854
C	173	173	11.03	7.87	0.953	12.65	10.15	0.930
D	169	169	12.59	10.11	0.899	18.11	13.80	0.851

One of the conclusions drawn from the above analysis is that the way of patterns partitioning did not substantially affect the results. Better results were obtained in cases A and B, the other ones were also acceptable. This was due to the fact that data were distributed evenly into the parameters space.

The performances of the fuzzy approximator are shown in Fig. 6 for testing and learning phases, respectively.

As can be seen from the figures, most of the experimental results are included in 20% error cone, which is a good result for this kind of analysis. Another illustration of the neural network performance is the plot of cumulative distribution function (CFD) of the results obtained. The cumulative distribution function describes the probability that a real-valued random variable X with a given probability distribution will be found at a value less than or equal to x. The shapes of the curves shown in Fig. 7 confirm the conclusions drawn from Fig. 6, that is the relatively high value of the probability to obtain results at the 20% error level.

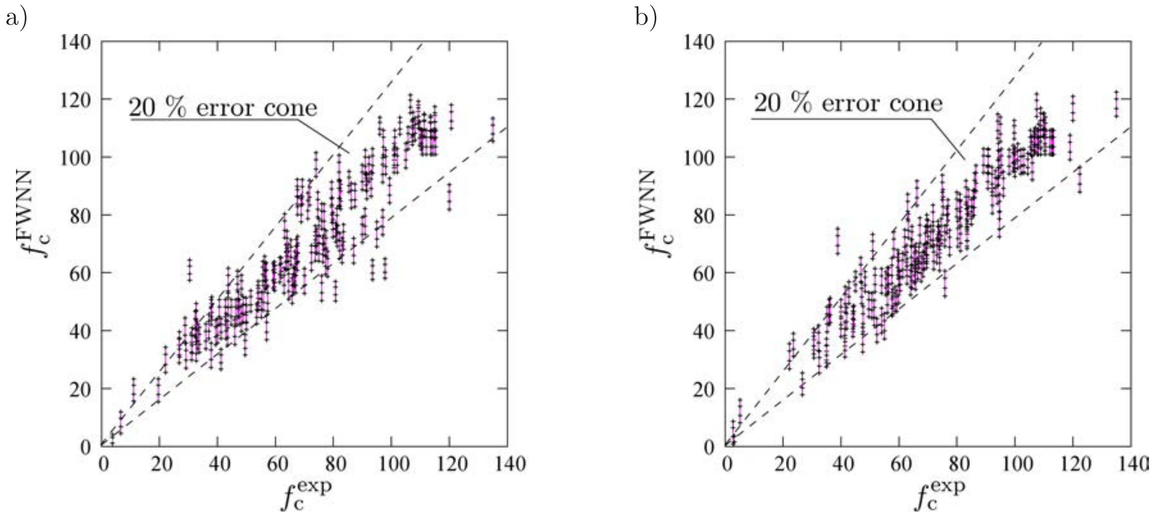

Fig. 6. Case **C** for a) learning patterns b) testing patterns. Comparison of fuzzy neural simulated value of f_c for $\alpha = 0.9,\ 0.75$ with experimental results.

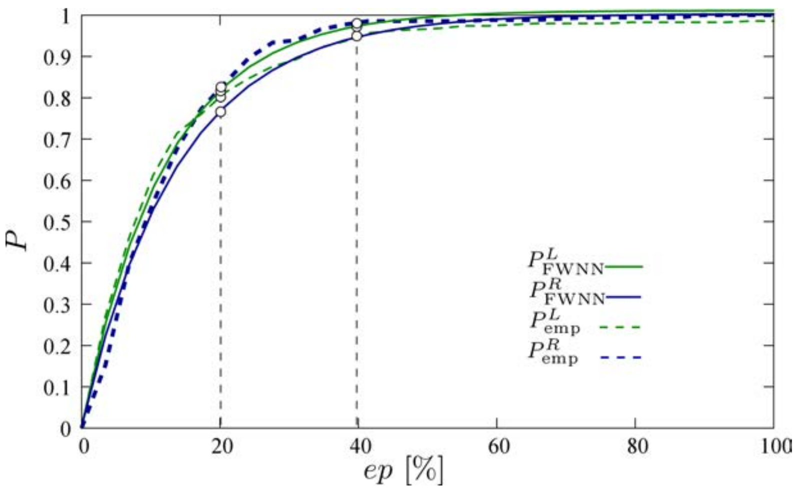

Fig. 7. Case **C**. Cumulative distribution functions for a neural network for $\alpha = 0.75$.

4.3. Neural network prediction of critical axial load for eccentrically loaded reinforced concrete columns

The analysis of the behaviour of reinforced concrete columns under critical axial load is a complex problem and it is based mainly on experimental results. In the presented case the data patterns taken from three independent sources [6–8] were used. In order to make it possible to compare the results pertaining to different test arrangements all data cases were scaled to equivalent reference column of L_{eqv} height.

Table 5 shows number of patterns that were selected from the data bases mentioned above.

Table 5. Number data patterns in data banks used in presented analysis.

No.	Data bank	Total number of patterns	Number of skipped patterns
1	PEER [8]	296	231
2	Chudyba [6]	36	9
3	Cranston [7]	336	0

Table 6, in turn, shows the number of learning and testing patterns that, together with the data bank name, were selected for the neural network analysis. The cases A-E pertain to different ways the patterns were selected for learning and testing. The aim of different ways of selection of patterns was to check the data bases consistency and to make sure that they cover more or less the same parameter space.

Table 6. The cases selected for neural analysis.

Case	Data bank	L	Data bank	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

The second problem addressed in the analysis was the selection of neural network input parameters in the input vector $\mathbf{x}$. It was decided that 6 parameters related to geometric and material properties would be used. The output vector $\mathbf{y}$ was reduced to only one value $\mathbf{y} = [F]$, where F is the critical load. The following input and output components were adopted:

$$\mathbf{x} = \{B, H, L, \rho, f_c, f_y\}, \qquad y = \{F\}, \tag{15}$$

where B, H – cross-section dimensions, L – equivalent column height, ρ – reinforcement percentage, f_c – concrete compressive strength, f_y – reinforcement yield stress, F – critical force. Having at hand the data from three data banks, it was decided that what is the most objective verification of neural network generalisation properties is to verify them on the data from the data bank that were not used to train the network.

Table 7 contains the results of network training and verification for the analysed cases. For case C the same patterns as for case A were used, except that in this case the number of network inputs was reduced from 6 to 5 by means of Principal Component Analysis (PCA) [4]. The PCA allows transformation of the set of possibly correlated inputs into a set of uncorrelated values called principal components. On the basis of the principal component variance it is thus possible to reduce the number of network input values by truncating the components with low variance. The truncated components carry the least significant input data, so their absence should not cause a significant loss of information.

Table 7. Comparison of learning and testing errors for the analysed cases.

Case	Critical load $[F_{\min}, F_{\max}]$ [kN]	L	T	Statistical parameters			
				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

The cases A-E were first analysed with classical, "crisp" neural networks. On the basis of performance parameters such as means square error (MSE), root mean square error (RMSE), standard variation (St ε) and correlation coefficient (r) two cases A and E were selected as the bases for the fuzzy network analysis.

Table 7 shows standard variation and correlation coefficient calculated for normalised output values. All values shown in this table were calculated for the α-cut of $\alpha = 1.0$.

Table 7 shows the number of learning (L) and testing (T) patterns that fall into the ranges defined by the respective α-cuts. Additionally, two last columns of the table show the so-called “success coefficient”, calculated as the percentage of number of patters for which the relative error is less than $(1 - \alpha) \cdot 100\%$, for the given value of α.

Figure 8 shows correlation between the experimental data and results obtained from fuzzy neural network analysis for α-cut with $\alpha = 1$. The correlation points are spread around the diagonal which indicates that no gross errors were made.

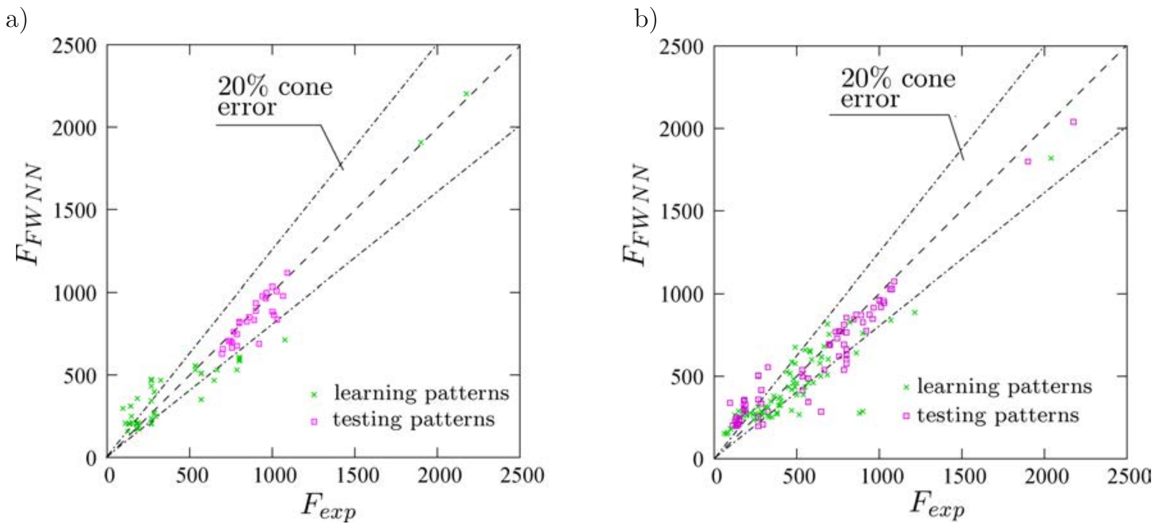

Fig. 8. Comparison of NN simulation results versus experimental data for: a) case **A**, b) case **E**.

Alternative presentation of the results can be made showing ogive curves for the relative error between the experimental critical force and the force obtained from fuzzy neural network. The ogive is the curve of a cumulative distribution function that describes the probability that a real-valued random variable X with a given probability distribution will be found at a value less than or equal to x. The ogive for the cases A and E are shown in Fig. 9.

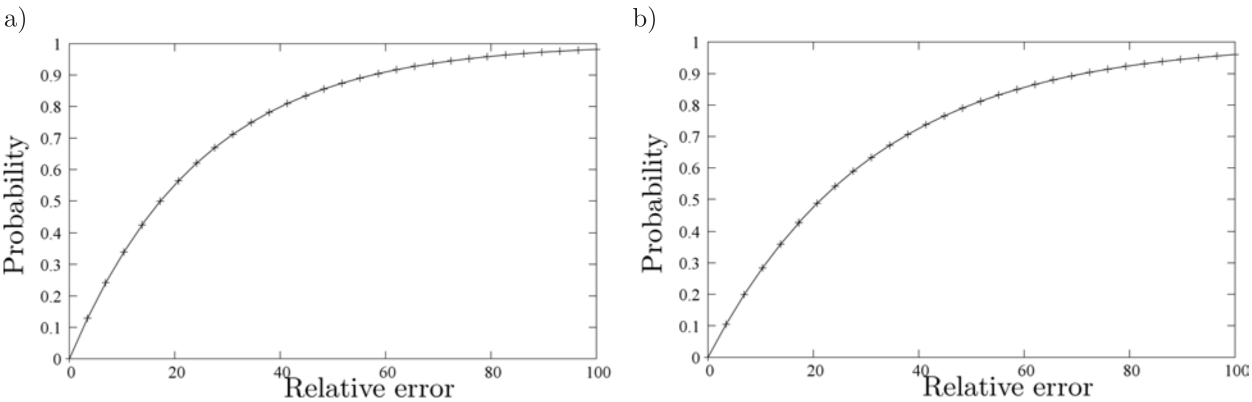

Fig. 9. Ogive curve for $\alpha = 1$: a) case **A**, b) case **E**.

5. Conclusions

The paper outlines an algorithm that enables processing of fuzzy number input and output variables by back propagation neural networks. The performance of the obtained so-called fuzzy weights neural networks is verified on the problems of prediction of concrete fatigue, concrete load capacity, and the prediction of critical load of reinforced concrete columns. For the first of the analysed problems it was shown that the fuzzy neural network can predict the number of cycles to fatigue as accurately as an empirical formula. Contrary to the later it can however handle the interval input data, which can be of great advantage. Concerning the concrete load capacity for high performance concrete, the application of fuzzy neural networks allows one to better handle the confidence in the input data. In the case of the third problem the analysis was strongly influenced by the inconsistencies among the data bases that reduced the reliable number of training patterns. Generally, the results obtained by FWNN are in agreement with empirical modelling and classical neural networks. In contrast to the later, the FWNNs can handle crisp input variables, interval variables and stochastic variables. In some cases they better capture the character of measurement data.

References

[1] L.A. Zadeh. Fuzzy sets. *Information and Control*, **8**: 338–353, 1965.

[2] G.J. Klir, Bo Yuan. *Fuzzy Sets and Fuzzy Logic – Theory and Applications*. Prentice Hall, Upper Saddle River, NJ, 1995.

[3] J-Sh.R. Jang, Ch.-T. Sun, E. Mizutani. *Neuro-Fuzzy and Soft Computing*. Prentice-Hall, Upper Saddle River, NJ, 1997.

[4] M. Jakubek. *Application of artificial neural networks to selected problems of experimental mechanics of materials and structures*. PhD Thesis, Cracow University of Technology, 2008.

[5] M. Jakubek, Z. Waszczyszyn. Neural analysis of concrete fatigue durability by the neuro–fuzzy FWNN. *Artificial Intelligence and Soft Computing*, pp. 1075–1080, Berlin–Heildelberg–New York, 2004. Springer.

[6] 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.

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

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

[9] J. Kasperkiewicz, J. Racz, A. Dubrawski. HPC strength prediction using artificial neural network. *J. Computing in Civil Engineering*, 357–373, 1995.

[10] J. Kasperkiewicz, D. Alterman. Application of artificial intelligence methods for design of concrete mixtures. *Proc. 47th Polish Conf. Civil Engineering*, 331–338, 2001.

[11] E. Pabisek, M. Jakubek, Z. Waszczyszyn. A fuzzy network for the analysis of experimental structural engineering problems. *Neural Network and Soft Computing*, pp. 772–777, Heildelberg – New York, 2003. Physica-Verlag, A Springer-Verlag Co.

[12] K. Furtak. Strength of concrete subjected to multiple repeated loadings (in Polish). *Arch. of Civ. Eng.*, **30**: 677–698, 1984.

[13] Kaliszuk, J., Urbańska, A., Waszczyszyn, Z.: Neural analysis of concrete fatigue durability on the basis of experimental evidence. *Arch. of Civ. Eng.*, **47**: 327–339, 2001.

[14] M. Jakubek, Z. Waszczyszyn. Analysis of concrete fatigue durability by the neuro-fuzzy network FWNN. *Arch. of Civ. Eng.*, **53**: 3–22, 2007.

[15] M. Jakubek and Z. Waszczyszyn. Neural analysis of concrete fatigue durability by the neuro-fuzzy FWNN. In: L. Rutkowski et al. [Eds.], *Artificial Intelligence and Soft Computing – ICAISC 2004*, Springer, LNAI 3070, Berlin-Heidelberg, (2004) 1075–1080.