NEURAL NETWORKS IN PLASTICITY:
SOME NEW RESULTS AND PROSPECTS OF APPLICATIONS

Z. Waszczyszyn
Institute of Computer Methods in Civil Engineering
Cracow University of Technology
Warszawska 24, 31-155 Kraków, Poland

Key words: Back-Propagation Neural Network, Hopfield-Tank Neural Network, Simulation, Identification, Constitutive Equations, Neural Procedures.

Abstract. Two types of artificial neural networks are described in short: i) forward back-propagation neural network, ii) recurrent Hopfield–Tank NN. These network are used for neurocomputing of the following simulation and identification problems: 1) implicit modelling of Chaboche’s viscoplastic constitutive equations, 2) neural procedure for simulating the return mapping algorithm in FE program, 3) analysis of a plane elastoplastic frame with concentrated plasticity, 4) application of inverse analysis and interaction of two types of neural networks for the yield surface tracing, 5) analysis of a tension elastoplastic strip with unilateral boundary constraints. Other fields of neurocomputing applications are also briefly discussed and finally some prospects of neural network applications in plasticity are pointed out.
1 INTRODUCTION

The interest in transferring of methods developed in one discipline to the analysis of problems in other disciplines has evidently increased in recent years. This concerns especially the ‘biologically’ inspired methods of information processing. From among those methods Artificial Neural Networks (ANNs) are worth emphasizing. ANNs have been applied to the analysis of a great amount of problems in science and technology. This concerns also mechanics of structures and materials.

ANNs are called Neural Networks for short and their computer simulation is called neurocomputing. The neurocomputing has many special features which distinguish it from standard computer data processing. Neural network parameters are ‘learned’ during the learning (training) process by means of known examples (patterns). After testing the well trained neural network can operate with a sufficient accuracy for data which were not used in the training and testing processes. ANNs can be used to change input into output data without relations between them known ‘a priori’.

ANNs have been used to the analysis of direct and inverse problems. Those problems corresponds to simulation and identification problems of mechanical systems.

The main emphasis of the paper is put on discussion of the neural analysis of some problems of plasticity by means of two types of ANNs, i.e. the forward multilayer back-propagation neural network (BPNN for short) and the recurrent Hopfield-Tank neural network (HTNN). These neural networks are very concisely described in the following Section 2.

The problems analysed in Section 3 display possibilities of BPNN and HTNN networks in the analysis of some selected problems. This concerns first of all the analysis of constitutive equations on the base of experimental or numerically generated data. ANNs features enable us to simulate directly models of materials with viscoplastic properties using only observable variables (implicit models). In the paper the neural simulation of uniaxial Chaboche’s model is considered.

ANNs have properties complementary to standard numerical methods. This idea is utilised in hybrid FE/ANN programs in which neural procedures are implemented instead of numerical procedures. Simulation of the Return Mapping Algorithms by the BPNN trained off-line is an example of formulation of a neural procedure. The efficiency of this approach is shown in the analysis of elastoplastic plane stress problem, illustrated by the perforated tension strip analysis.

The Panagiotopoulos approach applications are then discussed. The approach combines quadratic programming with bilateral constraints formulations with the HTNN analog. The first application is associated with the analysis of elastoplastic structures with concentrated plasticity and it is illustrated on an example of a plane frame.

Interaction of BPNN and HTNN is explored in the simulation problem of tracing the yield surface on the base of experimental results. Another interaction corresponds to a connection of the hybrid FEM/BPNN program and the HTNN analog for the analysis of a perforated tension strip with unilateral boundary constraints.

Other applications of ANNs in plasticity, discussed in short in Section 4, are associated with load parameters identification in an elastoplastic beam and damage identification using...
data of ultrasonic wave propagation\textsuperscript{13}. Another interesting problem is related to the reliability analysis associated with application of BPNNs for computing patterns in Monte Carlo methods\textsuperscript{14}. The other types of neural networks, i.e. Radial Basis Function NN and ANFIS (Adaptive Neuro-Fuzzy Interface System) are mentioned as neurocomputing tools, used for modelling of plastic and brittle properties of concretes\textsuperscript{15,16}.

At the end of the paper some final remarks are given as conclusions corresponding to prospective applications of neurocomputing in plasticity problems.

2 BASICS OF NEURAL NETWORKS

2.1 Back-Propagation Neural Network

The forward multilayer, back-propagation neural network (called BPNN for short) is shown in Fig. 1a. The main goal of BPNN is mapping of input, i.e. vector $\mathbf{x} \in \mathbb{R}^N$ into output, i.e. vector $\mathbf{y} \in \mathbb{R}^M$. This can be written in short:

$$\mathbf{x}_{N \times 1} \xrightarrow{\text{BPNN}} \mathbf{y}_{M \times 1}, \quad (1a)$$

and in general:

$$\mathbf{x}^{(p)} \rightarrow \mathbf{y}^{(p)}, \quad (1b)$$

where: $p = 1, \ldots, P$ – number of patterns. The mapping is performed by a network composed of processing units (neurons) and connections between them (Fig. 1a). The activation potential $v_i^l$ of a single neuron $i$ in layer $l$ is cumulated in the summing block $\Sigma$ and activated by function $F$ to have only one output $y_i^l$ (Fig. 1b):

$$y_i^l = F(v_i^l), \quad v_i^l = \sum_{j=1}^H w_{ij} x_j + b_i, \quad (2)$$

where layer superscript $l = 1, \ldots, H_{out}$ is omitted at network parameters: $w_{ij}$ – weights of connections, $b_i$ – threshold parameter (bias).

From among many activation functions only the binary and bipolar sigmoidal activation functions (3a) and (3b) are shown in Fig. 1c,d:

$$F(v) = \frac{1}{1 + \exp(-v)} \in (0,1), \quad (3a)$$

$$F(v) = \frac{1 - \exp(-v)}{1 + \exp(-v)} \in (-1,1). \quad (3b)$$

Network parameters $w_{ij}^1$ and $b_i^1$ are computed iteratively in the training process of the so-called supervised learning. This process is related to the training set $\mathbf{L}$ composed of
known pairs, i.e. input vectors $x^{(p)}$ and output vectors (known outputs) $t^{(p)}$. After training the network is tested on the testing set $T$. The training and testing pattern sets can be written in the following form:

$$L = \{(x, t)_{(p)}|p=1, \ldots, P\}, \quad T = \{(x, t)_{(p)}|p=1, \ldots, T\},$$

(4)

where: $L, T$ – number of patterns in the training and testing sets, respectively.

The output vectors $y^{(p)}$, computed by the network, are compared with the target vectors $t^{(p)}$ and then scalar network errors can be computed. Below two forms of the network errors are given:

$$E = \frac{1}{2} \sum_{p=1}^{P} \sum_{i=1}^{M} (t^{(p)}_i - y^{(p)}_i)^2, \quad \text{MSE} = \frac{2E}{P},$$

(5)

where: $E$ – least-mean-square-error, $\text{MSE}$ – mean-square-error.

The network parameters are iteratively computed using the formula:

$$w_{ij}(s+1) = w_{ij}(s) + \Delta w_{ij}(s),$$

(6)

where: $s$ – the number of iteration step. In what follows the layer superscript $l$ is omitted and the bias $b_i$ is formally treated as a weight $w_{i0} = b_i$ for the signal $x_0 = 1$ (Fig. 1b).

From among many formulae for the computation of weight increments $\Delta w_{ij}$ one of the most numerically efficient is Rprop (Resilient-propagation) formula, cf. 2, 17:

$$\Delta w_{ij}(s) = -\eta_i(s) \text{sgn} g_{ij}(s),$$

(7)
where: $g_{ij} = \partial E / \partial w_{ij}$ - gradient of error function (5), $\eta_{ij}$ - locally adaptive learning rate parameter:

$$
\eta_{ij}(s) = \begin{cases} 
\min(\eta^+, \eta_{ij}(s), \eta_{\max}) & \text{for } g_{ij}(s)g_{ij}(s-1) > 0, \\
\max(\eta^+, \eta_{ij}(s), \eta_{\min}) & \text{for } g_{ij}(s)g_{ij}(s-1) < 0, \\
\eta_{ij}(s) & \text{otherwise.}
\end{cases}
$$

Parameters $\eta^+, \eta^-, \eta_{\max}, \eta_{\min}$ are specified in the SNNS manual $^{17}$.

During the iteration process all the training patterns $p = 1,...,L$ are presented. One, forward transmission of signals for all patterns and back propagation of errors is called an epoch. From this point of view the epoch corresponds to iteration step $s$ in (6-8). The iteration process is ended according to stopping criteria. Usually such a criterion is associated with a prescribed number of epochs $S$.

### 2.2 Hopfield-Tank Neural Network

The Hopfield-Tank Neural Network (HTNN) is a recurrent network, i.e. a network with feed-back (Fig.2). The network was originally formulated by J.J.Hopfield in 1982 for recognition of patterns composed of discrete, i.e. binary or bipolar digits. In 1985 Hopfield’s ideas were generalized on continuous variables $^{4}$, and a corresponding HTNN network appears to be especially useful for the analysis of mathematical programming problems $^{2}$.
HTNN is ready for operation after the network parameter $w_{ij}$ and $b_i$ are specified (or computed in the so-called storage phase). HTNN operates in the retrieval phase during the dynamically stable process. This means that after introduction of initial values of the input vector components $x_i(t)$ their current values $x_i(s)$ are computed recurrently continuing the iteration procedure until the stable state of the network is reached.

HTNN is classified as a dynamic network since forward and feedback signals can circulate in the network. From this point of view BPNN is a static network since the input signals are transmitted into forward direction only, i.e. inputs are mapped into the outputs.

For HTNN the energy function is defined $^{1,2}$:

$$E(x(s)) = \sum_{i=1}^{N} \sum_{j=1}^{N} w_{ij} x_i(s)x_j(s) - \sum_{i=1}^{N} b_i x_i(s) + \sum_{i=1}^{N} \int_{0}^{x_i} F_i^{-1}(x) \, dx .$$

Function (9) is valid for the dynamic, continuous in time, process which is described by a set of ordinary differential equations $^{1,2}$:

$$\tau_{ij} \frac{d v_i}{dt} = -\frac{\partial E}{\partial x_i} \bigg|_{x_i(t)} = -v_i + \sum_{j=1}^{N} w_{ij} x_j + b_i ,$$

$$x_i(t) = F_i(v_i(t)) \quad \text{for} \quad i = 1, \ldots, N ,$$

where: $\tau_{ij}$ – time constants. Eqs (10) are called the HTNN evolutionary equations. The stable equilibrium state of the network is defined by the following stopping criteria:

$$\min_{E} = 0 \iff \frac{dx}{dt} = 0 .$$

In the papers by Panagiotopoulos and his associates $^{7,8}$ it was proved that HTNN can be used to the analysis of Quadratic Programming Problems (QPPs) with:

a) bilateral constraints:

$$\min \left\{ \frac{1}{2} x^T A x - b^T x \right\} ,$$

b) unilateral constraints:

$$\min \left\{ \frac{1}{2} x^T A x - b^T x \big| x \geq 0 \right\} .$$

The piece-wise linear activation function is used:

$$x_i = F_i(v_i) = \begin{cases} v_i & \text{if} \quad v_i \geq 0 , \\ 0 & \text{if} \quad v_i < 0 . \end{cases}$$

After substitution
and assumption: \( \tau_i = 1 \) for \( i = 1, \ldots, N \), Eqs (10) can be written in the form:

\[
\frac{d \mathbf{v}}{dt} = \left( -A \mathbf{x} + \mathbf{b} \right) |_\varepsilon \equiv \mathbf{r} ,
\]

where: \( \mathbf{r} \) – vector of residuals at time \( t \).

3 APPLICATIONS OF BPNN AND HTNN IN PLASTICITY

3.1 Modelling of constitutive relationships for viscoplasticity

Two classes of constitutive models can be defined, i.e. explicit and implicit models. The analytical formulation is associated with explicit models which are based on assumed forms of constitutive relationships. Parameters of the assumed functions have to be computed in the frame of identification procedures to match well experimental data. Implicit models do not use material parameters and they are based on observable variables only.

ANNs, and especially BPNNs, can be used to the analysis of both types of models. In case of explicit models BPNNs enable us to identify model parameters as components of the output vector. Neural simulation of implicit models is obviously much more valuable.

On the base of experimental data BPNNs were used for simulating implicit stress-strain relationship for biaxial loading of concrete and moment-curvature relationship of an inelastic beam. A much more complex problem of neural modelling of uniaxial stress of viscoplastic material was discussed in. A more intricate neural simulation of an implicit viscoplastic model was performed in. Some basic results from this paper are discussed below.

3.1.1 Basic relation for Chaboche’s model

In what follows only the uniaxial viscoplastic model is considered as a special case of Chaboche’s model. The strain \( \varepsilon \) is split into elastic and viscoplastic (inelastic and time-dependent) strains \( \varepsilon^e \) and \( \varepsilon^{vp} \), respectively:

\[
\varepsilon = \varepsilon^e + \varepsilon^{vp} .
\]

The flow rule, kinematic and isotropic hardening rules are as follows:
\[
\varepsilon^{vp} = \left( \frac{|\sigma - \chi| - R - k}{K} \right)^n \text{sgn}(\sigma - \chi),
\]
\[
\dot{\chi} = H \dot{\varepsilon}^{vp} - D \chi |\dot{\varepsilon}^{vp}|,
\]
\[
\dot{R} = (h - d \cdot R) |\dot{\varepsilon}^{vp}|,
\]

where: \( \dot{\cdot} \) becomes zero if the value inside is negative, \( \chi \) – back stress related to kinematic hardening, \( R \) – drag stress related to isotropic hardening; \( k, K, n, H, D, h, d \) – seven material parameters.

3.1.2 Neural simulation for reverse cyclic loading

The explicit model (17) was used to compute pseudo-experimental data associated with the reverse cycling loading test (Fig.3):

![Figure 3: Reverse cyclic loading test](image)

The implicit neural model is shown in Fig.4. It corresponds to BPNN in Fig. 1a. The input and output variables are components of the following vectors:

\[
\mathbf{x}_{(\text{in})} = \left\{ \sigma, \chi, R, \varepsilon^{vp} \right\}, \quad \mathbf{y}_{(\text{in})} = \left\{ \dot{\chi}, \dot{R}, \dot{\varepsilon}^{vp} \right\}.
\]

![Figure 4: Neural implicit model](image)
Patterns corresponding to vectors (18):
\[
(x, t)^{(p)} = \{ x^l, y^l \} = \{ \sigma^l, \dot{\chi}^l, R^l, \dot{R}^l, \dot{\epsilon}^l \}
\]  
(19)
are computed for the subsequent time instants \( t = p \) associated with a time increment \( \Delta t \):
\[
y_{t+1} = y_t + \dot{y}_t \Delta t.
\]  
(20)
Initial values are related to
\[
\epsilon = \epsilon_{\infty}, \quad \sigma = E(\epsilon_{\infty} - \epsilon_{\infty}),
\]  
(21)
and other values are extracted from experimental or pseudo-experimental data \( \epsilon \), cf. points in Figs 5a-c.

For computing the training and testing patterns the values of material parameters in (17) and parameters of the reverse cyclic loading test are assumed as those listed in Table 1.

<table>
<thead>
<tr>
<th>Material parameters (stresses in Mpa)</th>
<th>Parameters of reverse cyclic loading test</th>
<th>Number of patterns</th>
</tr>
</thead>
<tbody>
<tr>
<td>( k )</td>
<td>( K )</td>
<td>( n )</td>
</tr>
<tr>
<td>50</td>
<td>50</td>
<td>3</td>
</tr>
</tbody>
</table>

Table 1: Parameters for generating the training and testing patterns

The patterns were generated for five loading cycles (Fig. 5b) at piece-wise constant \( \varepsilon_c \) (Fig. 3a). Discretization of function \( \dot{\epsilon}(t) \) for \( t \in [0.0, 0.76] \) sec was carried out at \( \Delta t = 0.00125 \) sec. In such a way \( P = 613 \) patterns were generated. Starting from \( p = 0 \) every second pattern was included into the training set (corresponding points are shown in Figs 5b-d). Patterns lying between training patterns were collected as the testing set. Totally \( L = 307 \) training patterns and \( T = 306 \) testing patterns were generated.

The SNNS computer simulator and Rprop learning rule (8) were used in the neural analysis. After a number of numerical experiments the BPNN network with bipolar sigmoid activation function (3b) and the structure: 4-6-6-3 was formulated, cf. Fig. 4 (the network fully corresponds to the network shown in Fig. 1a at \( N = 4, H1 = H2 = 6, M = 3 \)). This means that the network has 93 parameters which well correspond to the number of training patterns \( L = 307 \).

The network was trained up to \( S = 10000 \) epochs. The training and testing errors were \( \text{MSEL} (S) = \text{MSET} (S) < 0.0016 \) (in Fig. 5a the measure of error \( \text{SSE} = P \times \text{MSE} \) was used).

In Figs 5b-d there are shown results of computations by the trained network. It is seen that the curves well correlate with the training and testing patterns placed in between the training data. This points out that the trained BPNN very well simulate Chaboche’s model.
Generalization properties of the trained network were examined on two ranges of cyclic strains \( \varepsilon_{\max/\min} = \pm 0.025, \pm 0.072\% \). In Figs. 6a-c results of neural simulation are shown. In case of the cyclic strain range \( \pm 0.025\% \) a good agreement of curves computed by BPNN and the Chaboche analytic model is achieved. This is not the case of the cyclic strain range \( \pm 0.072\% \) which is twice bigger than the range used for the network training. The neural simulation for \( \pm 0.072\% \) is far from “exact” curves corresponding Chaboche’s model. It is visible in Fig. 6d that the neural stress-strain curve deviates from Chaboche’s curve after the strain of 0.036\%, used for training, is exceeded\(^5\).

The BPNN of the same architecture: 4-6-6-3 was trained by means of experimental data taken from test on a steel at temperature of 673 K, cf. references in\(^5\). Tests were carried out only for a quarter of a cycle (tensile stress) at three strain rates: 0.0001, 0.01, 0.5 [%/sec].
Figure 6: Results of operation of the trained BPNN used for the cyclic strain range ± 0.25%:
a) Strain-time curve, b) Stress-time curve; c) Stress-strain curves for the cyclic strain range ± 0.25%,
d) Stress-strain curves for the cyclic strain range ± 0.72%

Fig. 7 there are shown results of simulation by the neural network and the best-fit Chaboche’s curves.

3.2. Analysis of elastoplastic plane stress problem

Analysis of elastoplastic problems by means of FEM programs is numerically much more laborious than the analysis of elastic problems because of constitutive equations of plasticity. They are not only nonlinear but history dependent and it is difficult to make mathematical manipulations with them, e.g. to invert them. That is why changes of numerical procedures related to plasticity models into neural procedures seem to be very prospective.
3.2.1 Return mapping algorithm (RMA)

One of the most important procedures in the FE analysis is associated with computation of the actual stress vector and consistent stiffness matrix at each Gauss point of plane finite elements. The analysis can be made by the Return Mapping Algorithm (RMA)\(^{21}\). RMA enables us to perform mapping shown in Fig. 8a:

$$\{ \sigma^A, \Delta \varepsilon \} \rightarrow \{ \sigma^A, E_D^{ef} \}, \quad (22)$$

where: \(\sigma^A, \sigma^D \in \mathbb{R}^3\) – stored and actual stress vectors, \(\Delta \varepsilon \in \mathbb{R}^3\) – increment of strain vector, \(E_D^{ef} \in \mathbb{R}^3 \times \mathbb{R}^3\) – consistent stiffness matrix.

The constitutive equations are based on the following assumptions: 1) material is homogeneous and isotropic with the Huber- von Mises yield surface and linear isotropic strain-hardening, ii) constitutive equations of classical flow theory are valid.

The main problem of RMA is the analysis of highly nonlinear equation associated with the yield surface \(\varphi_D\) (Fig. 8a):

$$\varphi_D \equiv \varphi (\sigma^D, \Delta \lambda_D) = 0. \quad (23)$$
After the increment of yielding multiplier $\Delta \lambda$ is computed (the actual stress vector $\sigma^D$ is computed simultaneously) the consistent stiffness matrix $E^D_{ep}$ can be computed analytically \(^6\), cf. Fig. 8c.

Figure 8: a) Return Mapping Algorithm in the stress space, b) Scheme of numerical computation of patterns, c) Scheme of BPNN neural procedure

### 3.2.2 Numerical simulation of RMA

BPNN was used as a neural simulator of RMA \(^6\). Patterns for the BPNN training and testing were computed analytically according to scheme shown in Fig. 8b. The strain hardening parameter $\chi$ was fixed and the patterns were formulated with respect to the following dimensionless variables:

\[
\sigma = \sigma / \sigma^B \quad \Delta \bar{\epsilon} = E \Delta \epsilon / \sigma^B \quad \Delta \bar{\lambda} = E \Delta \lambda \quad \chi = E / E
\]  

where: $\sigma^B$ – effective stress at the yield surface $\phi^B = 0$ in Fig. 8a, $E, H$ – elasticity and strain-hardening moduli, respectively.

The 360 stress vector points $\bar{\sigma}^B$ uniformly distributed at the yield surface $\phi^B$ were associated with the stress increments $\Delta \bar{\epsilon}_x, \Delta \bar{\epsilon}_y, \Delta \bar{\gamma}_{xy} \in [-1.0,1.0]$. In this way about $1 \times 10^6$ patterns were computed using a numerical procedure of RMA, assuming the strain-hardening parameter $\chi = 0.032$ as in references \(^{22,23}\).

BPNN with bipolar activation function (3b) was formulated corresponding to the following input and output vectors:

\[
x_{6\text{in}} = \{ \sigma^B, \Delta \bar{\epsilon} \} \quad y_{4\text{out}} = \{ \sigma^D, \Delta \bar{\lambda} \}
\]  

where:

\[
\sigma^S = \{ \sigma_x, \sigma_y, \gamma_{xy} \}^{S} \text{ for } S = B, D \quad \Delta \bar{\epsilon} = \{ \Delta \bar{\epsilon}_x, \Delta \bar{\epsilon}_y, \Delta \bar{\gamma}_{xy} \}
\]  

\[\text{where: } \]
From among $1 \cdot 10^6$ analytically computed patterns only plastically active patterns were randomly selected, i.e. $L = 3349$ and $T = 9044$ for training and testing, respectively. After a cross-validation procedure was used, the BPNN of structure 6-40-20-4 was numerically formulated. The SSNN computer simulator and Rprop learning formula (8) was used for the network training. After $S = 34000$ epochs the training and testing errors were $\text{MSEL} \approx \text{MSET} < 1 \cdot 10^{-4}$.

The efficiency of neural simulator was examined on 10000 patterns selected randomly. It was stated that BPNN needed about 40-50% of computational time consumed by numerical form of RMA.

3.2.3 Analysis of a perforated tension strip

The trained BPNN was incorporated as a neural procedure in the ANKA Finite Element code. In this way the program ANKA–H was implemented. The program can be called a hybrid or neurally supported program FE/BPNN.

Figure 9: a) Geometry and load data, b) Material characteristics, c) FE mesh for a quarter of strip
ANKA and ANKA-H programs were tested for bench-mark type problems \(^{22,25}\) and then the program was used in \(^6\) to the analysis of a perforated tension strip with data taken from \(^{23}\), Figs 9a,b. The 8-node isoparametric plane finite elements with four Gauss points of reduced integration were used. In Fig.9c the FE mesh is shown for a half of the strip.

The results of computations are depicted in Fig.10 as the equilibrium paths \(\Lambda(\dot{u}_A ; \chi)\), where: \(\Lambda\) – load parameter, \(\dot{u}_A\) – horizontal displacement at node, \(\chi \in [0.0, 0.1]\) – fixed values of strain-hardening parameter. The computations were carried out under displacement control \(\tau = u_A\).

The neural procedure BPNN was trained at the hardening parameter \(\chi = 0.032\) but its generalization properties are quite satisfactory for other values of the strain-hardening parameter \(\chi \in [0.0, 0.07]\). It was stated that for \(\chi = 0.032\) and \(\dot{u}_A = 0.55\) mm about 68% of the plate was yielded.

In order to compare processing time used by programs ANKA and ANKA-H the computations were carried out for 60 steps \(\Delta \dot{u}_A = 0.01\) mm up to \(\dot{u}_A = 0.6\) mm. In case of ANKA the processing time was about 16 sec at 132-136 global iterations. When the program ANKA-H was used the results depended on the value of strain hardening parameter. For \(\chi = 0.005\) the ANKA-H time was about 50% of ANKA time and the total number of iterations was 66. For \(\chi = 0.07\) the corresponding figures were: 81% at 108 global iterations.
3.3. Analysis of elastoplastic frames with concentrated plasticity

3.3.1. Basic relations

Holonomic relations of a discrete elastoplastic system were formulated in the following form\(^{26}\):

\[
es = e^e + e^p, \quad e^e = Cs, \quad e^p = N\lambda,
\]
\[
e = Bq, \quad B^Ts = p, \quad \phi = N^T s - H\lambda - k,
\]
\[
\lambda \geq 0, \quad \phi \leq 0, \quad \phi^T \lambda = 0,
\]

where the following matrices, i.e. vectors (one-column matrix) and square matrix of
generalized variables are:

- \(e\), \(e^e\), \(e^p\) – total, elastic and plastic strain vectors;
- \(s\) – stress vector;
- \(C\) – elastic compliance matrix;
- \(N\) – gradient matrix of plastic potentials;
- \(B\) – geometrical compatibility matrix;
- \(q\), \(p\) – vectors of nodal displacements and forces;
- \(H\) – work-hardening matrix;
- \(k\) – vector of yield point stresses;
- \(\lambda\) – vector of plastic multipliers.

Under assumptions of Prager’s hardening rule the work-hardening matrix \(H\) is composed
of matrices \(H^i\), corresponding to finite elements \(i = 1, ..., n\):

\[
H = \text{dia}[H^1, ..., H^n], \quad H^i = H^iN_i^TN_i,
\]

and the plastic energy function, associated with relations (27), is of the form\(^{26}\):

\[
\Omega(\lambda) = \frac{1}{2} \lambda^T D \lambda - \left( N^T s^e - k \right)^T \lambda, \quad \lambda \geq 0,
\]

where:

\[
D = H - N^T Z N, \quad Z = E B K^{-1} B^T E - E
\]
\[
s^e = E B K^{-1} p, \quad K = B^T E B, \quad E = C^{-1}.
\]

3.3.2. Analysis of a plane frame

The energy function (29) can be used in the Panagiotopoulos approach discussed in
Section 2.2. The problem is formulated for elastoplastic plane frames\(^9\). The simplest finite
element is used with the bending moment \(M\) as the only generalized stress and plasticity
properties are concentrated at the ends \(L\) and \(R\) of the element, Fig. 11a.

Plastic potential of the considered FE is of the form:

\[
\phi^i_{\text{pl}} = N_i^T s^i - H^i \lambda^i - k^i,
\]

where the following matrices are used:
Figure 11: a) Frame finite element, b) Plastic properties of the frame cross-section, c) Analyzed plane frame

\[
\mathbf{N}_i = \text{diag}[\mathbf{N}, \mathbf{N}], \quad \mathbf{N} = [1, -1], \quad \mathbf{s}^i = \begin{bmatrix} \mathbf{s}_{IL}^i \\ \mathbf{s}_{IR}^i \end{bmatrix},
\]

\[
\mathbf{H}^i = \mathbf{H} \mathbf{N}_i^\top \mathbf{N}_i = \mathbf{H} \text{diag}[\mathbf{H}, \mathbf{H}], \quad \mathbf{H} = \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix},
\]

\[
(\mathbf{\lambda})^i = \begin{bmatrix} \mathbf{\lambda}_L^i \\ \mathbf{\lambda}_R^i \end{bmatrix} \quad \Rightarrow \quad (\mathbf{k})^i = M \begin{bmatrix} \mathbf{\lambda}_L^i \\ \mathbf{\lambda}_R^i \end{bmatrix} = \begin{bmatrix} \mathbf{s}_{IL}^i \\ \mathbf{s}_{IR}^i \end{bmatrix} = \begin{bmatrix} \mathbf{z} \mathbf{\theta} \end{bmatrix}.
\]

It is assumed that element \( i \) is of constant cross-section with the same plastic properties at the element ends \( L \) and \( R \).

The vector of rotations in plastic hinges \( \mathbf{\theta}^i \) and vector of total stresses \( \mathbf{s}^i \) are:

\[
\mathbf{\theta}^i = \begin{bmatrix} \mathbf{\theta}_{IL}^i \\ \mathbf{\theta}_{IR}^i \end{bmatrix} = \mathbf{N} \mathbf{\lambda}^i, \quad \mathbf{s}^i = (\mathbf{s}^e + \mathbf{s}^p), \quad \mathbf{s}^p = \begin{bmatrix} \mathbf{s}_{IL}^p \\ \mathbf{s}_{IR}^p \end{bmatrix} = \mathbf{z} \mathbf{\theta}.
\]

The frame is composed of eight finite elements \(^9\), i.e. \( i = 1, ..., 8 \), cf. Fig.11c. The columns were made of steel shapes HEB220 \((M_p = 200 \text{ kNm})\) and beams HEB180 \((M_p = 116 \text{ kNm})\). The following hardening moduli were assumed: \( H^1 = 340 \text{ kNm}, \ H^2 = 163 \text{ kNm}, \ H^j = 229 \text{ kNm} \) for \( j = 2, ..., 7 \).

The computation was performed first by a classical QP algorithm by Keller’s direct pivoting method \(^{27}\). Then the QP problem with unilateral constraints (12b) was analyzed for the energy form (29) integrating numerically the set of \( N = 32 \) differential equation (15), i.e. for variables \( \chi_i = \lambda_{il}^i, \ i = 1, ..., 32 \).

Computations were carried out for the fixed values of load factor \( \Lambda \). In Figs 12a,b the values of rotations \( \theta_{il}^i \) and total moments \( s_{ih}^i \) at the sections \( iL \) and \( iR \) are depicted for the load factors \( \Lambda = 1.402, 1.583 \). The first value \( \Lambda = 1.402 \) is the one for which the frame would
collapse if the hardening moduli are zero $H^i = 0$, i.e. for the elastic perfect plastic material. The second value $\Lambda = 1.583$ corresponds to the first exceeding of the hardening rotational capacity $\theta^i_{\text{H}}$, cf. Fig.11b, at the last one of the plastic hinge. In the considered case this capacity was reached at the node $7R$ for $\theta^7_{\text{H}} = 0.144$.

Figure 12: a) Rotations – nodes diagram, b) Total moments – nodes diagram
The Runge-Kutta 4-th order method with step $\Delta t = 0.5$ was used for the integration of Eq. (15). Several numerical experiments led to the conclusion that the neural approach is very insensitive to changes of the initial conditions. Moreover, the neural algorithm needed 300 steps which after the 30th (resp. $\Lambda = 1.583$) reached almost the same minimum value of the network energy with small variations only. In Table 2 values of total moments $s_{ih}$ are given at nodes $iL$ and $iR$ of elements $i = 1,...,8$.

<table>
<thead>
<tr>
<th>Nodes/L.F.</th>
<th>1.402</th>
<th>1.410</th>
<th>1.415</th>
<th>1.420</th>
<th>1.425</th>
<th>1.430</th>
<th>1.583</th>
<th>Mp</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1L</td>
<td>85.239</td>
<td>84.841</td>
<td>84.592</td>
<td>84.343</td>
<td>84.093</td>
<td>83.844</td>
<td>80.833</td>
<td>200</td>
</tr>
<tr>
<td>1R</td>
<td>-123.588</td>
<td>-123.836</td>
<td>-123.990</td>
<td>-124.145</td>
<td>-124.300</td>
<td>-124.454</td>
<td>-124.707</td>
<td>200</td>
</tr>
<tr>
<td>2L</td>
<td>-123.588</td>
<td>-123.835</td>
<td>-123.990</td>
<td>-124.145</td>
<td>-124.300</td>
<td>-124.454</td>
<td>-124.704</td>
<td>116</td>
</tr>
<tr>
<td>3R</td>
<td>98.168</td>
<td>99.637</td>
<td>100.555</td>
<td>101.473</td>
<td>102.391</td>
<td>103.309</td>
<td>121.399</td>
<td>116</td>
</tr>
<tr>
<td>4L</td>
<td>98.168</td>
<td>99.637</td>
<td>100.555</td>
<td>101.473</td>
<td>102.391</td>
<td>103.309</td>
<td>121.399</td>
<td>116</td>
</tr>
<tr>
<td>4R</td>
<td>55.138</td>
<td>56.512</td>
<td>57.371</td>
<td>58.230</td>
<td>59.088</td>
<td>59.947</td>
<td>74.729</td>
<td>116</td>
</tr>
<tr>
<td>5L</td>
<td>55.138</td>
<td>56.512</td>
<td>57.371</td>
<td>58.230</td>
<td>59.089</td>
<td>59.947</td>
<td>74.729</td>
<td>116</td>
</tr>
<tr>
<td>5R</td>
<td>96.119</td>
<td>97.541</td>
<td>98.430</td>
<td>99.319</td>
<td>100.208</td>
<td>101.096</td>
<td>117.835</td>
<td>116</td>
</tr>
<tr>
<td>6L</td>
<td>96.119</td>
<td>97.541</td>
<td>98.430</td>
<td>99.319</td>
<td>100.208</td>
<td>101.096</td>
<td>117.835</td>
<td>116</td>
</tr>
<tr>
<td>8R</td>
<td>203.223</td>
<td>203.358</td>
<td>203.442</td>
<td>203.526</td>
<td>203.611</td>
<td>203.695</td>
<td>209.049</td>
<td>200</td>
</tr>
</tbody>
</table>

Table 2: Total moments $s_{ih}$ and $s_{jh}$ for different values of load factor $\Lambda$

### 3.4. Analysis of elastoplastic plane stress problem with unilateral boundary constraints

#### 3.4.1. Basic relations

Weights $w_{ij}$ and biases $b_i$ of the HT neural network can be specified as components of the matrix $A$ and vector $b$ in (12), associated with physical problems. In papers by Panagiotopoulos et. al. $7,8$ it was shown that this analogy can be explored in the analysis of problems formulated by means of the Finite Element Method.

Using displacement version of FEM the evolutionary set of HTNN equations (15) takes the form

$$
\frac{dv}{dt} = -KQ + P \equiv Z.
$$

where instead of matrix $A$ and vectors $b$ and $x$ the stiffness matrix $K$ and the following vectors are substituted: $Q, P, Z = P - F \in \mathbb{R}^n$ - vectors of nodal displacements, loads and residual forces, respectively.
The asymptotically stable equilibrium state of HTNN corresponds to the equilibrium state of FEM system:

\[
\frac{\text{d}v}{\text{d}t} = Z \rightarrow 0 \quad \Rightarrow \quad KQ = P.
\] (35)

This means that instead of analysis of the algebraic problem (35) the initial value problem (34) is considered, corresponding to the simple gradient method of solution of the set of algebraic equation. In case of displacements \( q_1 \) treated as bilateral constraints the iteration process of the simple gradient method in numerically inefficient because of slow convergence. This is not the case of bilateral constraints which can considerably accelerate the iteration by Panagiotopoulos et al. 7,8.

In papers 7,8 a number of problems from linear elasticity was analyzed. Eliminating the bilateral displacements, the QP problem (12b) with only bilateral constraints was considered:

\[
\min \left( \frac{1}{2} q^T kq - p^T q \mid q \geq 0 \right),
\] (36)

where: \( p, q \in \mathbb{R}^N \) – vectors of nodal and unilateral displacements, \( k \in \mathbb{R}^{N \times N} \) – condensed stiffness matrix.

More general constraints can also be considered 28. In the quoted paper the simple gradient method of the analysis of Eq. (15) was change into the conjugate gradient method:

\[
\frac{\text{d}v}{\text{d}t} = \beta_t v_{t-1} \quad \text{for} \quad \beta_t = \frac{x^T_t - x^T_{t-1}}{x^T_{t-1} - x^T_{t-1}},
\] (37)

where: \( t \) – actual time (iteration step), \( t-1 \) – previous time. In 28 the 5th and 6th order Runge-Kutta-Fehlenberg method 29 was used for numerical integration of Eqs (37) (in papers by Panagiotopoulos and associateds 7,8, Runge-Kutta 4th order method was used).

The analysis of elastoplastic problems by HTNN is much more complex since the network parameters, i.e. matrices \( K \) and \( P \) in (35), are influenced by the yielding process 24. Instead of Eq. (35) the incremental FE equation is used:

\[
K_T \Delta Q = \Delta \Lambda P^* + Z,
\] (38)

where: \( K_T \) – tangent stiffness matrix, \( \Delta \Lambda \) - increment of load factor, \( P^* \) - vector of reference loads.

Eq. (38) is solved iteratively using the load factor increments (steps) \( \Delta_m \Lambda \). A sequence of linearised FE equations 24 is analyzed:

\[
m \left. K^{(i-1)} \right. \Delta Q^{(i)} = \left. \Delta \Lambda \right. P^{*} + \left. (1+\alpha^{(i)}) \right. Z^{(i-1)}, \] (39)

where: \( m \left. K^{(i-1)} \right. = K_T (m \left. Q^{(i-1)} \right.), \quad m \left. Z^{(i-1)} \right. = Z (m \left. Q^{(i-1)} \right.) \) – tangent stiffness matrix and vector of residual forces, updated for the vector of total displacements:
\[ m Q^{(n-1)} = m Q + \sum_{k=1}^{n-1} \Delta Q^{(k)} \quad , \] (40)

and parameter \( \alpha^{(\alpha)} \) is:

\[
\alpha^{(\alpha)} = \begin{cases} 
1 & \text{for predictor at } \; it = 1, \\
0 & \text{for corrector at } \; it > 0.
\end{cases}
\] (41)

After condensation of the bilateral displacements the vector of residuals in (37) takes the form:

\[
r_t = -Ax_t + b = -k(m_q^{(\alpha-1)})\Delta q_t + \alpha^{(\alpha)}\Delta m_p^* + (1-\alpha^{(\alpha)})(m_q^* - f(m_q^{(\alpha-1)})).
\] (42)

The values of matrix \( A \) and vector \( b \) components are fixed in Eq. (37), and they are updated for the next load level \( m \) or iteration step \( it \). In case of the modified Newton-Raphson method matrix \( k \) is updated only once per each \( m \), i.e. \( k^{(\alpha)} = k(m_q^{(1)}) \).

### 3.4.2. Tension strip with a contact zone

The approach discussed above was used for the analysis of a perforated tension strip shown in Fig. 9.

Computations were performed for the tension of a strip with a rigid bolt of radius \( R = 5 \) mm which fit exactly to the circular hole inside the strip. It was assumed that a smooth, i.e. frictionless contact exists between the bolt and hole boundary. Because of the curve boundary the constrained condition was formulated in the form corresponding to the node \( i \) and displacement vector \( u \), cf. Fig. 13a:

\[ \Delta \mathbf{u}_t = \begin{cases} \mathbf{v}_t & \text{f.o.: } |\mathbf{x}_t| \geq R, \\
\Delta \mathbf{u}_t & \text{f.o.: } |\mathbf{x}_t| < R, \end{cases} \] (43)

where:

\[ \mathbf{x}_t = \mathbf{x}_o + \mathbf{u}_{t-1} + \mathbf{v}_t \quad , \quad \Delta \mathbf{u}_t = \frac{R}{|\mathbf{x}_t|} \mathbf{x}_t - \mathbf{x}_o - \mathbf{u}_{t-1}. \] (44)

The make it easier to follow the node index \( i \) and iteration superscript \( it \) were omitted.

Computations were carried out by means of the ANKA-H program, discussed in Point 3.2.3, supplemented by numerical procedures for static condensation of FE system and for numerical integration of the HTNN set of Eqs (37). The condensation was performed for 19 boundary nodes Nos 1-181, shown in Fig. 9c. This means that the reduced system has \( n = 2 \times 19 - 1 = 37 \) DOF.

In Fig. 13b the relation “load factor – horizontal deflection of the point A” (node No. 181) \( \Lambda (\mathbf{u}_A) \) is shown up to \( \Lambda = 2.0 \). This equilibrium path is compared with the relation \( \Lambda - \mathbf{u}_A \) for a perforated strip without rigid bolt. It is clear that the strip with bolt is stiffer than...
the strip with the contact-free hole. After unloading the parameter displacement of the strip with bolt \( u^*_A = 0.72 \) mm is about 15% smaller than for the strip with rigid inclusion.

In Figs. 14a,b the deformed strips are shown. As clearly seen, the contact takes place only along four FEs. In Figs. 15 c,d the yielding zones (separated by the counters of effective stress \( \sigma_e = 1.0 \)) are shown. The contact of the strip and bolt influences the shape and areas of the yielding zones.

### 3.5 Parameter identification problem

#### 3.5.1. Using HTNN to parameter identification

As it mentioned in Section the HTNN parameters can reflect features of mechanical structures, i.e. stiffness matrix \( K \) of FE systems, and the input \( x \) is associated with system response. In direct analysis the HTNN analogy enables us to simulate the response; in inverse analysis HTNN can be used for the system parameter identification. The analyses are usually based on a part of data related to \( K \) and \( x \), exploring formulations associated with both bilateral and unilateral constraints. For instance the Young modulus \( E \) and Poisson’s ratio \( \nu \) in an elastic body were identified on the basis of known values of displacements at 12 points of a FE mesh in the vicinity of an open crack 8.

For identification analysis the supervised learning of HPNN is used. The weights \( w_{ij} \) (biases \( b_i \) are treated as weights \( w_{i0} \) for unit input \( x_0 = 1 \)) are computed by means of the iterative formula:
Figure 14: Deformations of strips at $\Lambda = 2.0$: a) Strip with a free hole boundary; b) Strip with a rigid bolt; Distribution of effective stresses $\sigma^e$; c) Strip with a free hole, d) Strip with a bolt
where: \( \eta(\cdot) > 0 \) – adaptive-type learning rate, \( r_j(k) \) – reinforcement signals, \( k \) – iteration step.

A very simple Widrow-Hoff learning rule can be used:

\[
\begin{align*}
    x_i^*(k+1) &= x_i^*(k) + \eta(\cdot) r_i(k), \\
    x_i(k+1) &= x_i(k) + \sum_{j=0}^{N} v_{ij}(k) x_j(k),
\end{align*}
\]

where: \( x_i^* \) – desired response. Solutions \( x_i(\cdot) \) are used as initial values for HTNN equations (15) to compute values \( x_i(k+1) \). This iteration procedure associated with the improvement to weights \( w_{ij} \) and correction of responses \( x_i \) is continued until the desired results \( \{ x_i^* \} \) are obtained.

### 3.5.2. Identification of load factor for an elastoplastic frame

Identification problem \(^9\) is reduced to the computation of load factor \( \Lambda \) for the frame analyzed in Point 3.3.2. According to Eq. (29) the vector of plastic multipliers \( \Lambda \) is assumed to be a response variable. The accuracy of identification is estimated by the total moment vector \( s \):

\[
    s = \Lambda E B K^{-1} p^* + Z N \lambda,
\]

in which notation is taken from Point 3.3.1.

In order to avoid lengthy computations a direct method was applied to compute the load factor values \( \Lambda = 1.41 \) and 1.43. The corresponding values of total moments are given in columns (3) and (7) of Table 2. Then basing on (45-46) the iterative procedure could start taking initial values of the response vector \( \Lambda \) from linear interpolation. As a result of supervised learning the identified load parameters and corresponding nodal moments are shown in columns (4-6) of Table 2. The values of those moments have, up to the third digit, the same values as the moments computed by direct method.

### 3.5.3. Parameter identification by interaction of two neural networks

In many identification problems the identified vector \( s^{(r)} \) (usually defined in the state space) is related to the control vector \( z^{(r)} \), where: \( r = 1, \ldots, T \) – instants of observation. Then the identification problem can be formulated in the following form \(^{10,30}\):

\[
\begin{align*}
    N 1) & \quad \sum_{r=1}^{T} \left\| s_r^{(r)} - s_r(z^{(r)}) \right\|, \\
    N 2) & \quad K(z^{(r)}) q = p(z^{(r)}).
\end{align*}
\]
Of course, instead of (49) the corresponding QP problem (12) can be introduced.

Applying the N approach two neural networks can be used. The BPNN type network \( N_1 \), associated with (48), is used for the computation of control vector \( \mathbf{z}(\tau) \). The HTNN analog, called network \( N_2 \), is applied to the analysis of Eq. (49). It is clear that in case of bilateral constraints a FE program can be efficiently used for the analysis of Eq. (49). The computations are to be repeated for each \( \tau = 1, \ldots, T \) and for the sake of simplicity the superscript \( \tau \) is omitted in what follows.

The weights of network \( N_1 \) are computed by the following formula:

\[
v_{i,j}(k+1) = v_{i,j}(k) + \eta \sum_{l=1}^{n} v_{i,l}(k) s_{l} - s_{j}(k), \quad \eta > 0.
\]  

(50)

The control parameters \( z_v \) are computed in a similar way:

\[
z_v(k+1) = z_v(k) + \mu_z \sum_{q=1}^{Q} s_{q}^* - s_{q}(k) + \mu_v \sum_{l=1}^{n} v_{l}(k) s_{l}(k), \quad \mu_v > 0.
\]  

(51)

In formula (51) summing \( \sum_{q} \) means that the learning term is extended over all the parameters related to the \( v \)-th value of control parameters \( z_v \).

After control parameters \( \{z_v(k+1)\} \) have been computed for iteration step \( (k + 1) \) the matrix \( K(z(k+1)) \) and vector \( p(z(k+1)) \) can be calculated and new values of vector \( q(k+1) \) can be computed by means of Eqs (49). Then the state vectors \( s_{x}(q(k+1)) \) are computed. The computation is continued until the convergence criterion, related to (48), is satisfied:

\[
\max_{r,x} \| \mathbf{s}_{x}^{\tau(r)} - \mathbf{s}_{x}^{\tau(r)} \| < \delta,
\]

(52)

where: \( \delta \) – prescribed admissible error.

**3.5.4. Tracing yield surfaces**

The experiments by Shiratori and Ikegami were taken as a base for computing yield surfaces. The experiments were carried out on a cross-shaped specimen shown in Fig. 15a. The specimen of thickness \( h = 1 \text{ mm} \) was subjected to biaxial tension along proportional and combined loading paths, cf. Fig. 15b.

For numerical analysis it was assumed that the considered square of specimen of area \((100 \times 100) \text{ mm}^2\) is made of elastic material with four orthotropy coefficients. The coefficients are treated as components of control vector

\[
\mathbf{z} = \{z_v\} = \{\alpha_{11}, \alpha_{12}, \alpha_{12} = \alpha_{211}, \alpha_{33}\} = \mathbf{c}.
\]  

(53)

An intermediate yield surface was computed between the experimental yield surfaces which assumed to be elliptic paraboloid surfaces. This was done to get more reliable appro-
ximation of the anisotropic elastoplastic problem. After such a “rough” prediction neural networks N1 and N2 were used to solve the parameter identification problem, discussed in Point 3.5.3. The predictions of the yield surface position vector $\sigma_{pl}$ were corrected by the updating formula:

$$\sigma_{pl}(k+1) = \sigma_{pl}(k) + \eta(k) \sum (z(\sigma_{pl}(4)) - z^*) ,$$

where: $z(\sigma_{pl}(k)) = C(k)$ – vector of elastic coefficient computed by network N1, $z^* = C^*$ – vector of elastic coefficient computed on the basis of experimental results, $\Sigma$ – extension of formula (50) over all the FEs of the specimen square, $\eta(k)$ – a scale factor representing the learning rate.

In Fig.16b curves A, C, D, E, F are curves approximating experimental results according to stress points shown in Fig.15b. On the basis of these yield surfaces internal surface $D'$ and external surface $G$ are computed applying the algorithm sketched above. A slight nonconvexity (e.g. parts 1-2 and 1’–2’ of curve $G$) is observed at the computed yield surfaces. The larger the stresses the more visible is this lack of convexity. A more detailed discussion corresponding to other loading paths is given in paper 10.

4. OTHER FIELDS OF ANN APPLICATIONS IN PLASTICITY

4.1. Load identification problems

In papers 12,32 a simple supported beam, made of elastic perfect plastic material was considered. The problem was related to identification of three parameters of statically applied load: 1) resultant $Q$, 2) load centre location $l_Q$, 3) load width application $w_q$, cf. Fig.17a.
Figure 16: a) Experimental points at subsequent yield surfaces for prestressing $\frac{\sigma_y}{\sigma_x} = 1$, b) Subsequent yield surfaces by neural prediction

As input data increment of the first six eigenvalues were used and output corresponded to the dimensionless load parameters:

$$x_{i,\text{col}} = \{(1 - f_i / f_0) \mid j = 1,...,6\}, \quad y_{i,\text{col}} = \left\{ \frac{l_Q}{\bar{Q}} = \frac{Q}{Q_{\text{ub}}} \right\}, \quad \bar{w} = w/l \right\}, \quad (55)$$

Figure 17: a) Data of considered beam and load parameters, b) One-level BPNN, c) Three levels, cascade BPNN where: $f_j$ – $j$-th frequency of yielded beam, $f_0$ – reference frequency of elastic beam.

The patterns were computed by the ADINA FE code \(^{33}\). The best results were obtained for the identification of load centre $\bar{T}_q$. For the network shown in Fig. 17b, the following
linear correlation ratios were: \( r_L = 0.980 \) and \( r_T = 0.972 \) for learning and testing respectively (the ratio \( r \) correlates values of \( T_q \) neurally predicted and computed by ADINA values of \( T_q \)). Using the same BPNN: 6-10-3 the corresponding results were obtained for the width parameter \( w_q \), identification: \( r_L = 0.990 \) and \( r_T = 0.870 \).

The neural prediction was improved when instead of the one-level BPNN the three levels, cascade BPNN, shown in Fig. 17c were formulated \(^{32,34}\). The appropriate values of the correlation ratios were: 1) \( r_L = 0.992 \) and \( r_T = 0.990 \) for BPNN: 6-10-1 used for the identification of \( T_q \), 2) \( r_L = 0.931 \) and \( r_T = 0.899 \) for BPNN: 8-10-1 applied to the identification of load width \( w_q \).

### 4.2. BPNNs in damage identification using data of ultrasonic wave propagation

Ultrasonic wave techniques were developed as ones of nondestructive methods for damage evaluation. Neural networks can be used for the recognition of time diagram changes caused by cracks located in elastic solids \(^{35}\). This approach was also used for identification of a zone in which the stiffness modulus \( E_i \) is degraded \(^{13}\). The wave propagation diagrams measured at \( k \) points of a deformable bar (cf. Fig. 18b) were discretised into \( N \) values. Each of the sets of \( N \) values was compressed into \( n << N \) values by the BPNN network called replicator, cf. Fig. 18c. The master BPNN with \( k \times n \) inputs was used for the identification of \( M \) parameters. Besides the one-level BPNN of \( M \) outputs also cascade BPNNs were applied for the identification of \( M = 3 \) damage parameters \(^{13}\). Quite recently this approach has been explored to identify the location of a zone in which a prescribed value of the effective stress was exceeded \(^{36}\). Such results have been seen promising for the damage analysis in elastoplastic bodies.

![Figure 18](image_url)

Figure 18: a) A bar with stiffness defect, b) Wave propagation and changes caused by defect, c) Replicator for signal compression and a network for damage identification
4.3. BPNN in reliability analysis of elastoplastic structures

BPNNs were efficiently used for computation of patterns in Monte Carlo methods\textsuperscript{14}. The reliability analysis of a complex elastoplastic, plane frame was significantly accelerated due to neural computation of patterns. The network was trained on 20-60 examples, computed by a FE program. Due to generalization properties of the trained network up to $10^5$ patterns were generated for 2-4 random variables. The neural approach gave solutions exploring the processing time $10^{-3} – 10^{-4}$ shorter than the time used is standard Monte Carlo methods.

4.4. Application of other types of ANNs

Unfortunately the BPNN network has some disadvantages. They are associated with formulation of BPNN architecture (number of layers and neurons in them), selection of activation function and laborious training. Other disadvantages correspond to global approximation with respect to different relation between input/output variables. That is why other of ANNs should also be introduced in the analysis of plasticity problems.

For instance the Counter-Propagation NN (CPNN) has been used for the inversion of elastoplastic relation $M(\varepsilon, \kappa)$ and $N(\varepsilon, \kappa)$ into $\varepsilon(N, M)$ and $\kappa(N, M)$ which were needed in the Finite Difference analysis of plane framed structures\textsuperscript{37}. CPNN completed by the interpolation layer appeared to be much more efficient numerically than BPNN even in case of beam bending\textsuperscript{38}.

Radial Basis Function (RBF) neural networks are very attractive. These networks have only one hidden layer with RBFs which enable us to apply the local approximation in the space of input variables. After good selection of control parameters the number of training epochs is for RBF networks significantly smaller than for BPNNs. The RBF network processing is in fact very similar to processing by ANFIS (Adaptive Neural-Fuzzy Inference System)\textsuperscript{39}.

RBF and ANFIS were networks successfully applied to simulation of responses and parameters identification for concrete specimens of plastic and brittle properties\textsuperscript{15,16,40}.

5. FINAL REMARKS

The examples discussed in Section 3 and 4 enable us to formulate the following conclusions which can be treated as new prospects of the neurocomputing applications in the analysis of plasticity problems:

1. ANNs open the door for implicit modelling of constitutive equations on the base of observable data without model parameters.

2. Neural procedures formulated as ANNs trained off line can be incorporated into standard computational programs, e.g. into the FE codes. The corresponding hybrid programs can be numerically much more efficient than the pure computational programs are.
3. The Hopfield-Tank neural analog combined with the Quadratic Programming formulation and the FE program for the network learning (the Panagiotopoulos approach) can be used for the analysis of problems with unilateral constraints.

4. ANNs can be efficiently used not only for simulation, i.e. direct analysis but also to parameter identification, i.e. inverse analysis of plasticity problems. Interaction of various types of neural networks makes it possible to formulate new algorithms for the analysis of simulation and identification problems.

5. From among other problems, discussed in Section 4, the application of ANNs to nondestructive damage analysis seems worth developing.

6. Besides the BPNN and HTNN networks, discussed more extensively in the paper, also other networks and especially fuzzy-networks, should be widely introduced in plasticity.

The majority of results discussed in the paper were taken from research projects which have been developed at the Cracow and Rzeszów Universities by the author and his associates.

This is why the author would like to express his deep gratitude to all his coworkers and especially to Dr E. Pabisek from TU Cracow and Prof. L. Ziemiański from TU Rzeszów for their enthusiasm and activity on applications of neurocomputing in mechanics of solids and structures.

REFERENCES


L. Ziemiański, “Neural analysis of damage on the basis of wave propagation data”, (2000), (private communication)


