A FAST SELF-LEARNING PROCEDURE FOR SEISMIC VULNERABILITY ASSESSMENT: APPLICATION TO CHEMICAL PLANT STRUCTURES

Aoki T.\(^2\), Ceravolo R.\(^1\), De Stefano A.\(^1\), Genovese C.\(^1\), Sabia D.\(^1\)

\(^1\)Dip. Ingegneria Strutturale e Geotecnica, Politecnico di Torino, Torino, Italy
\(^2\)School of Design and Architecture, Nagoya City University, Nagoya, Japan

ABSTRACT

A chemical industrial plant represents a sensitive presence in a region and, in case of severe damage due to earthquake actions, its impact on social life and environment can be devastating. From the structural point of view, chemical plants count a number of recurrent elements, which are classifiable in a discrete set of typological families (towers, chimneys, cylindrical or spherical or prismatic tanks, pipes etc.). Within a single typology, each structure may differ from the others for its geometry, mechanical properties of the materials or boundary conditions. The final aim of this work is to outline a general procedure to be followed in order to assign a seismic vulnerability estimate to each element of the various typological families.

In this paper, F.E. simulations allowed to create a training set, which has been used to train a probabilistic neural system. In practical applications, the database may be integrated by real measurements and observations. This network is expected to return a vulnerability estimate for chemical plant structures and predict the type of collapse that presumably would affect the structure in case of strong earthquake. A sample application has concerned the seismic vulnerability of simple spherical tanks.

1 INTRODUCTION

Artificial neural networks are data processing systems possessing learning and generalisation capabilities. In a supervised neural network, training consists of presenting a set of examples and letting the network build up, on the basis of a well-defined algorithm, the interior structure it needs to perform its intended task. In mathematical terms this amounts to supplying a set of vector pairs, i.e. an input vector (X) and an output vector (Y), and having the network associate all the X - Y pairs, with the least possible error, so as to build up a unique functional relationship Y = F(X), no matter how complex is the phenomenon to be interpreted. Having concluded the training stage, the network must be able to effectively predict the output vector Y most closely associated with the input vector X.

In classical neural networks, the answer is univocal, and vector Y is the one involving the greatest possibility of occurrence. Probabilistic networks, on the other hand, supply several answers Y\(_i\), each of them associated with an estimate of the respective probability. In the case being considered, the probabilistic network yields a Bayesian type classification.
The final goal of this paper is to exploit the fundamental learning and generalisation capabilities of neural networks to estimate the vulnerability of structural elements typical of chemical plants and, more specifically, predict the damage mechanisms which may likely occur in any specific typology.

2 BAYESIAN CLASSIFICATION

Bayes’ theorem is also known as theorem of the probability of causes, in that it makes it possible to determine the probability of events \( A_i \) having been the cause of the occurrence of event \( B \). Bayes’ theorem is used to convert an “a priori” probability estimate \( P(A_i) \) into an “a posteriori” probability estimate \( P(A_i|B) \) knowing the conditional probability \( P(B|A_i) \): this theorem enables you to determine the conditional probability of event \( A \) given that event \( B \) has occurred. In the vulnerability context, we want to classify a set of m-dimensional vectors \( X =\{x_1,x_2,...,x_m\}^T \) each one representing the typological characteristics of a building, into a discrete number of classes \( \omega_1, \omega_2, ... , \omega_K \) representing the different damage mechanisms or different safety factor values. In this case (discrete parameter) Bayes’ theorem asserts that the probability mass function of \( \omega_i \) for a given vector \( X \) is:

\[
\hat{h}(\omega_i|X) = \left[ \frac{p(X|\omega_i)g(\omega_i)}{\sum_{j=1}^{K} p(X|\omega_j)g(\omega_j)} \right]
\]

- \( g(\omega_i) \) denotes the “a priori” probability density since it is determined prior to observing vector \( X \) in the current experiment (based upon previous understanding)
- \( p(X|\omega_i) \) denotes the probability density function of \( X \), conditional upon the class \( \omega_i \)
- \( \hat{h}(\omega_i|X) \) is called the “a posteriori” probability mass function of \( \omega_i \), given \( X \), since it is determined posterior to observing the current set of data.

The equation (1) estimates the “a posteriori” probability density that vector \( X \) belongs to class \( \omega_i \). The key to using Bayes’ theorem for classification problems is the ability to estimate \( p(X|\omega_i) \). In this context we adopt “a priori” probability densities to be equally likely and have to estimate the different \( p(X|\omega_i) \) by means of a probabilistic network based on training patterns because all that is given is a set of training samples (m-dimensional vectors \( Y \) obtained from standard vulnerability data forms) which provide the only clue to the unknown probability densities.

3 THE PROBABILISTIC NETWORK

The estimate of the probability density is obtained through the non-parametric core estimators technique, which supplies continuous estimates at all times and is ideally suited to deal with small sized samples [1,2]. The estimate \( \hat{p}(u) \) of a density \( p(u) \) based on the observation of \( N \) samples \( Y_i \) is given by the following expression:

\[
\hat{p}(u) = \frac{1}{N} \sum_{n=1}^{N} h_\alpha(u - Y_i)
\]

(2)

where \( h_\alpha(u) \) is a kernel function defined as probability core. Index \( \alpha \) shows that the core function depends on a parameter which must be made to vary with \( N \) to provide the estimator with suitable asymptotic properties. The probability core function used is:
where $C(m, \alpha)$ is a normalisation constant depending on dimension $m$ of the training vectors and on parameter $\alpha$. If the parameter $\alpha$ is selected so that:

$$\lim_{N \to \infty} \frac{N}{\alpha} = \infty \quad \text{and} \quad \lim_{N \to \infty} \alpha = \infty$$

then it can be demonstrated that the estimator $\hat{p}(u)$ simply converges via quadratic means towards $p(u)$ at all points where the latter is continuous [2]. A convenient choice can be $\alpha = \sqrt{N}$, where $N$ is the number of samples available.

### 3.1 Network architecture

The vector $X$ contains the input variable that influence the collapse mechanism of the structure and is associated with the class $\omega_i$ representing the specific damage mechanism suffered by the structure itself. The complete network producing the classification into $K$ classes of mechanisms is made up of $K$ identical subnetworks (figure 1) in which training and classification stages are conducted at different times. The supervised training of each of these subnetworks, each of which therefore represents a specific damage formation mechanism, is performed separately by selecting the network corresponding to the class being observed.

![Network Architecture](image)

**Fig. 1** Network architecture

### 3.2 Subnetwork training

The domain $D$ of $m$-dimensional vectors $Y_i^{(i)}$, $Y_2^{(i)}$, $\ldots$, $Y_N^{(i)}$, belonging to the class $\omega_i$, is represented by $P$ nodes or neurons each one is characterised by a $m$-dimensional vector $W_{p,i}$ (with $p=1, \ldots, P$), referred to as centroid later on, because it represents the “gravity centre” of the points of the hyperspace (input vectors) attracted towards a specific node, and by a scalar coefficient $a_{p,i}$, which measures the activation’s level of the node itself (Figure 2).

As a first stage of the training, a vector $W_{p,i}$ is associated to each node of the subnetwork; meanwhile, the coefficients $a_{p,i}$ are taken to be 1. At the beginning, the centroids $W_{p,i}$ are defined using a random generation procedure or using a number $P$ of training vectors that unequivocally represent each type class.

At the first layer of a subnetwork $i$ (figure 2) we find the so-called squaring cell; for each training vector $Y_i$ of size $m$ that is presented at the input, a vector $\overline{Y_i}$ of size $m+2$ is given at the output, defined as:

$$\overline{Y_i} = \left[\|Y_i\|^2, Y_i, -1/2\right]^T$$

Then, the information about this vector is transmitted to the $P$ neural cells where the same transformation is made for each centroid:

$$\overline{W}_{p,i} = \left[-1/2, W_{p,i}, \|W_{p,i}\|\right]^T$$
and equations (5, 6) allow us to determine the Euclidean distance of \( Y_i \) from each centroid as:

\[
W_{p,i}^T Y_i = \left\| Y_i - W_{p,i} \right\|^2 / 2
\]

(7)

![Subnetwork architecture](image)

This quantity is assigned to each of the \( P \) neurons belonging to the subnetwork which, in turn, through the transfer function defined by the sigmoid function, supply at the output the quantities:

\[
S_{p,i} = f_\alpha\left(-\left\| Y_i - W_{p,i} \right\|^2 / 2\right) = \left[1 + \exp\left(\alpha \left\| Y_i - W_{p,i} \right\|^2 / 2\right)\right]^{-1}
\]

(8)

The vector \( Y_i \) is assigned to the neuron \( p_{0,i} \) that produces the maximum answer \( S_{p,i} \), so that the new centroid-vector \( W_{p,i}^{(\text{new})} \) and respectively the updated coefficient \( a_{p_{0,i}}^{(\text{new})} \), become:

\[
W_{p,i}^{(\text{new})} = \frac{a_{p_{0,i}} W_{p_{0,i}} + Y_i}{a_{p_{0,i}} + 1} ; \quad a_{p_{0,i}}^{(\text{new})} = a_{p_{0,i}} + 1
\]

(9)

Thus, for each time that a training vector is assigned to one of the \( P \) neurons, its activation coefficient increases and its centroid-vector is replaced, according to equation (9), by a new centroid calculated as a weighted average between the older one and the new vector just introduced. The automatic grouping of all the training vectors belonging to a specific subnetwork is carried out according to the evolution’s rule of the centroids and can be improved by using the same training vectors several times (iterations) using different presentation orders. The learning stage is considered completed when the evolution of the centroids (that is the variation of their position) is not significant. The number of iterations, depending on the number of neurons adopted and on the choice for the initial centroids, may be very high; in that case it is worthwhile to limit the required computer’s memory by inserting a real number \( \eta \), selected in the \([0,1]\) range, in the equation (9) for the activation coefficients:

\[
a_{p_{0,i}}^{(\text{new})} = \eta\left(a_{p0} + 1\right)
\]

(10)

3.3 Classification stage

At the classification stage the centroids \( W_{p,i} \) are fixed. The outputs of the \( P \) neurons, for a given observation vector \( X \), are weighted by the \( a_{p,i} \) coefficients and then are added up as shown in figure 2. At this point a normalisation procedure is applied in order to ensure that the estimate supplied actually corresponds to a probability density. The expression of the estimated probability density of a vector \( X \) conditional upon the class \( \omega_i \) is:
The normalisation factor $A(i)$ takes into account the actual totals present for the construction of the density relating to class $\omega_i$. This procedure, repeated for all the $K$ classes (Fig. 1) allows us to classify an observation vector $X$, representing a given building, into one of the $K$ subnetworks representing the different classes.

3.4 Application of the method to forecast the seismic damage mechanisms on churches

The procedure illustrated above has been applied, with an excellent performance, to the Italian seismic vulnerability data-bases built-up by GNDT (Gruppo Nazionale per la Difesa dai Terremoti), with two different aims: to estimate the vulnerability level of standard traditional housing; to forecast rapidly the damage mechanisms on monumental churches [3-4]. The second kind of application is here shortly reported as an example. The set of churches here considered comes from the census taken by for about 100 churches damaged by the earthquake that hit Emilia Romagna in 1987 and about 250 churches damaged by the Friuli earthquake in 1976. To characterise a church from the structural viewpoint and analyse its damage, we have used the concept of macro-element and damage mechanism [5]. For each macro-element, elementary structural types are identified so that complex types can be described through the sum of the elementary ones. The damage mechanism represents a schematic reconstruction of the movements exhibited by the parts of the macro-elements and the relative displacements. They are combinations of elementary crack patterns to be identified. The elementary damage mechanisms identified for the facade are shown in Fig. 3.

The parameters taken into account to build training vectors are listed in table 1. The damage classes represent the different damage mechanisms for each macro-element. In the training stage each set of information (see table 1), is associated with a damage class, in order to constitute a vector to be sent to the sub-network associated with the chosen damage class.

<table>
<thead>
<tr>
<th>Table 1: Typological and geometrical parameter vectors</th>
</tr>
</thead>
<tbody>
<tr>
<td>Structural type of main body</td>
</tr>
<tr>
<td>Structural type of collapse in terms of macro-element</td>
</tr>
<tr>
<td>Structural type of transept and spire</td>
</tr>
<tr>
<td>Types of other buildings annex to the church (sacristies, porches)</td>
</tr>
<tr>
<td>Maximum width of the church</td>
</tr>
<tr>
<td>Maximum length and height of the church</td>
</tr>
</tbody>
</table>

During the classification stage each set of information is sent to every sub-network, leading to the estimation of the belonging probability to each damage class. A validation test for the classification stage has been performed on ten churches extracted from the same

$$p(X|\omega_i) = \frac{1}{A(i)} \sum_{p=1}^{P} a_{p,i} f_{\alpha}(\|X-W_p\|^2/2) \quad \text{where:} \quad A(i) = \sum_{p=1}^{P} a_{p,i} \quad (11)$$

The normalisation factor $A(i)$ takes into account the actual totals present for the construction of the density relating to class $\omega_i$. This procedure, repeated for all the $K$ classes (Fig. 1) allows us to classify an observation vector $X$, representing a given building, into one of the $K$ subnetworks representing the different classes.
database and not included in the training set. The forecasting resulted correct in all cases, with only some ambiguous outcomes, generally connected with real possibility of different mechanisms. Two examples are presented in the following.

- **Church of S. Maria Assunta in Scandiano**
The damage mechanism actually triggered by the quake was No. 4 (Translation in the plane of the facade). Figure 4a shows that the network identified the wrong mechanism (No. 7) at first, and then, with increasing number of iterations, pinned down the correct mechanism (No. 4).

- **Church of S. Chiara in Carpi**
On the facade macro-element, the real damage mechanism is No. 4 (Translation in the plane of the facade). The wrong mechanism (No. 7) is identified at first, and then, with increasing number of variations, the probability value ascribed by the network to mechanism No. 4 is seen to increase and eventually reach 73% at the end of the training session (figure 4b). This means that no recognition problems were experienced by the network throughout the validation stage.

![Fig. 4 Damage mechanisms prediction: S. Maria Assunta (Scandiano) and S. Chiara (Carpi) churches.](image)

4 **EXAMPLE: PREDICTION OF THE SEISMIC COLLAPSE MECHANISM IN SPHERICAL TANKS**

The theory reported in the previous paragraph has been applied to measure the safety of a spherical gravity tank. Figure 5 depicts the scheme of the structure constituted by a steel spherical shell supported by a steel frame. The frame, square and symmetric in plan, has four columns with built-up shape section (figure 5) and four flexurally rigid transverse beams, so that the frame may be assumed to exhibit a shear type behaviour. The spherical shell is connected to the beams through a median steel ring and is meant to be completely filled with liquid. A reinforced concrete bed foundation completes the structure. The thickness of the square foundation is fixed (0.5 m), whilst its in plan dimension \( B \) equals the diameter of the reservoir (increased of the column section dimensions). The thickness of the column section has also been fixed equal to 5 mm. The input parameters of the problem are summarised in table 2 together with the type of collapse to be predicted.

The final aim of this classification technique is to obtain a network which, at the end of the training stage, is able to supply a prediction of expected type of collapse for each individual tank. The reliability of the predictions supplied is strictly linked with the effectiveness of the training process and is evaluated in successive stages through validation steps consisting of presenting the network with examples not included in the training sample and whose damage data are known, so as to be able to assess the network’s capability to associate the maximum probability level with the type of damage actually observed. In this application a set of 20,000 simulated cases, 5,000 for each collapse class, has been randomly...
chosen to be used for training. A validation test has been performed based on 2000 cases, equally distributed among the four classes and not included in the training set. The results of the validation test are summarised in table 3, where the real collapse classes are correlated to the predicted ones. The diagonal terms show the percentage of successful classifications, whilst the ones that lie out of diagonal give information about the scattering of the estimation.

Table 2: Input parameters with their variability range and collapse mechanisms to be predicted

<table>
<thead>
<tr>
<th>1.1 Input parameters:</th>
<th>1.2 Output: collapse type</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Height, ( H ) (3-10m).</td>
<td></td>
</tr>
<tr>
<td>- Foundation width, ( B ) (2-6m)</td>
<td></td>
</tr>
<tr>
<td>- Liquid density, ( \rho ) (800-1400 kg/m(^3))</td>
<td></td>
</tr>
<tr>
<td>- Column section width, ( L ) (0.16-0.32m)</td>
<td></td>
</tr>
<tr>
<td>- Maximum tension on the soil, ( \sigma_t ) (0.2-1.0 N/mm(^2))</td>
<td></td>
</tr>
<tr>
<td>1) Foundation sliding</td>
<td></td>
</tr>
<tr>
<td>2) Foundation rupture/overturning</td>
<td></td>
</tr>
<tr>
<td>3) Plastic mechanism in the columns</td>
<td></td>
</tr>
<tr>
<td>4) Column instability</td>
<td></td>
</tr>
</tbody>
</table>

Actually, the classification procedure may be viewed as a pattern recognition process, so as that the elements of the parameter vectors can be defined as coordinates of points in an hyperspace. Figure 6 depicts an intersection of the hyperspace with a plane, which reduces to a bi-dimensional representation of the feature space. In this case, the two dimensions are respectively the foundation width, \( B \), and the column section width, \( L \). This graph allows a clear and intuitive interpretation of the results of the pattern recognition process and the space partitioning. In figure 6b also misclassified vectors are represented. One can stress out that all the misclassified points are near the borders of different regions, thus indicating intrinsically ambiguous cases.

![Fig. 6. Feature space partition supplied by the neural network compared to the expected one. The variables here represented are the foundation width \( B \) (abscissa) and the column width \( l=L/4 \). The other three parameters have been kept constant (\( H=3.5 \) m; \( \rho=1100 \) kg/m\(^3\); \( \sigma_t = 0.6 \) N/mm\(^2\)).](image)
Table 3: Results of the validation test on the trained neural network: prediction error for different collapse type

<table>
<thead>
<tr>
<th>Collapse class</th>
<th>% Error in neural classification</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>100.0% 0.0% 0.0% 0.0%</td>
</tr>
<tr>
<td>2</td>
<td>5.8% 92.0% 2.2% 0.0%</td>
</tr>
<tr>
<td>3</td>
<td>6.8% 2.8% 79.4% 11.0%</td>
</tr>
<tr>
<td>4</td>
<td>0.0% 0.0% 1.6% 98.4%</td>
</tr>
</tbody>
</table>

5 CONCLUSIONS

Probabilistic neural networks combine the advantages typical of all neural systems in the representation of functional relationships of any nature with the transparency of probabilistic representations, lending themselves to the simulation of complex phenomena.

In the first part of the paper some results of recent applications on real structures, namely churches and masonry buildings, have been reported. Those applications were successful and satisfactory and proved to be more reliable than other vulnerability assessment techniques based on deterministic models or a priori fixed vulnerability indexes.

The chemical plant component studied above as an application example is very simple and maybe not completely representative of a real structure used in industrial engineering practice. Nevertheless, components of chemical plants are generally belonging to well defined typological families, more easy to be described in a compact parameter vector than masonry structures. Since the method worked well on masonry structures, it is likely that it will perform even better on chemical plant components, but a necessary condition is the availability of a reliable database containing detailed descriptions of these structures and of damage already suffered during earthquakes. It may be argued that such a database is quite difficult and heavy to be constructed in the effective size, but one of the main advantages of the proposed neural expert system is that it is able to treat real cases and simulated ones together. Thus, any incomplete real database can be easily integrated by parametric simulations calibrated on real observations.

REFERENCES