CrossMark

ORIGINAL RESEARCH PAPER

Epistemic uncertainty assessment using Incremental Dynamic Analysis and Neural Networks

Dimitris G. Giovanis[1] · **Michalis Fragiadakis**[1] ⬥ ·
Vissarion Papadopoulos[1]

Abstract Incremental dynamic analysis (IDA) is a powerful method for the seismic performance assessment of structures. IDA is also very efficient for handling uncertainty due to the mechanical properties of the structure. In the latter case, IDA should be performed within a Monte Carlo framework requiring the execution of a vast number of nonlinear response history analyses. The increased computing effort renders the calculation of performance statistics time-consuming and hence the method is not always practical. We propose a scheme based on artificial neural networks (NN) in order to reduce the computational effort. Within a Monte Carlo approach, trained NN can rapidly generate a large sample of IDA curves and therefore allow us to easily calculate useful response statistics and fragility curves. The implementation of the proposed approach is quick, straightforward and quite accurate.

Keywords Epistemic uncertainty · Incremental dynamic analysis · Neural networks · Monte Carlo simulation · Reliability analysis

1 Introduction

The reliability assessment of structures under seismic loads is a topic that has drawn considerable attention over the last years. Reliability assessment in earthquake engineering involves ground motion uncertainty (also known as aleatoric uncertainty, or randomness)

✉ Michalis Fragiadakis
    mfrag@mail.ntua.gr

    Dimitris G. Giovanis
    dgiov16@yahoo.gr

    Vissarion Papadopoulos
    vpapado@central.ntua.gr

[1] School of Civil Engineering, National Technical University of Athens (NTUA), Iroon
    Polytechneiou Str., Zografou, 15780 Athens, Greece

and uncertainty owing to modeling assumptions, omissions and/or errors, otherwise known as epistemic uncertainty, or just uncertainty. The difference between aleatory and epistemic uncertainty is that the latter can be reduced as we acquire more information, e.g. more specimens. Although in practice the distinction is not clear, e.g. uncertainty on the mechanical properties doesn't vanish as the number of tests increases, in this work, the term 'epistemic' refers only to the structural properties. In earthquake engineering applications, emphasis is usually given primarily on the evaluation of the seismic demand due to the record-to-record variability. Recently, Vamvatsikos and Fragiadakis (2010) showed that epistemic uncertainty is equally important, especially at late limit-states, as the structure approaches collapse.

Several researchers have worked on the effect of modelling/parameter uncertainty on structural capacity. For example Celik and Ellingwood (2010) studied the effect of epistemic uncertainties on a reinforced concrete building (RC) providing fragility curves and the corresponding confidence intervals. Jalayer et al. (2010) assessed modelling uncertainty on RC frames using a Bayesian updating framework in order to improve their estimates using measured data. Furthermore, Celarec and Dolšek (2013) used either FOSM or LHS-based MCS combined with the N2 method, in order to evaluate the effects of modelling uncertainties on the seismic response of three RC buildings. Regarding steel frames, the effect of epistemic uncertainty was studied by Kazantzi et al. (2014) using an IDA-based MC approach. Empirical relationships derived from experimental data were used in order to model the cyclic behaviour of steel sections including intra- and inter-component correlation.

Incremental dynamic analysis (IDA) is a powerful method for seismic performance assessment. The method has been successfully combined with reliability analysis methods, e.g. Monte Carlo simulation (MCS), by several researchers (Vamvatsikos and Fragiadakis 2010; Kazantzi et al. 2014; Fragiadakis and Vamvatsikos 2010; Dolsek 2009; Liel et al. 2009). Although IDA-based methods are powerful, they necessitate the execution of a large number of nonlinear response history analyses which renders them beyond the scope of many practical applications. Vamvatsikos and Fragiadakis (2010) discussed the possibility of reducing the computational effort using approximate, moment-estimating methods such as the Rosenblueth's point estimating method (PEM) or the first-order, second-moment (FOSM) method. Using functional approximations or moment-matching, such schemes manage to propagate uncertainty from the random parameters to the model using a few IDA runs/simulations. However, they often become unstable and are sensitive to the properties of the problem's random variables. A different approach for the efficient seismic performance assessment with a reduced number of simulations can be found in Lupoi et al. (2006).

Surrogate models, such as response surface methods (RSM) and neural networks (NN) have emerged as powerful tools able to replace time-consuming procedures in many scientific, or engineering, applications. Such methods usually require a small number of analyses in order to provide quick response estimates. RSM involve fitting a function to given data and then using optimization techniques to determine the parameters of the function. A successful application can be found in Liel et al. (2009) who used RSM within a Monte Carlo procedure for the assessment of RC structures. However, often obtaining the right model may not be possible, or can be quite time-consuming.

Artificial neural networks (NN) is a method based on a simplified model of a biological neuron. The ability of NN to create a mapping between input and output data through training lets them represent quite complex system responses. Compared to RSM, the advantage of NN is that it does not require any prior specification of a suitable fitting

function, and also that it can approximate almost all kinds of non-linear functions, while RSM is limited to quadratic approximations. A review on the use of NN in structural reliability analysis is presented in Chojaczyk et al. (2015), while recently NN were used in the framework of stochastic mechanics in order to approximate the failure probability of a structure (Papadopoulos et al. 2012; Giovanis and Papadopoulos 2015). For the seismic assessment of structures, Lagaros and Fragiadakis (2007) used trained NNs to obtain fragility curves of steel moment frames. They trained a NN in order to predict the conditional interstorey drift demand as function of a vector of seismic intensity measures.

The major advantage when using a trained NN in the core of a Monte Carlo simulation (MCS) is that approximate results can be obtained with orders of magnitude less computational effort compared to the standard procedure. This is due to the associative memory that is featured by these artificial intelligence algorithms which allows them to become efficient surrogates of the finite element model. Once trained, a neural network can rapidly sample the random variables and generate response statistics with minimal computing effort. Therefore, the computing demand is focused on the training phase which is orders of magnitude smaller than that of the actual simulations. For example, in terms of computing cost, the difference between 10^2 and 10^6 simulations is practically negligible.

Neural networks are widely available to scientists and engineers through powerful numerical libraries (i.e. Matlab, NAG, IMSL). Different types of NNs such as feed-forward, multi-layer perceptron (MLP) and radial basis function networks are widely used. These networks, provide a generic functional representation and have been shown to be capable of approximating any continuous function with acceptable accuracy. Implementing the proposed NN-based method is straightforward once the user/engineer becomes familiar with the underline concepts.

2 Epistemic uncertainty assessment using incremental dynamic analysis

2.1 Incremental dynamic analysis (IDA)

Incremental dynamic analysis (IDA) (Vamvatsikos and Cornell 2002) offers thorough seismic demand and capacity prediction capability for the complete range of the model's response, from elastic to yielding then to nonlinear inelastic and finally to global dynamic instability. IDA involves subjecting a structural model to one (or more) ground motion record(s), each scaled to multiple levels of intensity, thus producing curve(s) of demand versus seismic intensity. Every dynamic analysis is characterized by two scalars, an intensity measure (IM) and an engineering demand parameter (EDP). The former represents seismic intensity and the latter measures the demand, or the 'damage'. For moderate-period structures with no near-fault activity, an appropriate choice for the IM is the 5 %-damped, first-mode spectral acceleration $S_a(T_1, 5\%)$. The EDP chosen in our work is the maximum interstorey drift θ_{max}, a reasonable choice for deformation-sensitive structural damage.

Multiple single-record IDAs are performed using the hunt-and-fill algorithm (Vamvatsikos and Cornell 2004). This method allows capturing every IDA curve with just 12 or 14 runs per ground motion record. The single-record IDAs are subsequently summarized to produce the median and the 16, 84 % fractile IDA curves that sufficiently characterize seismic demand and capacity. Figure 1a shows 30 single-record IDA curves for a nine-storey steel frame. Each curve has been obtained from 12 nonlinear response history
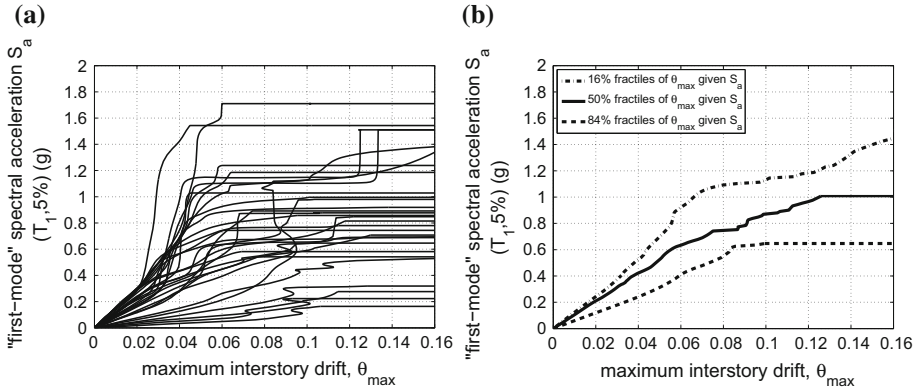
(a)

(b)

Fig. 1 Incremental dynamic analysis (IDA) curves: **a** 30 single-record IDAs, and **b** summarization of the 30 IDA curves into their median, 16 and 84 % fractile curves

analyses. The EDP-IM pairs of every simulation are interpolated with appropriate splines to obtain a continuous curve in the EDP-IM plane. The fractile capacities can be summarized in terms of θ_{max} given the spectral acceleration $S_a(T_1, 5\%)$ ($\theta_{max}|S_a(T_1, 5\%)$, or otherwise in terms of spectral acceleration $S_a(T_1, 5\%)$ given θ_{max} ($S_a(T_1, 5\%)|\theta_{max}$). In the remainder of the paper we concentrate on $S_a(T_1, 5\%)$ capacities conditional on θ_{max}. In either case, the median (50 % fractile) provides a "central" capacity curve, while the 16 and 84 % percentiles give a measure of the dispersion around the median (Fig. 1b).

2.2 IDA for epistemic uncertainty estimation

Monte Carlo simulation (MCS) is the most comprehensive method in structural reliability analysis. MCS creates a population of N possible instances of the structure by sampling N times from the parameter distributions and subsequently performing a "simulation" for every realization of the structure. Assuming that a sufficiently large number of the structure's instances have been sampled, we can reliably estimate the full distribution of every response statistic and also calculate limit-state exceedance probabilities. When IDA is adopted for seismic performance assessment using natural ground motion records, a "simulation" refers to a median IDA curve obtained after R single-record IDAs. We therefore, need $R \times N$ single-record IDAs to calculate unbiased estimates of the mean and the variance of the median IDAs due to epistemic uncertainty. Thus, if $\ln S_{a,j}^{50\%}, j = 1, \dots N$ are the median S_a-capacities for a given value of θ_{max} and $\overline{\ln S_a^{50\%}}$ is the mean of their natural logarithm, we obtain the overall median as:

$$\Delta S_a = \text{med}_j(S_{a,j}^{50\%}) \approx \overline{\ln S_{a,j}^{50\%}} \tag{1}$$

where "med_j" is the median operator over all indices j. We have chosen to simplify the notation writing $S_a(T_1, 5\%)$ as S_a, while S_a and ΔS_a are always conditional on the maximum interstorey drift θ_{max}, thus denoting IDA curves.

Furthermore, Cornell et al. (2002) proposed that the dispersion caused by epistemic uncertainty in the median capacity ΔS_a can be characterized by its β-value, β_U, where the

subscript 'U' denotes (epistemic) uncertainty. β_U is calculated directly as the standard deviation of the natural logarithm of the median capacities:

$$\beta_U = \sqrt{\frac{\sum_{j=1}^{N} (\ln S_{a,j}^{50\%} - \overline{\ln S_a^{50\%}})^2}{N-1}} \tag{2}$$

In Vamvatsikos and Fragiadakis (2010) and Fragiadakis and Vamvatsikos (2010) MCS is performed using Latin Hypercube Sampling (LHS). LHS is efficient when response statistics (e.g. mean and dispersion) are calculated and is used to improve the sampling and to reduce the number of simulations. When NN predictions are used, it is costless to perform a very large number of simulations and therefore simple, or 'crude', MCS is also efficient. In the case study here examined, the two methods produced identical results.

A simpler alternative to MCS is the use of moment-estimating methods such as the point estimate method (PEM) or the first-order-second-moment method (FOSM). These methods are typically based on the use of only a handful of runs for appropriately perturbed versions of the basecase structure, defined as the realization of the structure when all design variables are set equal to their mean. As shown in Vamvatsikos and Fragiadakis (2010), and here verified, both PEM and FOSM tend to introduce considerable error compared to the results of MCS. Thus, the reduction of the computational effort comes with a loss in accuracy, whilst the NN-based procedure retains the same small number of simulations but offers increased accuracy. Furthermore, some researchers prefer to use Eqs. (1) and (2) with a limited number of simulations (e.g. 20 runs). This approach may offer a reasonable estimate of the mean, but the estimate of the dispersion will not be acceptable. This was demonstrated in Fragiadakis and Vamvatsikos (2010), where 90 % confidence intervals show that there is little confidence on the conditional dispersion unless a sufficient number of MC simulations is available.

Once ΔS_a and β_U are known, risk assessment is straightforward. Fragility curves, i.e. the conditional probability that a limit-state is exceed, $P(\theta_{max} \geq \theta_{max}^{lim}|S_a(T_1, 5\%))$, can be easily calculated assuming that the data follow the lognormal distribution. The same result is obtained, bypassing the need to assume a distribution, through calculating the empirical distribution function i.e. counting the number of samples that the limit-state threshold was exceeded ($\theta_{max} \geq \theta_{max}^{lim}$). Fragility curves can be convolved with the site hazard curve to calculate limit-state mean annual frequencies.

3 Artificial neural networks

Artificial neural networks, or just neural networks (NN), are information-processing models configured for a specific application through a training process. Trained NN provide the rapid mapping between given input and output quantities to (similar to curve fitting) and thereby can be used as meta-models enhancing the efficiency of numerical simulation. Compared to the actual numerical simulation, the major advantage of a trained NN is that it can produce results in a fraction of wall clock time, requiring orders of magnitude less computational effort, provided that the network is successfully trained.

3.1 Basic structure of an artificial neuron

A neural network consists of a number of linked units (neurons) and attempts to create a desired mapping between the input and the output data of a training set. Artificial neurons (Fig. 2) process the information of a training set to predict the output vector. When operating within a network, a neuron, j receives the vector of input parameters x_i, $(i = 1, \ldots, n)$ from previous neurons. Every input x_i is multiplied by the connection weight, w_{ij} of neuron j. The connection weights correspond to the strength of the influence of each of the preceding neurons. After the input parameters have been multiplied with the weights, the sum of their products is calculated as: $a_j = \sum_{i=1}^{n} w_{ij} \cdot x_j + b$. b is a bias which acts as a level shifter, i.e increasing or decreasing by a constant the sum of the weighted inputs within every neuron. This allows the neuron to cover a wider input range. Furthermore, the activation function f is applied on the sum a_j. The activation function is an essential part of the artificial neuron structure which introduces the nonlinearity. Commonly used activation functions are the linear, the threshold and the sigmoid functions. The concept of the neuron is schematically shown in Fig. 2.

3.2 Architecture of an artificial neural network

A large number of different NN architectures have been developed over the recent years: the multilayer feed-forward, the radial basis function, Bayesian regularized, networks with self-organizing maps recurrent networks, among others (Hagan et al. 1996). In this work a feed-forward neural network was used. Multilayer feed-forward neural networks are the most commonly used and most appropriate for problems regarding non-linear modeling and function approximation, such as those considered here. In Fig. 3 two simple feed-forward NN architectures are shown. The input neurons are shown as squares because they only act as input terminal points. The circular nodes represent basic processing neurons which process information, that is, the inputs are summed and sent through an activation function. With feed-forward networks, there is always an input layer of neurons and some hidden layers leading to the output layer. Note that the input is forward propagated
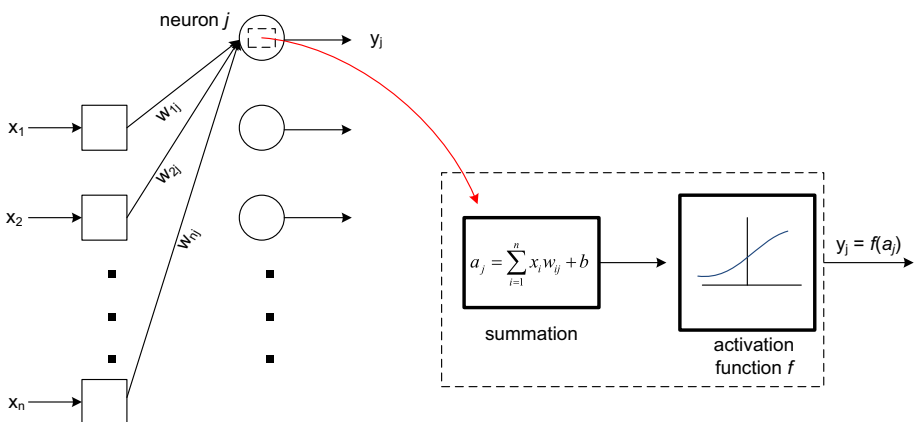

Fig. 2 Structure of an artificial neuron (**x**, **y** are the vectors of input and output variables, respectively, and f is the activation function)

(a)   **(b)**

output layer   hidden layer

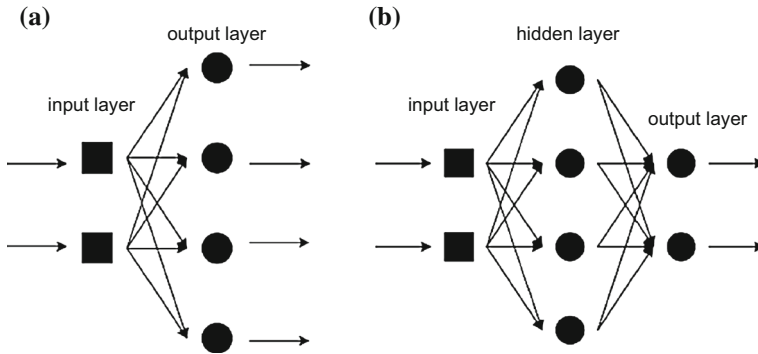input layer   input layer   output layer

Fig. 3 **a** A single-layer network and **b** a multi-layer network

meaning that data and calculations flow in a single direction, from the input data towards the output.

3.3 Training of neural networks

The workflow for the neural network design process has six primary steps: (1) collect the available data, (2) configure the network so that it is compatible with the problem under consideration, as defined by sample data, (3) initialize the weights and biases, (4) Train the network, (5) validate the network and, (6) use the network. The most important step is the fourth, the *training phase*, during which the synaptic weights are tuned in order to obtain a mapping that fits closely the training set. The training of a NN can be considered as a general function optimization problem, where the adjustable parameters are the weights **w** of the network (Fig. 2b). Several algorithms have been developed for this optimization problem over the years. It is very difficult to know which training algorithm will be best for the problem considered beforehand. It depends on many factors, including the complexity of the problem, the number of data points in the training set, the number of weights and biases in the network, the error goal.

One of the common problems occurring during the training of a neural network is the overfitting. In such case the error of the NN prediction over the training data is small, but when new data is presented to the network this error increases. Possible remedies for the overfitting problem are: (1) stop the training early, (2) retrain several neural networks,each starting with different initial weights (3) add some form of regularization term to the error function to encourage smoother network mappings, (4) add noise to the training patterns to smear out the data points.

In this work the Bayesian regularisation algorithm proposed by MacKay (1992) was used. Different regularisation algorithms could have been also adopted, however, this was preferred since based in our experience it minimises the possibility of overfitting. Regularisation is a way of dealing with the negative effect of large weights, which can cause excessively large variance of the output. The idea of regularisation is to make the network response smoother through the modification in the objective function. This is achieved adding a penalty term, which is equal to the sum of squares of the network weights. The additional term favours small values of weights and minimises the tendency of the model to overfit. MacKay introduced the Bayesian regularisation which sets the optimal performance function to achieve the best generalisation based on Bayesian inference techniques.

The Bayesian optimisation of the regularisation parameters requires the computation of the Hessian matrix at the minimum point which is obtained using the Levenberg–Marquardt optimization algorithm.

4 Proposed methodology for neural network training

As discussed in Sect. 2.2, in MCS we sample N times the epistemic random variables of the problem and we create N realizations of the structure. For each of the N realizations, a series of nonlinear response history analyses is performed in order to obtain N median (multi-record) IDAs. Δ_{S_a} and the dispersion β_U are then estimated as function of θ_{max} according to Eqs. (1) and (2). The N realizations are performed using a properly trained NN and the samples are chosen either with LHS or with crude Monte Carlo sampling. Since there is no limitation on the size of N, both sampling methods will give identical results. In the remainder of the paper crude MCS is preferred because of its simplicity.

It is evident that network training is the most important step. The training procedure is schematically presented in Fig. 4 and consists of three phases:
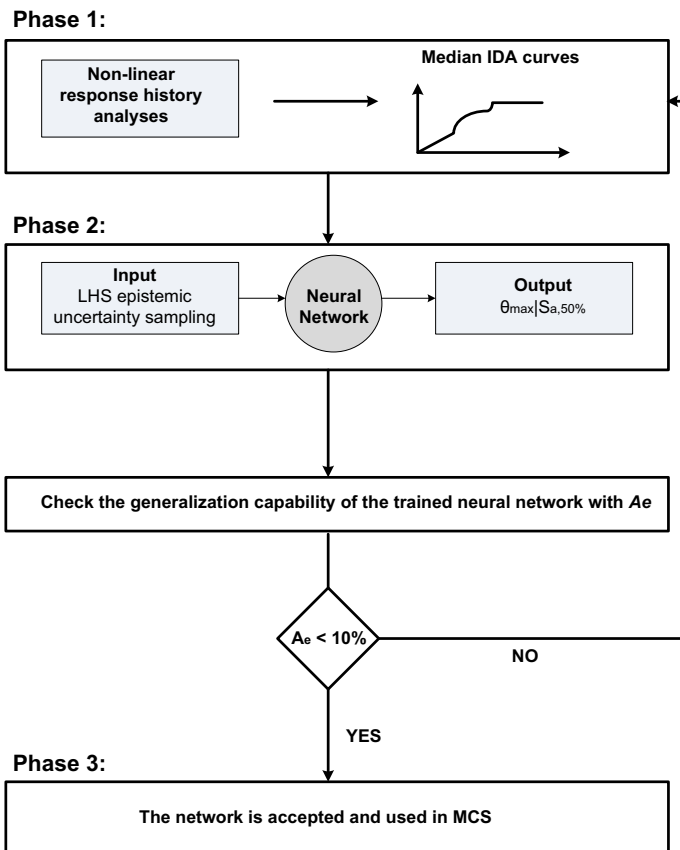

Fig. 4 Schematic representation of the proposed NN training approach

Phase 1   Generate N_{train} and N_{test} samples from the epistemic random variables using LHS.

Phase 2   Perform $N_{train} + N_{test}$ IDAs and calculate their median $S_a^{50\%}$ capacities as function of the EDP, which here is the maximum interstorey drift (θ_{max}).

Phase 3   Use the N_{train} samples to train the network and the N_{test} samples to test its generalization capability, i.e. its ability to give accurate predictions with a new, unknown, sample.

In Phase 1 we perform a series of IDA analyses in order to obtain N_{train} and N_{test} training and testing patterns, respectively, in the form of θ_{max}–$S_a^{50\%}$ pairs. Every training and testing pattern has been obtained using a sample of the K random parameters which describe the uncertainty. The sample size N_{train} that is required for a "good" NN training depends on the problem at hand. Different sample sizes should be attempted on a trial-and-error basis. This is performed only once, at the beginning of the training phase. N_{test} simulations are also performed in order to test the prediction capability of the trained network. Usually, N_{test} is fraction of N_{train}, while, in general, there is no restriction on the ratio of N_{test} over N_{train}; a ratio as high as possible is always desirable. Typically N_{test} is equal to 10–20 % of N_{train}.

The appropriate selection of the training data is important for successfully training the network. Attention should be paid on choosing input/output data that cover the entire range of possible response estimates for the random variables assumed. Although the number of learning patterns plays its own role in the accuracy of the predictions, the distribution of the samples is of greater importance. We therefore use latin hypercube sampling (LHS) in order to obtain training data that cover as uniformly as possible the range of values that the random variables may take. This should not be confused with the use of LHS within a Monte Carlo simulation for calculating mean and dispersion estimates. Therefore the network is trained with an input vector that contains LHS samples of the K random variables and the output vector contains the corresponding IDA curves.

In order to avoid overfitting during the training phase, several NNs, each with different initial weights, were trained in a Bayesian regularization framework in order to ensure that a network with good generalization is found. In every NN training, an early stopping criterion was used in order to further improve the network's generalization performance, assessed with the following error function:

$$er^i = \sum_{j=1}^{m} \frac{|y_j - t_j|}{y_j} \times 100 \quad i = 1, \ldots, N_{test} \tag{3}$$

where m is the size of θ_{max} values equal to the number of points that the IDA curve is discretized, y_j is the exact $\theta_{max}|S_a(T_1, 5\%)$ value and t_j is the NN prediction. The average of this error, A_e, is obtained as:

$$A_e = \frac{\sum_{k=1}^{N_{test}} er^{(k)}}{N_{test}} \tag{4}$$

The trained NN adopted should have an A_e value smaller than a user-specified threshold value. If this criterion is not specified, the number of training patterns N_{train} should be increased and/or different NN architectures and learning algorithms should be examined.

5 Latin hypercube sampling

Latin hypercube sampling (LHS) is adopted so that each of the random variables is sampled from equal-sized bins. According to LHS, a single point (or sample) is randomly chosen from every bin. Figure 5 shows conceptually the stratified sampling of two random variables, X_1 and X_2. More specifically, Fig. 5a shows $N_{LHS} = 5$ samples of X_1 with respect to its cumulative distribution function and Fig. 5b shows that a single-value of X_1 and X_2 is sampled from each of the five equal-sized bins in order to generate $N_{LHS} = 5$ samples.

The advantage of LHS over crude sampling is that it reduces the variance of statistical estimates, while the random variables are sampled from the complete range of their possible values, thus ensuring that no sub-domain is over-sampled. Moreover, during LHS sampling random correlation can be introduced between the random variables, especially in the case of very small N_{LHS} number where the number of interval combinations is rather limited. In order to ensure zero correlation among the variables describing epistemic uncertainty the approach discussed by Iman and Conover (1982) was adopted.

6 Numerical example

6.1 Structural model considered

The structure considered is a nine-story steel moment-resisting frame (Fig. 6a) with a single-story basement. The building was designed for a Los Angeles site, following the 1997 NEHRP provisions. A centerline model with nonlinear beam-column connections was formed allowing for plastic hinge formation at the beam ends while the columns are assumed to remain elastic. The problem random variables are the properties of the plastic hinges, as discussed in the next subsection. The structural model includes $P - \Delta$ effects while the internal gravity frames have been directly incorporated with a gravity-carrying frame, as shown at the right of the building in Fig. 6a. The fundamental period of the
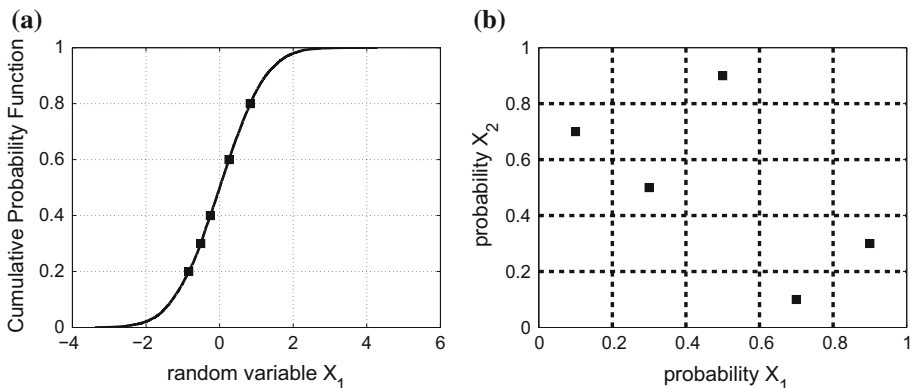

Fig. 5 The latin hypercube sampling (LHS) method: **a** cumulative distribution function of variable X_1 in order to obtain five equally probable bins, **b** the samples are randomly generated sampling once from each bin
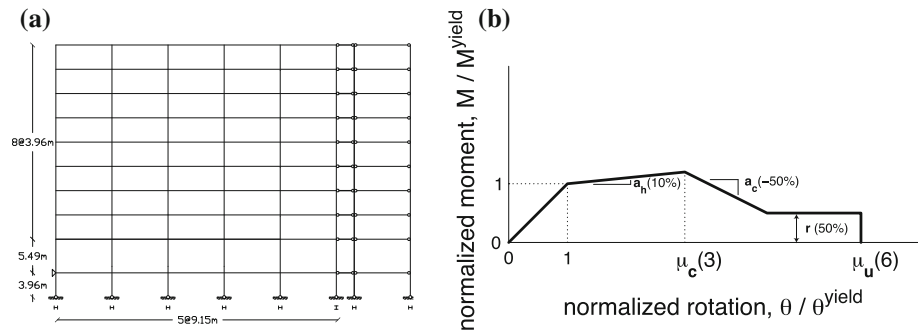
(a)

(b)

Fig. 6 **a** The nine-storey steel moment-resisting frame case study, **b** moment-rotation relationship of a beam point-hinge in normalized coordinates and definition of the problem random variables

reference frame (when all random variables are set equal to their mean) is $T_1 = 2.35$ s and accounts for approximately 84 % of the total mass. A suite of thirty ordinary ground motion records representing a scenario earthquake was used. These records belong to a bin of relatively large magnitudes of 6.5–6.9 and moderate distances, all recorded on firm soil and bearing no marks of directivity. More details about the structure and the ground motion records can be found in Vamvatsikos and Fragiadakis (2010), Fragiadakis and Vamvatsikos (2010).

6.2 Random variables

The epistemic random variables are the properties of the beam-hinges. The beam-hinges are modelled as rotational springs with a quadrilinear moment-rotation backbone as shown in Fig. 6b. The behaviour of the springs is symmetric for both positive and negative rotations. Six parameters are necessary to fully describe the backbone of the monotonic envelope of the hinge moment-rotation relationship. More specifically, the backbone hardens after a normalized yield moment a_{M_y}, having a non-negative slope of a_h up to a normalized rotation (or rotational ductility) μ_c where the negative stiffness segment starts. The drop, at a slope a_c, is arrested by the residual plateau appearing at normalized height r that abruptly ends at the ultimate rotational ductility μ_u. All quantities are normalized either with the elastic slope a_e, the yield moment M_y or the yield rotation θ_y.

The variability of the properties of the quadrilinear backbone is the only source of epistemic uncertainty and the six normalized parameters (a_{M_y}, a_h, μ_c, a_c, r, μ_u) are the problem random variables. The random variables are independently normally distributed with the mean and coefficient of variation (c.o.v.) shown in Table 1. The mean values

Table 1 Random variables and their probabilistic description	Distribution $\mathcal{N}(\mu, \sigma^2)$	c.o.v	Lower bound	Upper bound
a_{M_y}	$\mathcal{N}(1.0, 0.04)$	0.2	0.70	1.30
a_h	$\mathcal{N}(0.1, 0.016)$	0.4	0.04	0.16
μ_c	$\mathcal{N}(3.0, 1.44)$	0.4	1.20	4.80
a_c	$\mathcal{N}(-0.5, 0.04)$	0.4	−0.80	−0.20
r	$\mathcal{N}(0.5, 0.04)$	0.4	0.20	0.80
μ_u	$\mathcal{N}(6.0, 5.76)$	0.4	2.40	9.60

represent the best estimates of the backbone parameters. We, therefore, used c.o.v values equal to 40 % for all the parameters, except for the yield moment where 20 % COV was assumed. These c.o.v values practically cover the whole range of possible values that an engineer may chose. To avoid assigning values of no physical meaning, e.g. $a_h > 1$ or $r < 0$, the distributions are appropriately truncated within 1.5 standard deviations and the resulting boundaries are as shown in the last two columns of Table 1.

The plastic hinge properties are assumed to be varying simultaneously for every frame connection, hence being fully correlated. This is expected to have a more pronounced effect on the global capacity. Moreover, from the designer's standpoint, there is very little information regarding the values she/he should assign to the building's inelastic model, with M_y being perhaps the only exception. Hence our assumption reasonably captures the response variation due to different modeling assumptions, since any designer would most likely assume the same values throughout the structure and perform some sort of sensitivity analysis to check the sensitivity of the model. A methodology considering spatial correlation can be found in Kazantzi et al. (2014).

6.3 Neural network training

We generate three samples of size N_{train} equal to 20, 30 and 40 using latin hypercube sampling (LHS), as discussed in Sect. 5. For every sample we obtain a new realisation of the structure and we perform an incremental dynamic analysis with thirty ground motion records. Therefore, the network is trained using N_{train} samples of the six random variables (NN input) and the corresponding median IDA curves (NN output). Figure 7a, b show the median IDA curves for learning patterns of 20 and 40 samples, respectively. When sampling to obtain the NN input, care is taken so that there is no spurious correlation among the samples, as discussed in Iman and Conover (1982). Increasing the number of samples (i.e 20 instead of 40) results to a better representation of the probability space and thus to better network training at the expense of more IDA simulations.

According to Hagan et al. (1996), the number of training samples should be larger than the number of adjustable parameters:
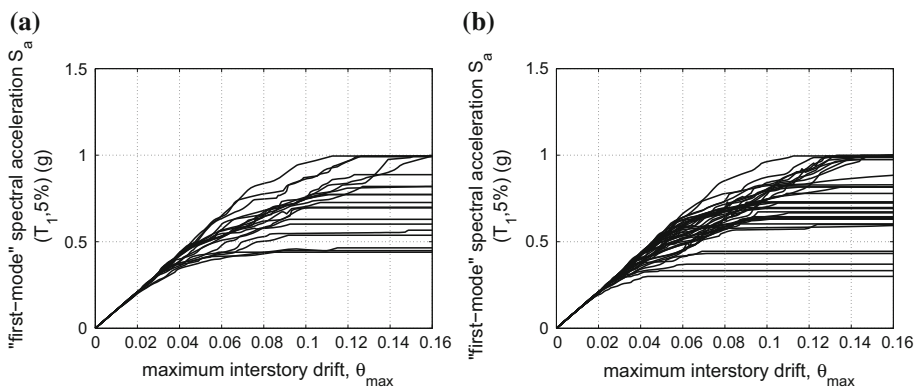

Fig. 7 N_{train} median IDA curves used for NN training: **a** $N_{train} = 20$ samples, and **b** $N_{train} = 40$ samples

$$(n+2)K + 1 < N_{train} \tag{5}$$

where n is the number of neurons of the single hidden layer of a network. The "optimum" number of samples depends on the type of the problem and on the number of random variables. For the nine-storey steel frame, twenty IDA simulations give quite accurate results, while larger samples were also investigated and were found to further improve the prediction capability of the network. Once the input and the output vectors are obtained, we choose the appropriate architecture for the Neural Network. The network architecture also depends on the size of each input vector and on the number of learning patterns N_{train}. The generalization capability of each N_{train} size is tested using $N_{test} = 15$ testing samples. Often the trained network with the smallest mean prediction error A_e of Eq. (4) or with a mean prediction error less than a threshold value (e.g. 10 %) is adopted.

Figure 8 shows four NN-generated median IDA curves obtained with the best NN trained with learning patterns of size equal to 20, 30 and 40. The four samples shown, were randomly chosen and have parameters equal to: $\{a_{My}, a_h, \mu_c, a_c, r, \mu_u\} = \{0.77, 0.09, 3.10, -0.55, 0.42, 5.17\}$, $\{0.86, 0.15, 4.19, -0.51, 0.40, 8.81\}$, $\{1.04, 0.14, 4.02, -0.32, 0.60, 4.99\}$ and $\{0.73, 0.08, 2.77, -0.50, 0.46, 9.13\}$. In all four cases, $N_{train} = 40$ gives estimates very close to the exact median IDA curves, while the smaller samples give also good
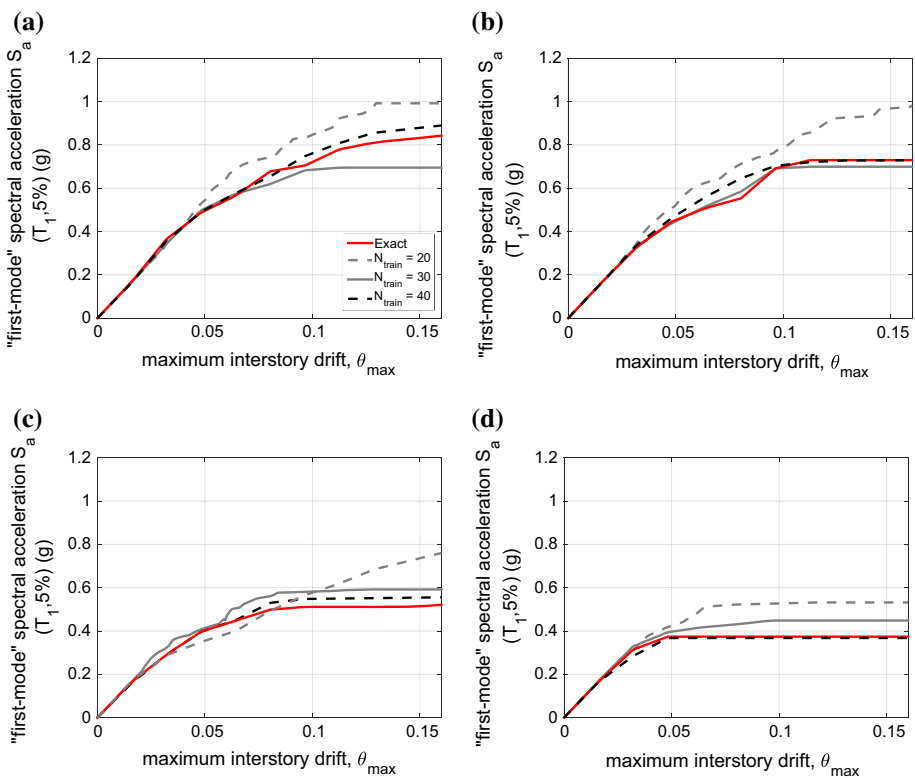

Fig. 8 Single-record IDA curves obtained with the proposed NN-based algorithm. The plots compare the actual IDA curve to those obtained with training sets of size $N_{train}= 20$, 30 and 40

estimates up to a θ_{max} value in the early inelastic range. Error is introduced as the structure approaches collapse, where the $N_{train} = 20$ case sometimes may lead to large errors.

Figure 9 summarizes the results of NN testing for the three sample sizes compared. The testing patterns are compared with respect to the overall error of Eq. (3). In most cases the error is <10 %, while the error for $N_{train} = 40$ is below 5 %, practically for all samples. This plot should be seen together with Fig. 8, which indicates that some, small, error exists mostly at limit-states beyond yielding. Cases a, b, c and d of Fig. 8, correspond to $N_{test}= 3$, 8, 1 and 4, according to their numbering in Fig. 9. One should bare in mind that 20 % error, close to that of the $N_{train} = 20$ case, is often acceptable in earthquake engineering applications. Finally, the mean error of the 15 testing samples for $N_{train} = 20$, 30 and 40 were found equal to 8.92, 6.1 and 2.2 %, respectively.

6.4 Response statistics

Once the network is trained and tested, we perform Monte Carlo simulation sampling the six epistemic random variables. Figure 10a shows 1000 median IDA curves obtained with the trained NN for the case of $N_{train} = 20$ and $N_{train} = 40$. The median IDAs produced by the trained NNs are close, while the $N_{train} = 40$ curves become wider for large S_a-capacities. Furthermore, in order to have a reference solution Monte Carlo simulation was performed without the aid of the neural network, running a vast number of nonlinear response history analyses. As discussed in Vamvatsikos and Fragiadakis (2010) a sample size of $N_{train} = 150 - 200$ Monte Carlo simulations with LHS sampling is expected to provide a close estimate of the response statistics of the problem considered here. Hence, Fig. 11 shows 150 median IDA curves obtained creating 150 instances of the structure and using thirty records. In order to obtain the reference solution, $30 \times 150 = 4500$ single-record IDAs were carried out, performing in total $12 \times 4500 = 54{,}000$ nonlinear response history analyses.

The IDA curves of the Monte Carlo simulation are post-processed to produce the overall median ΔS_a conditional on θ_{max} and also the conditional β_U-dispersion according to Eqs. (1) and (2), respectively. As we can see in Fig. 12a, the conditional median, Δ_{S_a}, is very close to that of the actual IDA for the every limit-state, until $\theta_{max} = 0.1$, while for

Fig. 9 Generalization performance of NN in term of the error of Eq. (3) for learning patterns of size $N_{train} = 20$, 30 and 40
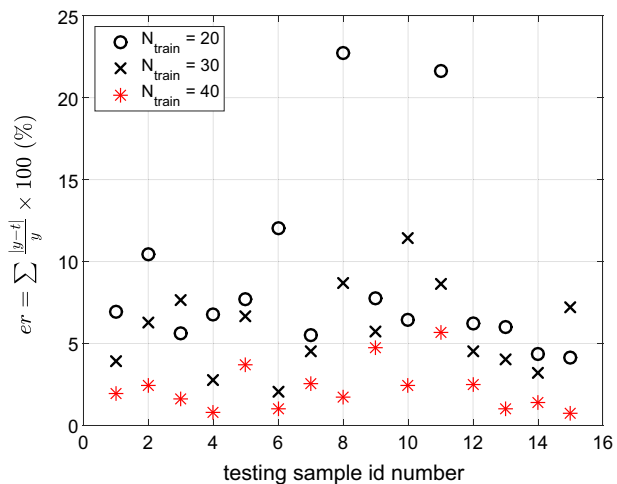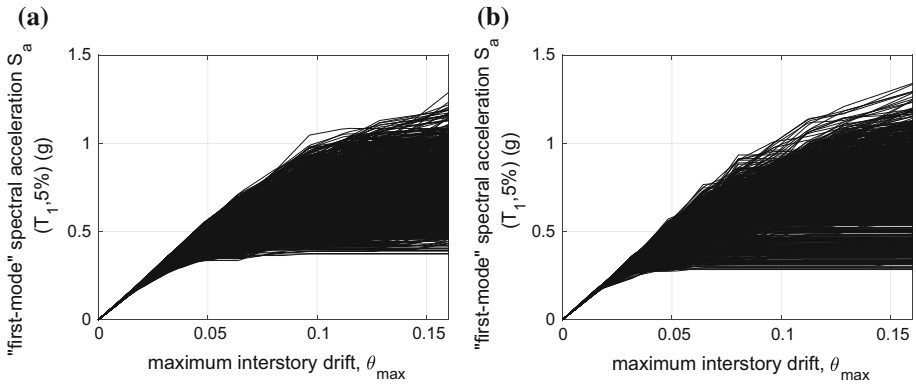
Fig. 10 1000 θ_{max}-capacities given $S_{a,50\%}$ (median IDA curves), generated with a NN: **a** $N_{train} = 20$ samples, and **b** $N_{train} = 40$ samples

Fig. 11 $N = 150$ median response curves obtained through IDA. These curves are here used as our reference solution as they were obtained performing nonlinear response history analysis
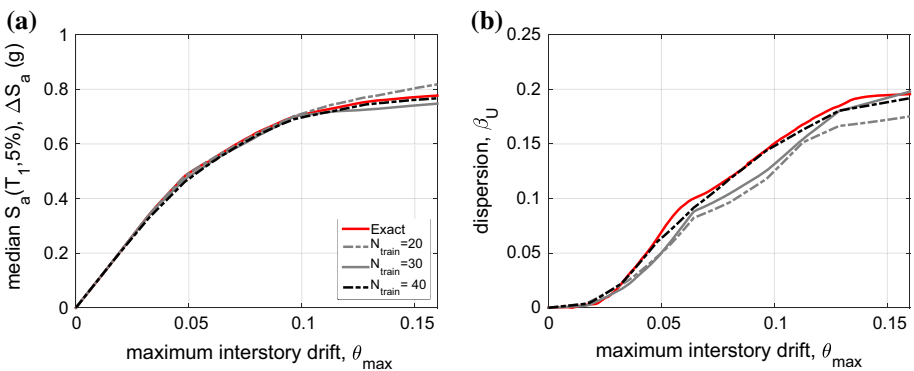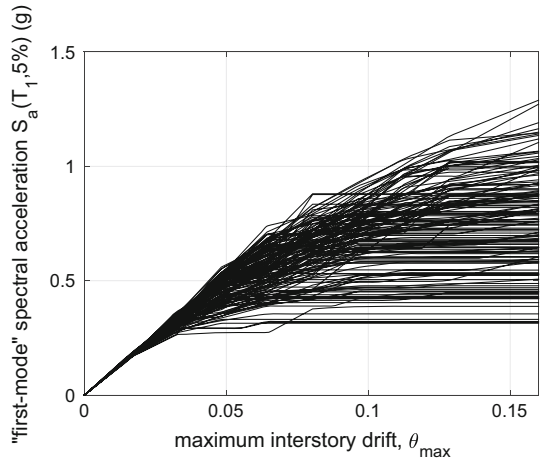

Fig. 12 Response estimations of the proposed NN-based method for $N_{train} = 20$, 30 and 40 against the exact results: **a** ΔS_a, and **b** dispersion β_U
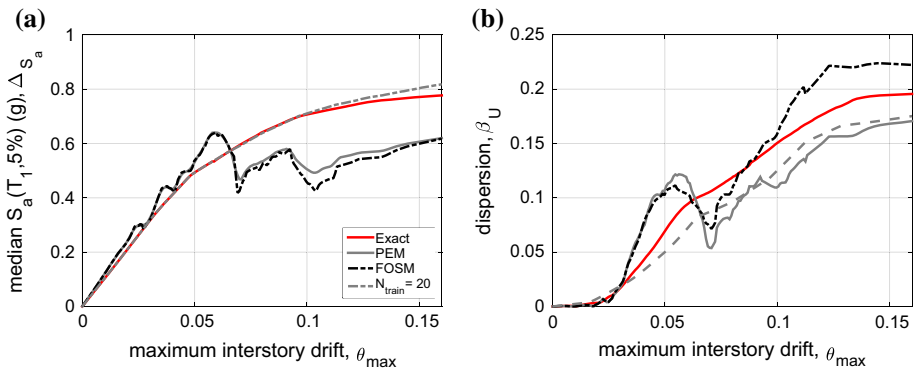
Fig. 13 Post-processing of the MCS IDA curves: **a** median ΔS_a, and **b** dispersion β_U

higher θ_{max} values the agreement is still remarkable. Figure 12b shows the dispersion b_U of the $S_a(T_1, 5\%)$-capacities conditioned on θ_{max}. All trained networks provide excellent predictions of the the overall median Δ_{S_a}, while the results of the $N_{train} = 40$ case practically cannot be discerned from the exact solution. The $N_{train} = 20$ and 30 cases provide also good results but with a small bias, underestimating the actual b_U values. Hence as N_{train} increases, the bias quickly vanishes, converging to the exact dispersion values.

To demonstrate the importance of the proposed NN-based method, we also examine the performance of other approximating IDA-based methods. Figure 13 shows the performance of FOSM and PEM, which in references Vamvatsikos and Fragiadakis (2010), Fragiadakis and Vamvatsikos (2010) are reported as efficient alternatives. These methods require $2K + 1 = 13$ simulations, where K is the number of random variables. As shown in Fig. 13, these methods were outperformed by the 20-sample, proposed NN-based method, which using only 20 samples/simulations gave improved results.

Figure 14 shows the limit-state fragility curves for the steel frame considered. As already discussed the proposed approach allows to directly calculate the probability through the empirical distribution function. The curves have been obtained for θ_{max} values

Fig. 14 Limit-state fragility curves obtained with the proposed NN-based algorithm
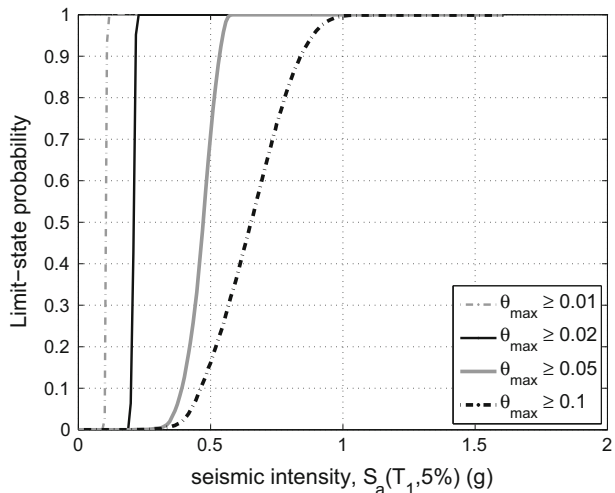
equal to 1, 2, 5 and 10 %. For the frame considered, the limit-states that indicate the yielding of the structure ($\theta_{max} \geq$ 1 % and $\theta_{max} \geq$ 2 %), curves are very steep, indicating the sudden exceedance of the limit-state threshold at S_a values close to 0.1 and 0.2 g, respectively. Limit-states $\theta_{max} \geq$ 5 % and $\theta_{max} \geq$ 10 %, indicate severe damage and collapse, respectively, and have a milder slope and thus more dispersion around them. These vulnerability curves can be convolved with the site's hazard curve to calculate the limit-state mean annual frequencies.

7 Conclusions

A neural network-based procedure is proposed for obtaining seismic response estimates considering epistemic uncertainty. The use of neural networks is motivated by the need for an accurate reliability analysis procedure with reduced computing effort. The proposed method was demonstrated on a nine-storey steel frame considering six epistemic random variables that describe the monotonic backbone of the beam connections. The success of the method relies on the training of the network, and on the number of samples used. The Bayesian regularisation algorithm was used for this purpose along with a proper NN architecture. Since each 'sample' corresponds to a median IDA curve, the selection of the sample size controls the computing effort. Twenty samples were found adequate for case study examined, while the selection of the samples was based on a LHS scheme. More samples, e.g. 40, were shown to completely remove any error in the NN predictions. Once the NN is trained either crude or LHS-based Monte Carlo can be used to calculate the median and the dispersion, or directly estimate the vulnerability thus bypassing the log-normality assumption. The proposed method was also compared to PEM and FOSM, which are also approximating methods that require, roughly a similar number of simulations. The comparison showed the NN-based method is stable for the whole range of limit-states and more accurate than other methods in the literature.

Appendix: Matlab code

Below we provide two basic scripts that in a few lines of Matlab code show how a NN can be trained and then called (Demuth et al. 2008). These are basic scripts and have been kept as simple as possible. Therefore, no guidance on selecting the training samples and avoiding the problem of overfitting is given.

The first script loads the input–output training samples (TrainingSamples.mat, SaT1.mat, maxTheta.mat) and calls the newff command to create a feed-forward backpropagation network. The network is trained with the train command and stored as net. Lines starting with '%' are comments which are provided to make the code self-explanatory.

```
% Input (Ntrain samples of the K random variables)
input = load('TrainingSamples.mat');

% Output (Ntrain IDA curves - Sa(T1,5%) values of the median IDAs)
SaT1 = load('SaT1.mat');
theta_max = load('maxTheta.mat');
output = SaT1;

% number of neurons of the hidden layer (H)
H = 2;

% Create a feed-forward backpropagation network
net = newff(input,output,H);

% Train the network
net = train(net,input,output);
```

sim command is used to obtain the NN's prediction for a given input. Note that the sim, newff and train are commands available provided that the Matlab Neural Network Toolbox is installed.

```
TestingSample = load('TestingSamples.mat');
SaT1_NN = sim(net,TestingSample);
```

References

Celarec D, Dolšek M (2013) The impact of modelling uncertainties on the seismic performance assessment of reinforced concrete frame buildings. Eng Struct 52:340–354

Celik OC, Ellingwood BR (2010) Seismic fragilities for non-ductile reinforced concrete frames: role of aleatoric and epistemic uncertainties. J Struct Saf 32(1):1–12

Chojaczyk AA, Teixeira AP, Neves LC, Cardoso JB, Soares CG (2015) Review and application of artificial neural networks models in reliability analysis of steel structures. Struct Saf 52A:78–89

Cornell CA, Jalayer F, Hamburger RO, Foutch DA (2002) The probabilistic basis for the 2000 SAC/FEMA steel moment frame guidelines. ASCE J Struct Eng 128(4):526–533

Demuth H, Beale M, Hagan M (2008) Neural Network Toolbox^{TM} 6

Dolsek M (2009) Incremental dynamic analysis with consideration of modelling uncertainties. Earthq Eng Struct Dyn 38(6):805–825

Fragiadakis M, Vamvatsikos D (2010) Fast performance uncertainty estimation via pushover and approximate IDA. Earthq Eng Struct Dyn 39(6):683–703

Giovanis DG, Papadopoulos V (2015) Spectral representation-based neural network assisted stochastic structural mechanics. Eng Struct 84:382–394

Hagan MT, Demuth HB, Beale M (1996) Neural network design. PWS Publishing Company, Boston

Iman R, Conover W (1982) A distribution-free approach to inducing rank correlation among input variables. Commun Stat Simul Comput 11(3):311–334

Jalayer F, Iervolino I, Manfredi G (2010) Structural modeling uncertainties and their influence on seismic assessment of existing RC structures. J Struct Saf 32:220–228

Kazantzi AK, Vamvatsikos D, Lignos DG (2014) Seismic performance of steel moment-resisting frame subjected to strength and ductility uncertainty. Eng Struct 78:69–77

Lagaros ND, Fragiadakis M (2007) Fragility assessment of steel frames using neural networks. Earthq Spectra 23(4):735–752

Liel A, Haselton C, Deierlein G, Baker JW (2009) Incorporating modeling uncertainties in the assessment of seismic collapse risk of buildings. Struct Saf 31(2):197–211

Lupoi G, Franchin P, Lupoi A, Pinto PE (2006) Seismic fragility analysis of structural systems. J Eng Mech 132(4):385–395

MacKay DJC (1992) A practical Bayesian framework for backprop networks. Neural Comput 4:448–472

Papadopoulos V, Giovanis DG, Lagaros ND, Papadrakakis M (2012) Accelerated subset simulation with neural networks for reliability analysis. Comput Methods Appl Mech Eng 223:70–80

Vamvatsikos D, Cornell CA (2002) Incremental dynamic analysis. Earthq Eng Struct Dyn 31(3):491–514

Vamvatsikos D, Cornell CA (2004) Applied incremental dynamic analysis. Earthq Spectra 20(2):523–553

Vamvatsikos D, Fragiadakis M (2010) Incremental dynamic analysis for estimating seismic performance sensitivity and uncertainty. Earthq Eng Struct Dyn 39(2):141–163