applied sciences

MDPI

Article

Optimal Design of Earthquake-Resistant Buildings Based on Neural Network Inversion

Carlo Calledda [1], Augusto Montisci [2,*] and Maria Cristina Porcu [1]

[1] Department of Civil-Environmental Engineering and Architecture, University of Cagliari, 09123 Cagliari, Italy; car.call86@gmail.com (C.C.); mcporcu@unica.it (M.C.P.)
[2] Department of Electric and Electronic Engineering, University of Cagliari, 09123 Cagliari, Italy
* Correspondence: augusto.montisci@unica.it

Abstract: An effective seismic design entails many issues related to the capacity-based assessment of the non-linear structural response under strong earthquakes. While very powerful structural calculation programs are available to assist the designer in the code-based seismic analysis, an optimal choice of the design parameters leading to the best performance at the lowest cost is not always assured. The present paper proposes a procedure to cost-effectively design earthquake-resistant buildings, which is based on the inversion of an artificial neural network and on an optimization algorithm for the minimum total cost under building code constraints. An exemplificative application of the method to a reinforced-concrete multi-story building, with seismic demands corresponding to a medium-seismicity Italian zone, is shown. Three design-governing parameters are assumed to build the input matrix, while eight capacity-design target requirements are assigned for the output dataset. A non-linear three-dimensional concentrated plasticity model of the structure is implemented, and time-history dynamic analyses are carried out with spectrum-consistent ground motions. The results show the promising ability of the proposed approach for the optimal design of earthquake-resistant structures.

Keywords: optimal structural design; earthquake-resistant buildings; inverse artificial neural network; non-linear dynamic analysis

Citation: Calledda, C.; Montisci, A.; Porcu, M.C. Optimal Design of Earthquake-Resistant Buildings Based on Neural Network Inversion. *Appl. Sci.* **2021**, *11*, 4654. https://doi.org/10.3390/app11104654

Academic Editor: Amadeo Benavent-Climent

Received: 3 May 2021
Accepted: 16 May 2021
Published: 19 May 2021

1. Introduction

The design of earthquake-resistant reinforced-concrete buildings is based on the concepts of ductility, strength hierarchy, and capacity design. Under exceptional seismic events buildings should deform plastically, exhibiting a deformation capacity greater than the demand in elements (typically beams) that are delegated for developing plastic hinges, while keeping more fragile elements (typically columns) safe. To assure this dissipative ductile behavior the code guidelines impose the fulfillment of ductility and strength hierarchy rules, which entail an iterative process for the more suitable choice of the geometrical and reinforcement design parameters. Commercial design programs help the designer to meet building code provisions and to check the fulfillment of rules, but an optimal cost-effective design is not usually ensured.

To design earthquake-resistant buildings, the present paper proposes a procedure, which is based on the inversion of an artificial neural network and on an optimization algorithm to achieve the minimum total cost under code constraints. The procedure is tested on a multistory reinforced-concrete building assumed to be built in a medium seismicity Italian area. Reference to the Italian Building code [1] and to the European Code EC8 [2] was made. Three design parameters are adopted as input variables to obtain about a thousand suitable combinations for the training dataset, while seven code checks are chosen as constraints to obtain the output matrix. A three-dimensional finite-element model of the building was developed and non-linear dynamic analyses under a suite of seven spectrum-consistent earthquakes were carried out. Among the methods

covered by the design regulations, the non-linear dynamic analysis is the most accurate to assess the seismic performance of ductile dissipative structures, although it requires higher computational burden and designer skill [3,4]. It can be successfully applied to both the design of new buildings [5] and the seismic retrofit of existing structures [6,7].

Artificial neural networks (ANNs) have been successfully applied for different purposes in Civil Engineering, from the early warning of ground settlements in urban areas [8] to the control of dynamic effects in bridges and the development of decision support systems for industrialized manufacturing of buildings [9]. ANNs have been also used to predict the seismic response of buildings [10,11] and even to perform an optimal seismic design [12]. A method based on an ANN is proposed in this paper for the code-compliant seismic design of reinforced concrete buildings. The novelty of the present approach is that of adopting an inversion of the ANN, which allows one to solve an inverse problem, that is to find the combination of design variables that meet given code-based performance constraints. Based on the ANN inversion procedure, an optimization algorithm is also presented in this paper which leads to a design configuration that minimizes the cost function. The neural network inversion is a non-standard tool which was exploited for pollution evaluation [13–16] and for electric capacitance tomography in nondestructive testing [17]. It was never applied, however, for structural design optimization.

To test the proposed method, a multistory reinforced-concrete building will be considered as a case-study. Based on non-linear dynamic analyses carried out under a suite of seven spectrum compatible earthquakes, an input matrix made of many combinations of three design parameters (corresponding to different building design configurations) and a target matrix containing seven performance variables calculated for each of the building configurations, will be obtained. The ANN will be trained on the input and the target datasets and the inversion algorithm will be then applied to find a design solution that falls within the intersection between the input and the output feasibility domains. Based on the trained network, an optimization algorithm will be applied to find a design configuration that minimizes the cost of the building while meeting the code-based seismic performance constraints chosen as target.

2. Case Study to Test the Procedure

2.1. Reference Multi-Story Building

A reinforced concrete (RC) multi-story frame is considered to test the procedure. Figure 1 provides a scheme of the building while Table 1 lists the main characteristics of the building. For the sake of simplicity, all beams are assumed to have the same section and reinforcement. Similarly, all columns are supposed to have the same squared section and reinforcement. The base lengths b_b of the beams is assumed to be equal to the base length b_p of the columns. Clay/cement mix slabs are considered for the floor and roof levels. Table 2 provides the dead loads G_1 and G_2 and the variable loads Q.

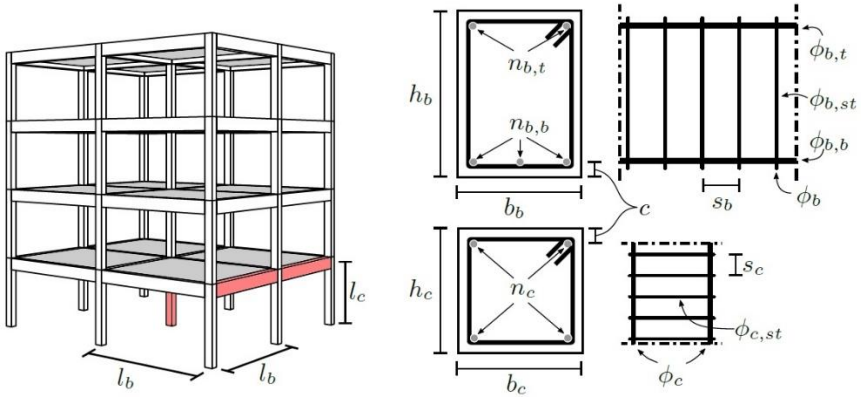

Figure 1. Reference building with its main characteristics.

Table 1. Main characteristics of the R/C structure.

Beams				**Columns**			
beam length	l_b =	5	m	column length	l_c =	3	m
section height	h_b =	x_1	cm	section height	h_c =	x_2	cm
section base	b_b =	x_2	cm	section base	b_c =	x_2	cm
concrete cover	d =	3	cm	concrete cover	d =	3	cm
top rebars [ϕ]	$\phi_{b,t}$ =	f(x_3)	mm	rebars [ϕ]	ϕ_c =	f(x_2)	mm
top rebars [n°]	$n_{b,t}$ =	f(x_3)	-	rebars [n°]	n_c =	f(x_2)	-
top rebars [%]	$\rho_{b,t}$ =	$0.6 \cdot x_3$	-	rebars [%]	ρ_c =	0.025	-
bottom rebars [ϕ]	$\phi_{b,bott}$ =	f(x_3)	mm				
bottom rebars [n°]	$n_{b,bott}$ =	f(x_3)	-				
bottom rebars [%]	$\rho_{b,b}$ =	x_3	-				
stirrup ϕ	$\phi_{b,st}$ =	10	mm	stirrup ϕ	$\phi_{p,st}$ =	12	mm
stirrup wheelbase	s_b =	60	mm	stirrup wheelbase	s_p =	80	mm

Materials		
	Characteristic Strength	**Young Modulus**
concrete	f_{ck} = 35 N/mm^2	E_{cm} = 40,600 N/mm^2
steel	f_{yk} = 450 N/mm^2	E_s = 200,000 N/mm^2

Table 2. Loads at the story level.

	G1 (N/m)	**G2 (N/m)**	**Q (N/m)**
Floor slab	2550.8	6410	2000
Roof slab	2432	1495	800

2.2. Design Parameters and Performance Variables

To obtain a database of cases for the training set, three parameters are taken as design parameters and referred to as x_1, x_2, and x_3. The first one is the base of beams and columns, namely $b_b = b_c = x_1$. The second one is the height of the beams $h_b = x_2$. Finally, the third variable is the ratio between the area of the tension longitudinal steel (typically placed at the bottom of the beam section) and the beam section area, namely $\rho_b = x_3$. The compression steel ratio was assumed to be $\rho_{b,comp} = 0.6x_3$. The ranges of variation of the design parameters are given in Table 3. A Phyton script [18] helped to vary the design parameters to obtain a dataset of examples. Eventually, 1011 cases were found, and a 3×1011 input matrix was obtained to be used for the training set.

Table 3. Ranges of variation of the design parameters.

Parameter	**Range of Variation**	**Increment Step**
$x_1 = b_b = b_c$	42 ÷ 78 cm	+2 cm
$x_2 = h_b$	42 ÷ 90 cm	+2 cm
$x_3 = \rho_b$	0.0058 ÷ 0.019	0.0013

Selected among those stated by the Italian Building Code [1], seven key structural checks were considered to build the performance target matrix. Based on the capacity design and meeting some strength hierarchy rules, these structural checks are expressed as difference between capacity and demand values and assigned to seven parameters y_i ($i = 1, 7$). They are provided in Table 4. The building was assumed to be designed in a high ductility class (DCH) at the life protection (LP) limit state. The values of parameters y_i calculated at the most demanded sections for each building's configuration will be stored in the 7×1011 target matrix.

Table 4. Structural checks for the network target matrix.

Performance Check [1]	Variables for the Target Matrix
Bending moment hierarchy rule	$y_1 = \sum M_{c,Rd} - \gamma_{Rd} \sum M_{b,Rd}$
Shear vs. moment on columns	$y_2 = \frac{\gamma_{Rd}\left(M^1_{c,Rd}+M^2_{c,Rd}\right)}{l_c} - V_{p,Ed}$
Beam mid-section bending moment	$y_3 = M_{b,mid,Rd} - M_{b,mid,\,Sd}$
Beam end-section bending moment	$y_4 = M_{b,end,Rd} - M_{b,end,Ed}$
Shear vs. moment on beams	$y_5 = V_{b,Rd} - \left(\frac{2\,\gamma_{Rd} M_{b,Rd}}{l_b} + \frac{(G+\psi_2 q) l_b}{2}\right)$
Shear on beam-column node (x direction)	$y_6 = 0.6\,\eta\, f_{cd}\, b_{jx}\, h_{jcx} \sqrt{1-\frac{N_{c,Ed}}{\eta}} - \left[\gamma_{Rd}\left(A_{b,1x}+A_{b,2x}\right) f_{yd} - V_{p,Ed}\right]$
Shear on beam-column node (z direction)	$y_7 = 0.6\,\eta\, f_{cd}\, b_{jz}\, h_{jcz} \sqrt{1-\frac{N_{c,Ed}}{\eta}} - \left[\gamma_{Rd}\left(A_{b,1z}+A_{b,2z}\right) f_{yd} - V_{p,Ed}\right]$

Symbols: M = bending moment; V = shear; γ = resistance factor; l = length; N = normalized axial load; h_{jc} = distance between column's longitudinal bars; b_j = length of the node; f = strength; A = area of bars; a = area of stirrups; G = gravity load; $\psi_2 q$ = reduced variable load; f_{cd} = concrete design strength; $\eta = \left(1 - \frac{f_{ck}}{250}\right)$ with f_{ck} in MPa. Subscripts and quotes: c = column; b = beam; d = design; R = resistance(capacity); y = yield; k = characteristic; st = stirrup; 1 = bottom section; 2 = top section; E = calculated (demand).

2.3. Response Spectrum and Spectrum-Consistent Earthquakes

The reference response spectrum for the Life Protection (LP) limit state is plotted in Figure 2, as obtained according to [1]. A medium-seismicity zone characterized by a peak ground acceleration of 1.6 m/s^2 (for the LP) was considered, with reference to the values provided by Annex B of the Italian Building Code [1]. Consistent with the spectrum in Figure 2, seven couples of records in the x and y directions are obtained through the software REXEL [19], according to code rules [1,2]. Earthquake name, date of occurrence, magnitude M_w, and distance Δ from the epicenter are listed in Table 5.

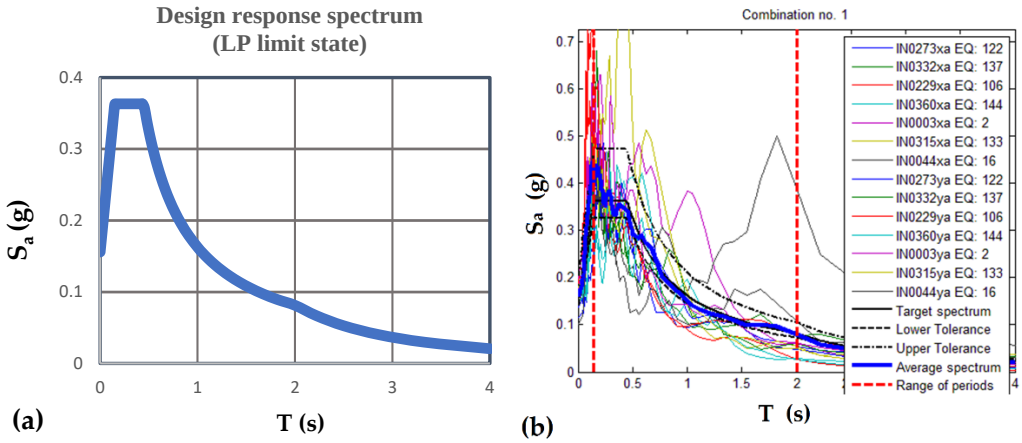

Figure 2. (**a**) Target response spectrum and (**b**) spectra of the records in the x and y directions.

Table 5. Earthquakes data.

Code	Earthquake	Date	Station ID	M_w	Δ (km)
EQ 122	MID NIIGATA PREF	11 March 2011	NIG022	6.2	23.05
EQ 137	Darfield	3 September 2010	RKAC	7.1	25.87
EQ 106	MID NIIGATA PREF	16 July 2007	NIG018	5.6	16.55
EQ 144	Christchurch	22 February 2011	SMTC	5.6	14.31
EQ 002	Near Niijima Island	1 July 2001	TKY010	6.2	20.03
EQ 133	EMILIA_Pianura_Padana	29 May 2012	SMS0	6.0	14.95
EQ 016	Mid Niigata Prefecture	23 October 2004	NIG018	6.6	28.44

2.4. Non-Linear Dynamic Analysis on an FEM Model

The Open System for Earthquake Engineering Simulation (OpenSees) software framework [18] was adopted to develop the numerical analyses. A 3D finite element model of the frame was developed in OpenSees by considering force-based beam-column elements. A scheme of the model, with elements and nodes, is provided in Figure 3. A Rayleigh classic damping was assumed in the elastic range, where a constant damping factor $\xi = 5\%$ was considered for each mode. A distributed plasticity (fiber-based) was assumed within the beam–column elements. The non-linear Scott-Park [20] and Giuffrè-Menegotto-Pinto [21] stress–strain constitutive models were adopted for concrete and steel, respectively, as implemented in OpenSees [18]. The qualitative scheme of the two models is provided in Figure 4, while the values of the parameters are provided in Tables 6 and 7.

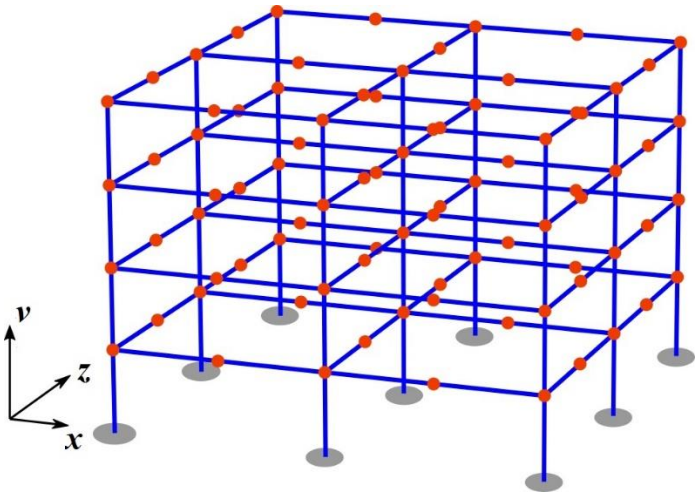

Figure 3. FEM model of the building with beam–column elements (blue) and nodes (orange).

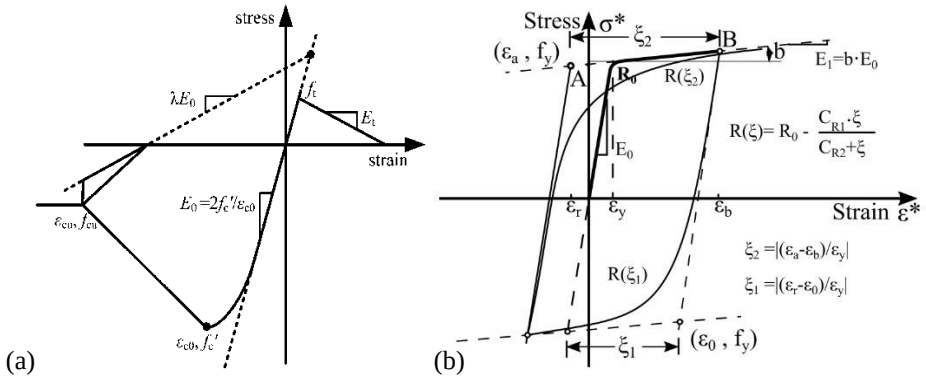

Figure 4. Constitutive stress–strain models for (**a**) concrete and (**b**) steel, refer to Tables 6 and 7.

Table 6. Values adopted for the constitutive model of concrete (see Figure 4a).

	f_c (MPa)	ε_{c0}	f_{cu} (MPa)	ε_{cu}	λ	f_t (MPa)	E_t (MPa)
Unconfined	35	0.0294	7.02	0.0016	0.1	2	1500
Confined		f (x_1, x_2, x_3) f (x_1, x_2, x_3) f (x_1, x_2, x_3) f (x_1, x_2, x_3)			0.1	2	1500

Table 7. Values adopted for the constitutive model of reinforcement steel (see Figure 4b).

f_y (MPa)	E_0 (MPa)	B_s	R_0	C_{R1}	C_{R2}	$a_1 = a_3$	$a_2 = a_4$
450	210,000	0.005	15	0.925	0.15	0	1

Seven non-linear time-history analyses were carried out for each one of the 1011 building's configurations under the seven spectrum-consistent earthquakes introduced in Section 2.3 (two orthogonal records are simultaneously applied to the building for each earthquake). A total of 7077 time-history non-linear analyses were finally carried out.

The averaged values of the effects over the seven analyses are calculated, this being allowed by codes [1,2] and the values of the seven parameters y_i ($i = 1,7$) defined in Table 4 are obtained for all of the involved elements of the structure. The lower the value of y_i the more demand on the element. The lowest values of y_i found for each of the 1011 cases are stored in the 7×1011 target matrix. It can be noted that, almost always, the most demanded elements of the case-study building were the column and the beam colored in red in Figure 1.

3. Network Inversion for Structural Design

Artificial Neural Networks (ANNs [22]) represent the smart core of the Decision Support System (DSS) presented in this paper. ANNs are analytical structures the working of which depends on both their topology and a set of parameters. Several paradigms of ANNs exist referring to the topology, the kind of problem to handle, and the time dependency. The most popular type of ANNs is the Multi-Layer-Perceptron (MLP, Figure 5), which can be used to solve both Classification and nonlinear Regression problems. MLPs have a unidirectional structure with no feedbacks. The layout consists of an input layer of nodes (neurons), one or more intermediate (hidden) layers, and an output layer. Whatever its type, an ANN typically leads to the creation of a model of a physical system starting from the data, rather than from the knowledge of the analytical relationship among the variables. In particular, the aim of MLPs is to imitate the relationship between independent (input) and dependent (output) variables that describe the physical system. For this purpose, a training process is performed to calculate the parameters of the ANN (weights), starting from a randomly assigned set of values. The training consists of an iterative procedure where the minimum of a performance function is sought. Most parts of used algorithms are first or second order minimization procedures, where the performance function is typically represented by the mean squared error of the network, the error being the gap between the output and the measured (target) values.

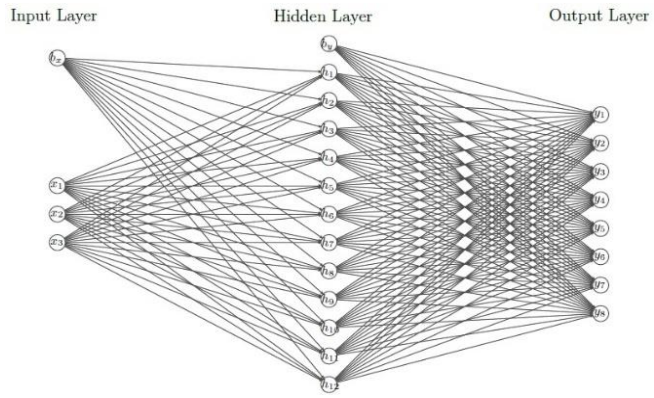

Figure 5. Multilayer Perceptron Layout.

The training process could be a challenging task since the performance depends on choosing an appropriate training set of examples, adopting a suitable layout of the MLP

and assuming an efficient training strategy. Usually a trial and error procedure is performed to determine the best set of these hyper parameters.

3.1. Inversion Algorithm

A representation of the relationship between inputs and outputs is conveyed in the trained MLP. This information can be exploited to solve the inverse problem, that is to determine the design parameters which meet the set performance requirements [14–17,23–27]. Since the design parameters represent the input and the performance of the physical system is the output, finding the input corresponding to a given output means determining the design parameters which guarantee the fulfilment of the given requirements. The model frozen inside the MLP is represented by a set of equations (Equation (1)), which describe the relationship between the input $\boldsymbol{x}$ and the output $\boldsymbol{y}$ of the network, see Figure 5.

To disentangle the algebraic structure of the MLP, two auxiliary variables, namely $\boldsymbol{k}$ and $\boldsymbol{h}$, are introduced. They represent respectively the input and the output of the hidden layer. Equation (1) (a) describes the linear relation between the output of the hidden layer and the output of the MLP ($\boldsymbol{h}$ and $\boldsymbol{y}$, respectively). Equation (1) (b) relates the input and the output of the hidden layer ($\boldsymbol{k}$ and $\boldsymbol{h}$, respectively), this being a non-linear relation. Finally, Equation (1) (c) describes the linear relation between the input of the MLP and the input of the hidden layer ($\boldsymbol{x}$ and $\boldsymbol{k}$, respectively). Once the MLP has been trained, Equation (1) allow to propagate the input $\boldsymbol{x}$ up to the output $\boldsymbol{y}$, which leads to calculate the performance of the system corresponding to a given set of design parameters. On the other hand, if the output variable $\boldsymbol{y}$ is set, the three Equation (1) can be solved in series to obtain the input $\boldsymbol{x}$, which means finding the design parameters that correspond to the given performance.

$$\begin{array}{ll} (a) & \\ (b) & \\ (c) & \end{array} \quad \begin{cases} \boldsymbol{W}_y \cdot \boldsymbol{h} + \boldsymbol{b}_y = \boldsymbol{y} \\ \boldsymbol{h} = \sigma(\boldsymbol{k}) \\ \boldsymbol{W}_x \cdot \boldsymbol{x} + \boldsymbol{b}_x = \boldsymbol{k} \end{cases} \tag{1}$$

In general, a domain of feasibility can be set, rather than requiring the fulfillment of a given value of the performance. For the sake of simplicity, in this formulation such domain is assumed to be linear and convex, so that it can be expressed by means of the set of inequalities given by Equation (2).

$$\boldsymbol{A} \cdot \boldsymbol{y} \leq \boldsymbol{a} \tag{2}$$

The feasibility domain of the output, introduced in Equation (2), sets the requirements of the building. The constraints on the output correspond to as many constraints on the input, namely the design parameters. Thanks to Equation (1), we can transfer the feasibility domain from the output space to the input space. In other words, the requirements are translated into a feasibility domain of the design parameters. In their turn, a set of bounds will be stated on the design parameters, defining a feasibility domain of the input. Solving the inverse problem means finding a design solution that falls within the intersection between the two feasibility domains, respectively defined on the input and on the output.

As described in Figure 6, we will make use of the following four geometrical spaces: Input Space X where the design parameters are defined, Upstream Hidden Space K representing the input of the hidden layer, Downstream Hidden Space H representing the output of the hidden layer, Output Space that is the output of the network. Equation (1) is subdivided into three subsystems: a linear equations system that relates space X with space K, a nonlinear equations system that relates space K with space H, and, finally, a linear equations system that relates space H with the output space Y.

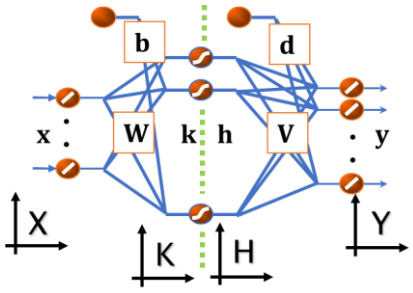

Figure 6. Nomenclature of the MLP algebraic structure.

Equation (1) allows us to project both points and domains from one space to any other of the structures. In particular, the feasibility domain of the output, expressed by Equation (2), namely the performance requirements of the building, can be projected into space H by substituting the output variable $\boldsymbol{y}$ as derived from Equation (1) (a) in Equation (2):

$$(\boldsymbol{A}\cdot\boldsymbol{W}_y)\cdot\boldsymbol{h} \leq \boldsymbol{a} - \boldsymbol{A}\cdot\boldsymbol{b}_y \tag{3}$$

Equation (3) denotes a constraint for variable $\boldsymbol{h}$ deriving from the feasibility domain of the output. At the same time, variable $\boldsymbol{h}$ is limited by the range of the nonlinear activation function of the hidden neurons. Such activation function typically has a sigmoidal shape, and has a range in the interval $[-1, 1]$. In this work, the hyperbolic tangent function is assumed (see Figure 7).

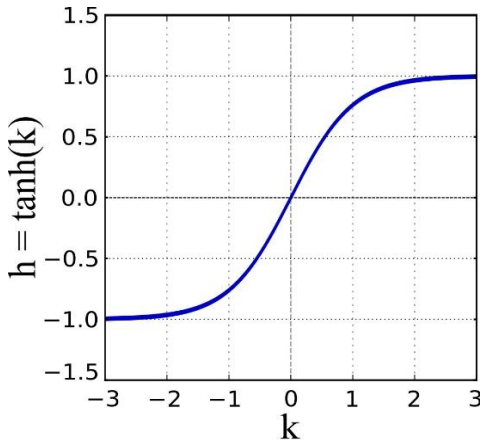

Figure 7. Sigmoidal activation function of the hidden neurons.

Vector $\boldsymbol{h}$ corresponding to the sought solution must be attainable from a feasible $\boldsymbol{k}$, the feasibility range being $-1+\varepsilon \leq h \leq 1-\varepsilon$, where ε is a margin from the saturation of the activation function. We can re-write this equation in the following form:

$$\begin{cases} \boldsymbol{h} \leq 1-\boldsymbol{\varepsilon} \\ -\boldsymbol{h} \leq 1-\boldsymbol{\varepsilon} \end{cases} \tag{4}$$

The two systems (3) and (4) can be included in a unique system as:

$$\begin{bmatrix} \boldsymbol{A}\cdot\boldsymbol{W}_y \\ \boldsymbol{I} \\ -\boldsymbol{I} \end{bmatrix} \cdot\boldsymbol{h} \leq \begin{bmatrix} \boldsymbol{a} - \boldsymbol{A}\cdot\boldsymbol{b}_y \\ 1-\boldsymbol{\varepsilon} \\ 1-\boldsymbol{\varepsilon} \end{bmatrix} \; or \; \boldsymbol{M}\cdot\boldsymbol{h} \leq \boldsymbol{m} \tag{5}$$

A first check is required to establish if the domain (5) is empty. In this case, no design solution exists which meets the requirements, and the requirements should be relaxed.

The domain (5) can be projected on space K by means of Equation (1) (b):

$$\boldsymbol{M}\cdot\sigma(\boldsymbol{k}) \leq \boldsymbol{m} \tag{6}$$

Finally, the domain (6) can be projected on the input space

$$\boldsymbol{M}\cdot\sigma(\boldsymbol{W}_x\cdot\boldsymbol{x} + \boldsymbol{b}_x) \leq \boldsymbol{m} \tag{7}$$

Equation (7) describes a nonlinear domain in the input space X.

Generally, a feasibility domain is a priori stated on the input space. Often, this domain contains a feasibility range for each design parameter, making the feasibility domain to be a box: $\boldsymbol{L}_{bnd} \leq \boldsymbol{x} \leq \boldsymbol{U}_{bnd}$. More in general, the feasibility domain of the input could be defined by a set of constraints of any kind. For the sake of simplicity, in this work we assumed that such constraints are linear, thus, the domain can be expressed as:

$$\boldsymbol{V}\cdot\boldsymbol{x} \leq \boldsymbol{v} \tag{8}$$

The solution of the problem must fall within both the feasibility domains (7) and (8). Therefore, the design problem is transformed into an existence problem. Unfortunately, one of the two systems is nonlinear, therefore finding a feasible solution could be a challenging task.

The existence problem can be solved iteratively making use of the linear programming (LP) [28]. The number of neurons of the hidden layer usually is larger than the number of inputs, which implies that the number of Equation (1) (c) is larger than the number of the input variables, thus the equations system is overdetermined. By means of the pseudoinverse matrix of Moore–Penrose, Equation (1) (c) can be solved with respect to $\boldsymbol{x}$ according to the least square error criteria:

$$\boldsymbol{x} = \left(\boldsymbol{W}_{\boldsymbol{x}}^T\boldsymbol{W}_x\right)^{-1}\boldsymbol{W}_{\boldsymbol{x}}^T\cdot(\boldsymbol{k} - \boldsymbol{b}_x) \tag{9}$$

By substituting Equation (9) into Equation (8), the feasibility domain of the input is projected in space K, which is still linear:

$$\boldsymbol{V}\cdot\left(\boldsymbol{W}_{\boldsymbol{x}}^T\boldsymbol{W}_x\right)^{-1}\boldsymbol{W}_{\boldsymbol{x}}^T\cdot(\boldsymbol{k} - \boldsymbol{b}_x) \leq \boldsymbol{v} \tag{10}$$

By exploiting Equation (1) (b), any point from space K can be projected to space H and vice-versa. The feasibility domain of the input is linear in space K and nonlinear in space H, while the feasibility domain of the output is linear in space H and nonlinear in space K. Therefore, an iterative procedure can be defined, where the current solution is projected alternatively on the two domains changing space K and H, so that the domain where the solution should be projected is always linear.

4. Application of the Inversion Algorithm to the Case-Study

With reference to the case-study described in Section 2, an ANN is firstly trained over a dataset of examples and then the inversion algorithm described in Section 3.1 is applied to find a set of design parameters meeting the chosen code-compliant seismic performance requirements.

4.1. Training Phase

A population of 1011 examples was considered for the training phase, given by an input matrix and a target matrix. The input matrix M_x stored 1011 sets of the design parameters x_1, x_2, and x_3 defined in Section 2.2, all of them falling within the ranges provided in Table 3. Corresponding to the above sets of design parameters, a 7×1011 output target matrix M_y was obtained, storing the seven values of the capacity-design performance variables y_i (see Table 3) calculated through numerical non-linear FEM analyses under spectrum consistent earthquakes (as described in Section 2.4) for each of the 1011 cases. It

should be stressed that the output performance variables y_i ($i = 1, 7$) were not asked to meet the code-compliant constraints. This means that the population of examples also includes cases of unsafe seismic designs of the building, as well as sets of design parameters out of the feasibility domain.

The 1011 examples dataset was divided into three subsets: a training set (80% of the population), a validation set (10% of the population), and a test set (10% of the population). The training process was applied to the training set to find the connections' weights. The verification set was involved in the training process but only to monitor the evolution of the error, this set having no influence on the calculus of the connections' weights. The test set is not involved in the training process, but it was used to monitor the generalization capability of the trained neural network.

4.2. Inversion Phase

Once the network was trained, the inversion algorithm was applied. The design constraints on the output variables (Equation (2)) were in this case given by the following set of inequalities imposed over the seven performance variables y_i defined in Table 4, according to the Italian Building code [1]:

$$y_1 \geq 0, \;\; y_2 \geq 0, \;\; y_3 \geq 0, \;\; y_4 \geq 0, \; y_5 \geq 0, \; y_6 \geq 0, \;\; y_7 \geq 0 \tag{11}$$

The minimum number of neurons of the hidden layer needed to guarantee the required precision was found to be 12. The inversion procedure converged after 88 iterations, (see Figure 8).

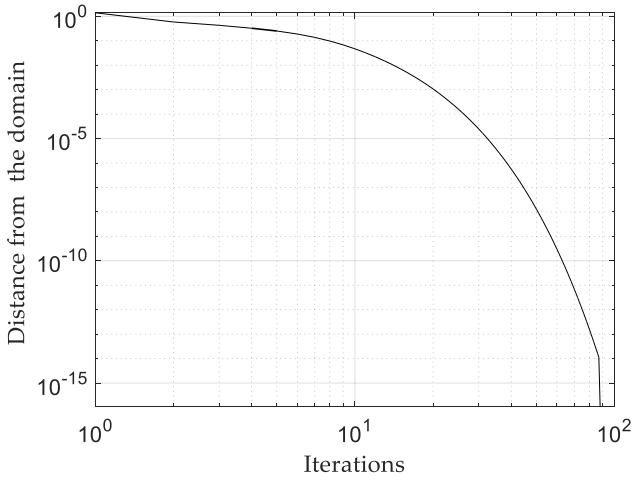

Figure 8. Distance from the domain during the inversion procedure.

As a result of the inversion algorithm, the set of design parameters listed in the first row of Table 8 was found. To check the validity of the procedure, a numerical analysis on the FEM model of the building was performed. In fact, integer values, rounded up, were considered for the section's dimensions x_1 and x_2, while the values of the steel percentage x_3 was rounded up by considering commercial steel diameters. The second row of Table 8 gives the values of the design parameters adopted in the numerical model of the building for the comparison test. Non-linear time-history analyses under the suite of seven earthquakes were carried out, the averaged effects were calculated and the numerical values of the variables y_i were finally obtained. They are given in Table 9. It can be noted that the values of y_i ($i = 1, 7$) meet the code-compliant constraints (11).

Table 8. Design parameters from the inversion procedure and used for the numerical test.

	x_1 (m)	x_2 (m)	x_3 (-)
Inversion values	0.5335	0.6023	0.0077
FEM model values	0.54	0.60	0.0077

Table 9. Performance variables from the FEM model with the design parameters in Table 7.

y_1 (Nm)	y_2 (N)	y_3 (Nm)	y_4 (Nm)	y_5 (N)	y_6 (N)	y_7 (N)
0.83×10^5	0.29×10^5	6.13×10^5	6.20×10^5	3.75×10^5	4.32×10^5	4.32×10^5

5. Optimization Procedure for the Case-Study

The solution found in the previous section fulfils the requirements of both Input (design parameters) and Target (performance of the building). While meeting these specifications, it is generally possible to improve the solution found according to predefined target functions. Since the compliance with the target is expressed in terms of thresholds, the margins for optimization concern the design parameters, which are linked to the cost of the RC frame. In the case-study example, beam and column base $b_b = b_c = x_1$, beam height $h_b = x_2$, and beam tension longitudinal steel ratio $\rho_b = x_3$, have been assumed as design parameters. Related to these parameters, the following cost function was defined:

$$C = n_b l_b \left(\gamma_{con}\, c_{con} x_1 x_2 + \gamma_{ste}\, c_{ste}\, x_1\, x_2\, 1.6\, x_3\right) + n_c l_c \left(\gamma_{con}\, c_{con}\, x_2^2 + \gamma_s\, c_s\, \rho_c\, x_2^2\right) \quad (12)$$

Here n_b and n_c denote the number of beams and columns, l_b and l_c being their longitudinal length, respectively; γ_{con} and γ_{ste} are the specific weight of concrete and steel; c_{con} and c_{ste} the cost per unit weight of concrete and steel; while ρ_c is the longitudinal steel ratio in columns.

By introducing the following parameters:

$$\alpha = n_b l_b\, \gamma_{con}\, c_{con}\ ;\ \beta = 1.6\, l_b\, \gamma_{ste}\, c_{ste}\ ;\ \delta = n_c l_c (\gamma_{con}\, c_{con} + \gamma_{ste}\, c_{se}\, \rho_c) \quad (13)$$

the cost function writes:

$$C = \alpha\, x_1 x_2 + \beta\, x_1\, x_2\, x_3 + \delta\, x_2^2 \quad (14)$$

The numerical values of the coefficients α, β, and δ for the case under study are provided in Table 10, the values of the cost function being directly expressed in euros (€).

Table 10. Values of coefficients α, β, and δ.

α (€/m^2)	β (€/m^2)	δ (€/m^2)
60,720	4,823,040	15,357.6

To find the set of parameters x_1, x_2, and x_3 that minimize function C, the gradient descent algorithm is here adopted. This algorithm is based on an iterative procedure that moves in the negative direction of the function gradient ∇C from an initial set of parameter values, say point X_i, toward a set of final parameter values that minimize the cost function. Each iteration step is governed by the rule:

$$X_{(i+1)} = X_i - \eta \nabla C \quad (15)$$

where η is the step size, taken initially equal to 1×10^{-6} and then updated at each iteration.

According to (14), the gradient ∇C writes:

$$\frac{\partial C}{\partial x_1} = \alpha\, x_2 + \beta\ x_2\, x_3;\ \frac{\partial C}{\partial x_2} = \alpha\, x_1 + \beta\ x_1\, x_3 + 2\delta\, x_2;\ \frac{\partial C}{\partial x_3} = \beta\ x_1\, x_2 \quad (16)$$

The solution provided by the inversion algorithm (first row of Table 8) was chosen as a starting set of design parameters. The optimization algorithm returns the set of design parameters listed in Table 11, which corresponds to a point on the frontier of the domain where the Karush–Kuhn–Tucker condition is met [29]. The set of design parameters were assigned to the FEM model and the results of the dynamic analyses lead to the values of the performance variables given in Table 12. Such values confirm that the building designed with the optimum set of design parameters meets the capacity design requirements assumed as target.

Table 11. Design parameters from the optimization procedure.

	x_1 (m)	x_2 (m)	x_3 (-)
Optimum design values	0.42	0.42	0.0058

Table 12. Performance variables from the FEM model with the optimal design parameters.

y_1 (Nm)	y_2 (N)	y_3 (Nm)	y_4 (Nm)	y_5 (N)	y_6 (N)	y_7 (N)
0.94×10^5	0.17×10^5	1.71×10^5	1.77×10^5	3.60×10^5	2.93×10^5	2.93×10^5

The trend of the cost function during the optimization procedure is plotted in Figure 9. The optimal point was reached after five iterations, with a final cost of EUR 18,323 over the initial cost of EUR 26,664. It is worth noting that other categories (labor, transport, and non-structural components) contribute to the total cost of the construction. For the sake of simplicity, however, reference only to the naked RC frame was made in the optimization procedure here presented. A more comprehensive study should of course account for all the categories of cost, even if it would not affect the validity of the method proposed. It should be noted finally that the design parameters found by the optimization algorithm do not belong to the dataset of training examples.

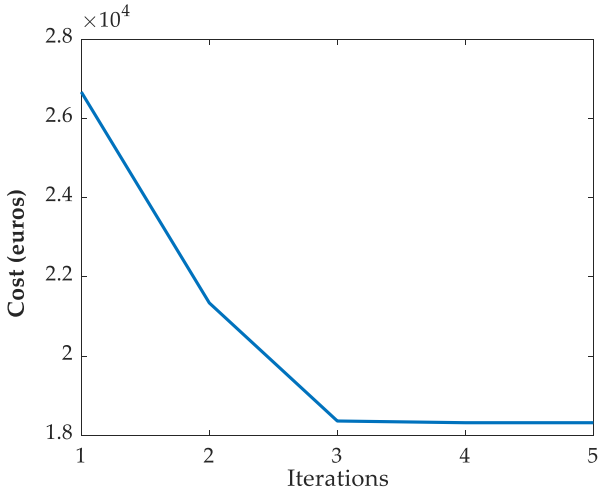

Figure 9. Optimization of the cost function.

6. Conclusions

An Artificial Neural Network (ANN) inversion procedure was proposed in this paper to obtain an optimal code-compliant seismic design of reinforced-concrete buildings. This procedure allows to solve an inverse problem, finding the design parameters which guarantee the fulfilment of some given code-compliant performance requirements while minimizing the cost function.

The procedure was applied to a multistory three-dimensional (3D) building supposed to be built in a medium-seismicity Italian area. A set of 1011 different configurations of the

building were obtained by varying three design parameters in suitable ranges. A 3 × 1011 input matrix storing the considered sets of design parameters was thus obtained. Non-linear dynamic analyses on 3D finite element models of the 1011 building's configurations under a suite of seven spectrum compatible earthquakes were then carried out. The values of seven capacity-design performance variables calculated at the most demanded sections for each of the building configurations were then stored in the 7 × 1011 target matrix. The ANN was trained on the input and target datasets and the inversion algorithm was then applied to find a design solution that fell into the intersection between the input and the output feasibility domains. As checked by means of the FEM numerical analyses, this led to a design solution meeting the code-compliant performance requirements chosen as the target.

An optimization iterative procedure was finally proposed to find a set of design parameters that minimized the cost of the building while meeting the code-based seismic performance constraints. The optimization procedure was based on the trained ANN and applied a gradient descent algorithm, which iteratively moved in the negative direction of the cost function gradient, starting from the set of design parameters found through the inversion procedure and stopping at a set of final values of design parameters that minimized the cost function. The seismic performance of the building corresponding to the design parameters provided by the optimization procedure was checked by means of the FEM numerical analyses, finding that it met the seven capacity-design requirements assumed as target control parameters. It should be noted that, for the sake of simplicity, only three design parameters were assumed to obtain the building configurations while only some of the performance checks stated by the Italian Building code were taken as indexes of the building's structural safety in the present application of the method. However, the method can be easily extended to account for any number of design parameters in the input matrix and any number of code-based performance checks in the target matrix.

Author Contributions: Conceptualization and research design, A.M., M.C.P. and C.C.; methodology, A.M. and M.C.P.; software, C.C.; investigation, C.C., A.M. and M.C.P.; pre and postprocessing results, C.C., A.M. and M.C.P.; writing-review and editing, M.C.P. and A.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in the paper are available upon request.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. *DM 17 Gennaio 2018, Aggiornamento delle Norme Tecniche per le Costruzioni*; Ministero delle Infrastrutture e dei Trasporti: Rome, Italia, 2018.
2. *EN 1998-1: 2004: Design of Structures for Earthquake Resistance—Part 1: General Rules, Seismic Actions and Rules for Buildings*; Bruxells, European Committee for Standardization: Brussels, Belgium, 2005.
3. Porcu, M.C. Code inadequacies discouraging the earthquake-based seismic analysis of buildings. *Int. J. Saf. Secur. Eng.* **2017**, *7*, 545–546. [CrossRef]
4. Vielma, J.C.; Porcu, M.C.; López, N. Intensity Measure Based on a Smooth Inelastic Peak Period for a More Effective Incremental Dynamic Analysis. *Appl. Sci.* **2020**, *10*, 8632. [CrossRef]
5. Porcu, M.C.; Bosu, C.; Gavrić, I. Non-linear dynamic analysis to assess the seismic performance of cross-laminated timber structures. *J. Build. Eng.* **2018**, *19*, 480–493. [CrossRef]
6. Porcu, M.C.; Vielma, J.C.; Panu, F.; Aguilar, C.; Curreli, G. Seismic retrofit of existing buildings led by non-linear dynamic analyses. *Int. J. Saf. Secur. Eng.* **2019**, *9*, 201–212. [CrossRef]
7. Vielma-Perez, J.-C.; Porcu, M.C.; Gómez, M.A. Non-Linear analyses to assess the seismic performance of RC buildings retrofitted with FRP. *Rev. Int. Métodos Numér. Cálculo Diseño Ing.* **2020**, *36*. [CrossRef]
8. Montisci, A.; Porcu, M.C. Self-Organizing-Map Analysis of InSAR Time Series for the Early Warning of Structural Safety in Urban Areas. In *International Conference on Computational Science and Its Applications*; Springer: Cham, Germany, 2020; pp. 864–876.

9. Flood, I. Towards the next generation of artificial neural networks for civil engineering. *Adv. Eng. Inform.* **2008**, *22*, 4–14. [CrossRef]
10. Brown, A.S.; Yang, H.T.; Wrobleski, M.S. Improvement and Assessment of Neural Networks for Structural Response Prediction and Control. *J. Struct. Eng.* **2005**, *131*, 848–850. [CrossRef]
11. Conte, J.P.; Durrani, A.J.; Shelton, R.O. Seismic Response Modeling of Multi-Story Buildings Using Neural Networks. *J. Intell. Mater. Syst. Struct.* **1994**, *5*, 392–402. [CrossRef]
12. Möller, O.; Foschi, R.O.; Quiroz, L.M.; Rubinstein, M. Structural optimization for performance-based design in earthquake engineering: Applications of neural networks. *Struct. Saf.* **2009**, *31*, 490–499. [CrossRef]
13. Cannas, B.; Carcangiu, S.; Fanni, A.; Farley, T.; Militello, F.; Montisci, A.; Pisano, F.; Sias, G.; Walkden, N. Towards an automatic filament detector with a Faster R-CNN on MAST-U. *Fusion Eng. Des.* **2019**, *146*, 374–377. [CrossRef]
14. Foddis, M.L.; Ackerer, P.; Montisci, A.; Uras, G. ANN-based approach for the estimation of aquifer pollutant source behaviour. *Water Supply* **2015**, *15*, 1285–1294. [CrossRef]
15. Foddis, M.L.; Ackerer, P.; Montisci, A.; Uras, G. Polluted aquifer inverse problem solution using artificial neural networks. *AQUA Mundi* **2013**, *4*, 15–21.
16. Foddis, M.L.; Matzeu, A.; Montisci, A.; Uras, G. The Arborea plain (Sardinia-Italy) nitrate pollution evaluation. *Ital. J. Eng. Geol. Environ.* **2017**, *1*, 67–76.
17. Carcangiu, S.; Fanni, A.; Montisci, A. Electric capacitance tomography for nondestructive testing of standing trees. *Int. J. Numer. Model. Electron. Netw. Devices Fields* **2019**, *32*, 2252. [CrossRef]
18. Zhu, M.; McKenna, F.; Scott, M.H. OpenSeesPy: Python library for the OpenSees finite element framework. *SoftwareX* **2018**, *7*, 6–11. [CrossRef]
19. Iervolino, I.; Galasso, C.; Cosenza, E. REXEL: Computer aided record selection for code-based seismic structural analysis. *Bull. Earthq. Eng.* **2010**, *8*, 339–362. [CrossRef]
20. Bunni, N.G.; Scott, B.D. Stress-Strain behavior of concrete confined by overlapping hoops at low and high-strain rates-discussion. *J. Am. Concr. Inst.* **1982**, *79*, 13–27.
21. Menegotto, M. Method of Analysis for Cyclically Loaded RC Plane Frames Including Changes in Geometry and Non-Elastic Behavior of Elements under Combined Normal Force and Bending. In *Proc. of IABSE Symposium on Resistance and Ultimate Deformability of Structures Acted on by Well Defined Repeated Loads*; IABSE: Zurich, Switzerland, 1973.
22. Hertz, J.A. *Introduction to the Theory of Neural Computation*; CRC Press: Boca Raton, FL, USA, 2018.
23. Foddis, M.L.; Montisci, A.; Trabelsi, F.; Uras, G. An MLP-ANN-based approach for assessing nitrate contamination. *Water Supply* **2019**, *19*, 1911–1917. [CrossRef]
24. Cau, F.; Di Mauro, M.; Fanni, A.; Montisci, A.; Testoni, P. A Neural Networks Inversion-Based Algorithm for Multiobjective Design of a High-Field Superconducting Dipole Magnet. *IEEE Trans. Magn.* **2007**, *43*, 1557–1560. [CrossRef]
25. Carcangiu, S.; Fanni, A.; Montisci, A. Multi Objective Optimization Algorithm Based on Neural Networks Inversion. In *Bio-Inspired Systems: Computational and Ambient Intelligence*; Springer: Berlin, Germany, 2009.
26. Carcangiu, S.; Montisci, A. A Locally Recurrent Neural Network Based Approach for the Early Fault Detection. In Proceedings of the 2018 IEEE 4th International Forum on Research and Technology for Society and Industry (RTSI), Palermo, Italy, 10–13 September 2018; IEEE: Piscataway, NJ, USA, 2018.
27. Carcangiu, S.; Fanni, A.; Montisci, A. A constructive algorithm of neural approximation models for optimization problems. *COMPEL Int. J. Comput. Math. Electr. Electron. Eng.* **2009**, *28*, 1276–1289. [CrossRef]
28. Bazaraa, M.S.; Jarvis, J.J.; Sherali, H.D. *Linear Programming and Network Flows*; Wiley: Hoboken, NJ, USA, 2009.
29. Kuhn, H.W.; Tucker, A.W. Nonlinear Programming. In Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability, Berkeley, CA, USA, 31 July–12 August 1951.