Artificial Intelligence for the Analysis of Structures in Civil Engineering

Nuno Pinhão
Instituto Superior Técnico, Universidade de Lisboa
Lisboa, Portugal
nuno.bras.pinhao@tecnico.ulisboa.pt

Abstract

The analysis and design of structures rely on numerical models that are computationally very expensive in the general nonlinear case. In recent years machine learning has been used to approximate functions in many domains, so there is a possibility to use it to approximate the structure responses of new designs of structures in civil engineering. In this work, we explore data driven approaches to calculate the responses of new designs of beam structures. We consider linear and nonlinear beam models, investigate different neural networks architectures, calculate relevant structure responses and behaviours, and validate it in relation to precise numerical simulations. Our results show that neural networks can approximate the behaviour of realistic beam structures to predict the moments, tensions, and maximum load, all this 1000x faster than the corresponding numerical simulation. This work can be used as a tool to quickly help an engineer to validate several variants of preliminary designs of new structures before committing to a long precise numerical simulation.

1 Introduction

Machine Learning has proven to be efficient and effective in approximating very complex problems such as function regression, pattern recognition or time series prediction [Lopez *et al.*, 2008] and when enough data is available, Neural Networks are extremely useful at complex nonlinear problems [Chaudhary *et al.*, 2007]. Compared with more traditional machine learning techniques, Neural Networks do not require as much effort in feature selection and input processing since they learn through their own errors.

In civil engineering, specially in structure design, very complex and computationally expensive simulations are necessary to optimize the structure parameters to achieve a good efficiency in terms of safety, constraint satisfaction, and reduced cost. Only recently have authors considered the idea of complementing the slow and precise numerical simulations with approximate and fast learned solutions[Salehi and Burgueno, 2018]. This combination has the potential to allow an engineer to pre-validate hundreds of designs in a matter of minutes instead of days.

1.1 Motivation

The analysis and design of civil engineering structures span a wide range of complex phenomena, corresponding to non-linear behaviours. Those behaviours are evaluated through sophisticated numerical models, which require some computational effort. Hence, in a preliminary design of a structure simplified methods are applied (some times hand calculations), which have a limited accuracy. However, when dealing with non-linear materials or forms on the structure, complex non-linear equations need to be solved. Conventional methods of solving these equations such as FEM (Finite Element Method), can be very time consuming and require a lot of computational effort.

This work aims at finding out whether neural networks can be an effective alternative for approximate the structure responses to guide the preliminary design and study of new structures. With a precise and accurate prediction of the structure responses and behaviours, we could estimate the safety of a structure scenario without having to calculate the complex non linear equations, therefore improving significantly the computational time and effort when laying out a new structure design.

In particular, for this work, we will study different types of beam designs. Beams are one of the main building block of all the structures, being present in bridges and high-rise buildings. We study beams under different types of loads and find ways to model this analysis problem as a regression problem. We then need to create a dataset of relevant structure responses, such as bending moments, displacements, rotations and curvatures from those beams structures to then train neural networks. After having a learned model, we need to study how that learned neural networks can be used to extract information such as bending moments, rotation, and maximum load.

1.2 Contributions:

Our contributions are:

- study of neural networks to approximate
- use of neural networks predict maximum load

2 Related Work

2.1 Neural Networks

Biology inspiration

Artificial Neural Networks are inspired by the way the human brain works. A human brain can process enormous amounts of information using the data obtained by the human senses (touch, sight, hearing, smell and taste). This data is processed through neurons, which pass electrical signals through them. The human brain can learn simply because synapses can adapt, by changing its number of vesicles or the number of receptor molecules, synapses change their effects and adapt their own weights. This can make the human brain learn to perform every kind of activity such as recognizing objects, understand language, movements of the body. Much like Human neural networks, artificial neural networks also learn their functions by adapting weights.

Definition and Structure

An Artificial Neural Network is composed by at least three layers: the input layer which is the information represented in nodes that we give to our model; the output layer that is the computed output the network predicted giving the input; and the hidden layer, which can be more than one, the more hidden layers a network has the deeper the network is.

Each Layer is made of a certain number of nodes or neurons, these neurons represent values and transmit their values as signals to the neurons in the next layer that they are connected to. Neurons use an Activation Function to compute their own value using the values that were signalled to them from previous neurons, their own weight, and bias values. The connection between the Neurons can also vary, in most cases a NN (Neural Network) is densely connected, meaning every neuron is connected to every neuron in the adjacent layer.

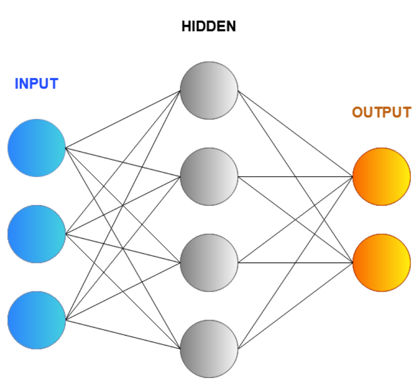

Figure 1: Neural Network Structure

Neural Networks Architecture

A Neural Network's architecture represents its number of hidden layers (depth), number of nodes or neurons in each layer (width), the connections between nodes and layers and the activation functions used in the layers to determine connection weights [Fausett, 2008].

Choosing the optimal number of hidden layers and nodes is one of the most difficult tasks when building a NN, because too few can lead to high training error but too many can also lead to high generalization errors [Lee *et al.*, 2018].

Training

The process of training a Neural Network has 2 phases:

- The feedforward phase, where neurons using their weights and bias and an Activation Function compute and transmit the values to the layers ahead until the output layer computes some final value.
- Then there is the backpropagation phase where the neurons will update their values of weight and bias based on the error of the output in relation to the true value, this error is calculated by a function called the loss function, these two phases repeat until the error is low enough, each repetition is called an epoch.

Popular loss functions used in Neural Networks are mean squared error, mean absolute error or mean squared logarithmic error, the best option may vary between each problem to solve.

Activation Functions and Optimizers

Activation Functions are used in feedforward to, giving the weights and bias of each neuron, compute the respective output signals [Lee *et al.*, 2018]. Across many possible Activation Functions, ReLu has proven to be the most widely used with most success [Nwankpa *et al.*, 2018]. In [Lee *et al.*, 2018] ReLu was shown outperforming Sigmoid, Tanh and Softplus activation functions, achieving a lower mean square loss in a 1000 epoch training.

Other parameters we can change are loss functions to evaluate our network and optimizers for Back-propagation when training our Neural Network. In [Lee *et al.*, 2018] comparisons between optimizers SGD, AdaGrad, Adadelta, RMSProp and Adam were made, theoretically Adam should be the best overall choice and it obtained the best performance in training, however, conclusions were that the most efficient method depends on problems and conditions.

Convolutional Neural Networks

Neural Networks are also very popular in image recognition problems, with CNN (Convolutional Neural Networks) prevailing in this topic. CNN were inspired by biological processes, their connectivity patterns between neurons resembles the organization of the animal visual cortex and making use of its pooling process and sparsely connected neurons, CNN are remarkably used for image processing in Computer Science. Among all NN architectures CNN have established as the most popular in civil engineering over the last years, since CNN are capable of capturing the 2D topologies, because of a pooling process and sparsely connected neurons[Salehi and Burgueno, 2018]. Recent studies have also showed that CNNs can perform better in both speed and accuracy compared to conventional Artificial Intelligence methods and can extract and learn optimal features from raw data [Salehi and Burgueno, 2018].

2.2 Artificial Intelligence in Civil Engineering

The analysis and design of structures rely on the numerical models, which are derived in order to accurately represent the real behaviour of a structure. Essentially, for the analysis of stresses and deformations of a structure, models based on the finite element method are widely and successfully adopted in structural engineering. However, in an attempt to improve some computational time efficiency, recently there has been a growing interest in applying Artificial Intelligence in Civil Engineering projects [Salehi and Burgueno, 2018]. Mainly with the use of machine learning methods, including deep neural networks, for complementing the model-based approaches, e.g. finite-element simulations, that are the state-of-the-art in civil engineering.

Today one of the biggest areas of application of AI in Civil Engineering is in aiding SHM (Structure Health Monitoring) and for the past two decades, significant progress in developing SHM models for different kinds of structures has been made [Salehi and Burgueno, 2018].The capabilities and the recent developments of artificial intelligence algorithms has made possible to analyse significant amounts of data measured "in-situ" from the real behaviour of structures. Depending on the structure relevance, it is common to have the structures monitored (e.g., reading accelerations, deformations) and thus access to a valuable data representing the real behaviour of structures. The use of such data allow to have a better insight on the structural behaviour and to tune the corresponding modelling. AI is an excellent auxiliary tool providing an efficient and faster way to deal with the structural health monitoring of structures, identifying patterns, contributing to select the most appropriate parameters to read and the corresponding location, making possible then, to act with regard to the safety of the structure as fast as possible. On the other hand, by being capable of analysing the significant amount of data from the structure monitoring, it allows to build a reliable digital twin of the structure.

Neural Networks at Predicting Structure Behaviour

Another line of research that has been much less explored is to use AI to improve the calculation time of design parameters. Neural Networks can be a good tool for this computations [Chaudhary *et al.*, 2007] [Kaczmarek and Szymańska, 2016]. Today, FEM are commonly used to calculate structure behaviours, however when dealing with nonlinear examples, FEM needs to solve non-linear PDE (Partial Differential Equations) and that can lead to solving systems of non-linear equations, which is highly computationally expensive [Al-Aradi *et al.*, 2018]. Since FEM approach is to discretize the whole computational domain in small regions, compute some form of solution and then gather all regions to put the whole global solution together, non-linearity can be too expensive and hard to calculate and whenever we decide to change something in the initial conditions and design parameters all the calculations must restart from the beginning[Pedro *et al.*, 2019]. Using AI, it could be possible to calculate directly the structure behaviour values or integrate AI models in the FEM to speed up the more computationally expensive steps, either way the goal would be to make calculations quicker and save both human and machine time and effort when designing a new structure, making also possible to try more alternative designs to find the best possible one.

For the design of civil engineering structures, the corresponding structural behaviour of its structural elements (beams, slabs columns) and the structural system as whole has to be evaluated. This corresponds to determine the distribution of integral forces and stresses, displacements, deformations and accelerations. All this data will allow to evaluate the performance of the structure under service conditions and to evaluate the safety to ultimate limit states (i.,e collapse of the structure) In a non-linear scenario, determining such responses of beams and other structural elements can be a computational challenge, particularly due to geometrical an physical non-linear behaviours. As for example implementing the reinforced concrete elements deflections are hard to compute because of the non-linear stress-strain relationship of and steel [Kaczmarek and Szymańska, 2016]. Neural Networks can be a valuable tool to compute these calculations. Neural Networks capabilities at prediction, approximation, grouping, interpolation have always been an aid at solving engineering problems like non-destructive testing results analysis, construction processes planning or geotechnical problems [Kaczmarek and Szymańska, 2016].

3 Analysis of Beam Structures

In this section we provide a quick overview of beam structures to understand the difficulties and uses.

First we start with simple linear scenarios, focusing on the neural network development. This linear cases could already prove the potential of neural networks at modeling structure responses. Only after proving that initial idea, we escalate into more complex and non linear scenarios. These scenarios could show improvement in computational time and effort of the NN model compared to conventional FEM based procedure.

Inside our beam scenarios, there are 2 main parameters that can vary, as seen in Fig. 2. First, the size of the beams, not only in the actual length in meters of each beam but also if it is a single span beam or a multiple span continuous beam. Second, the type of load on the beam, not only in its magnitude but also in its type, being concentrated in one or more points or evenly distributed across one beam.

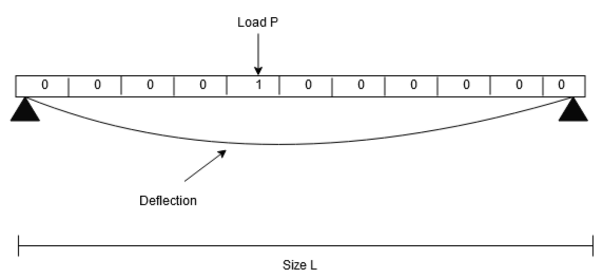

Figure 2: 1-Span Beam Diagram (Concentrated Load)

We then needed some form of representation to translate this information into a form capable of being read by Neural Networks. What we did was to discretize the whole beam model into X positions, where we could have a load or not, and simply place a value of 1 if there is a load, or a value of 0 if there is not any load. Figure 2 show this interpretation as well as the first scenario we chose for the first test, that was a 1-span beam with one concentrated load.

3.1 Structure Behaviours and Responses

We also need to know which behaviours and results we want to predict.

For the design of a beam structure, it is required to evaluate several parameters characterising the structural response due to the applied load. A wrong evaluation of those parameters can lead to designing a structure that is unsafe and that could have catastrophic consequences. As our goal is to create a model that approximate the real values of these parameters in the preliminary phase, it is important that we can accurately evaluate the most important and significant behaviours and structure responses. Factors such as the Bending Moment, that corresponds to an internal force associated with the beam flexure introduced by the loading. Or Rotation and Section Curvature, that show how much the structure displaces from its undeformed shape after subjected to the load. These responses evaluate the safety of a beam design.

Non linearity

An important parameter to know when designing beam structure is the maximum force it can withstand safely without causing damage to its structure, which is commonly known as the beam load capacity. But nonlinear behaviour can be challenging to predict with Neural Networks. To understand why, we take the example of the material we've later chosen for the tests, which is admitted to be elasto-plastic. This nonlinear material on our structure will have 3 different phases during the stress test that we want to predict. The first one is when the beam has an elastic behaviour behaviour, as load increases, some point in the beam will reach its bending yield moment and eventually will become fully yielded, this will form the first plastic hinge and we enter the second phase that is the plastic behaviour, meaning additional load will be redistributed to other parts of the structure as the hinge rotates. Eventually other hinges will be formed as the whole process repeats, until there are enough hinges to make the structure collapse (mechanism), the final third phase is after the structure collapsed, we will still have to predict this phase with NN to find out the beam load capacity.

One way to know the values of force that cause a plastic hinge is calculating results for increasing load in a scenario, and then combining the bending moment with the rotation in a plot. Because after plastic hinges occur, the plastic hinges allow free rotations to distribute the load along the beam, consequently the rotation levels increase even without increasing the load. This means we can identify the force necessary to create the plastic hinges by plotting the bending moment and rotations together and observe when the bending moment stops increasing but the plastic hinge allows free rotations so the rotation levels keeps increasing.

Yield Moment and Maximum Force Supported in ABAQUS

To show how we can plot the Bending Moment with the Rotation values to calculate the force to cause a plastic hinge, we present an example with data we obtained from ABAQUS. So, for the same beam design, we did twenty simulations, each with a different value of load, loads changed from 30kN to 600kN, increasing 30kN in each simulation. Knowing the plastic hinges would form in position 2 and 6, we extracted the values of bending moment and rotation for those positions in each one of the 20 simulations and plotted them together.

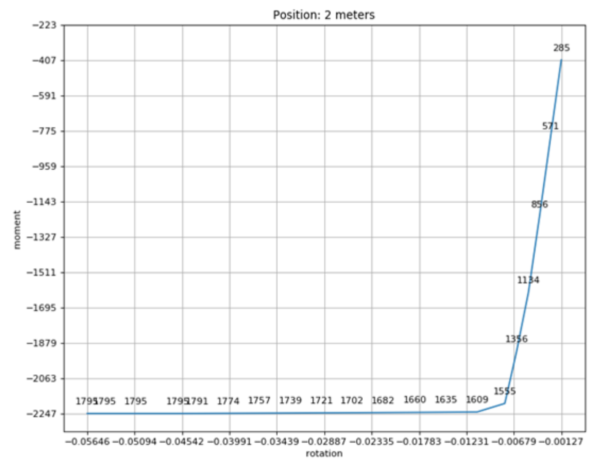

Figure 3: Moment vs Rotation at position 2 meters

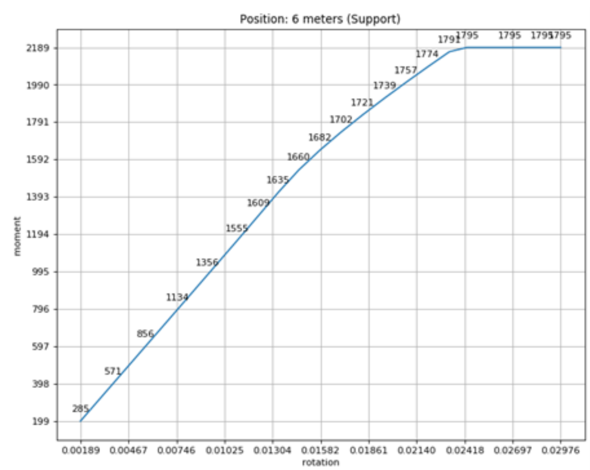

Figure 4: Moment vs Rotation at position 6 meters (support)

Figures 3 and 4 show these graphs, where each point is represented with the value of load that originated it. It is simple to observe that at a certain point the slope of these graphs becomes null which is due to the fact of the bending moment stopped increasing but the rotation kept increasing. We can understand that this means the load has reached the maximum load before a plastic hinge occurs. Since the value of load that originated this point is 1609kN for the position 2 graph but for the position 6 graph it is 1795kN, it means that the first plastic hinges occurs in position 2 when load is 1609kN and we enter a elastoplastic regime, only when the load is 1795kN the second plastic hinge occurs in position 6 and that would cause the structure to collapse. It would be interesting to try to predict this with Neural Networks.

Using Section Curvature to identify plastic hinges positions

To this end, it would be needed to determine the positions where the plastic hinges will occur, only then we can predict the Bending Moment vs Rotation graphs on those positions and find the magnitude of the forces that cause the plastic hinges. One can find this out by looking at the section curvature values after the analysis of a beam. Just like Rotation, Section Curvature also represents how much a node rotates. After plastic hinges occur and as load increases, the plastic hinges will rotate to distribute the load along the beam. Observing the nodes where these values are the highest, we can assume that those are the nodes where the plastic hinges occurred.

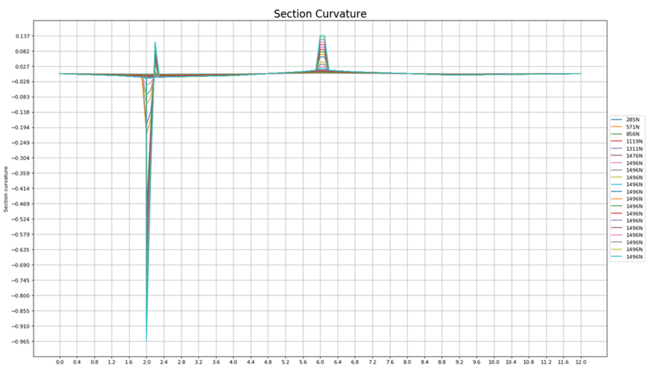

Figure 5: Section Curvature along the beam

Figure 5 is the result of twenty ABAQUS simulations of the same scenario with increasing loads. We can see that as the load input increases, the curvature value spikes in position 2 and 6, this is because in this scenario, the plastic hinges occurred in the positions 2 and 6.

This is the method we will use to know the positions of the plastic hinges on the beam, and after that, plot the bending moment vs rotation in those positions to find out what are the minimum loads to cause the plastic hinges and consequently find the maximum load supported by the beam.

Most conventional methods of calculating structure responses are based on FEM. FEM is numerical procedure to determine approximate solutions of differential equations, FEM approach is to subdivide the domain in smaller parts called the finite elements and calculate solutions for those smaller parts before putting everything together. However, structure characteristics such as the material properties or the shape of the structure can involve non linearity. This can make FEM a procedure much harder to solve that even the most advanced software can take an immense amount of time to complete a simulation.

In this dissertation, we focused on beam scenarios, looking to significantly improve the time and effort of laying out a new beam design, by using a neural network model that can calculate an approximate solution almost instantly.

4 Approximating the Behavior of Beam Structures with Neural Networks

Having defined the beams representations and learning how to obtain the metrics we needed, now we could settle the beam scenarios and simulate all examples of beams to create the datasets. Then, create the neural networks and train them with the datasets, so we can finally use the neural networks to predict new designs and evaluate the results.

We considered 4 scenarios in total, linear single span beams, linear 2-span beams, non-linear 2-span beams and non-linear 3-span beams.

4.1 Linear 1-Span Beams

We used the Linear examples as an initial proof that NNs can predict beam structural responses, and only predicted the Bending Moments and Deflections.

For the 1-span beams we discretized the whole beam in 11 positions equidistant from each other, the position of the load could be on any of the 11 positions of the beam except the extremity, so it could vary through 9 possible positions.

We choose the 1-span beam as our first scenario exactly because it is simple and linear, to create the dataset necessary for these beams we could simply use python to solve the polynomial equations, this helped save a lot of time in the data gathering phase. We calculated 3 different datasets for this 1-span beam scenario, one with a concentrated load deflection, another with concentrated load bending moment and then with a uniform load deflection.

The equations calculated to create the data necessary to train the different neural networks were the following:

Concentrated Load Deflection

$$y(x) = \frac{Pbx}{6EI}(L^2 - x^2 - b^2)$$

Where P is the Force of the Load, E is the Young's modulus and I is the area moment of inertia, which are all constant along the beam and multiplied by the whole formula, so we can ignore them for now because we can simply multiply them after the NN prediction. L is the size of the Beam, b represents the distant between the position of the load and the end of the beam and x represents the position where we are calculating the deflection.

The data-set included the dimension of L that varied from 1 to 101, the position of the load P that could be on any of the 9 possible positions and the resulting Deflection value in each one of the 11 positions. The data-set had 10 000 samples.

Concentrated Load Bending Moment

$$y(x) = \frac{Pbx}{L}$$

This was the formula used to calculate the Beding Moment for the 1-span beam.

Uniform Load Deflection

$$y(x) = \frac{Px}{24EI}(L^3 - 2Lx^2 + x^3)$$

And finally this was the formula used to calculate the Deflection data for the Uniform Load 1-span beam. The variables and dataset size are the same as before. This time, since the position of the load was constant, the dataset only included the dimensions of L, varying between 1 and 101 as well, and the values of Deflection in each position.

4.2 Linear 2-Span Beams

Still in the linear examples we decided to test a 2-span Beam with a concentrated load as seen in Figure 6. For this beam we only created a dataset with the values of deflection. The dataset for this scenario was composed with the size of each span (varying between 1 and 10), the position of the load, the force created by the load (varying between 1 and 100) and the resulting deflection values in each positions of the whole composite beam. This time and henceforth the data was obtained with the use of ABAQUS.

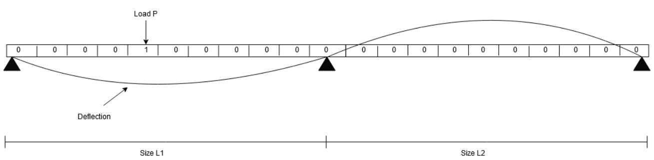

Figure 6: Linear 2-span Beam Diagram

4.3 Non Linear 2-Span Beams

For the non linear scenarios, we changed the material of our beams to an elastoplastic material that introduces non linearity in the behaviour and therefore in the calculations. Here is where we will see the potential of neural networks computational time and efficiency, since the first linear examples are also very quickly computed with traditional methods using FEM, but non linear ones can take a long time to calculate.

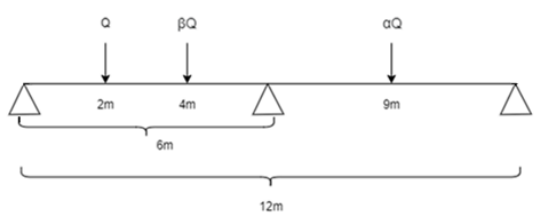

Figure 7: Scenario for the Non-Linear 2-Span Beams

The scenario for our first nonlinear tests is the one represented in Figure 7. We chose a 2-span beam where each span has 6 meters, so 12 meters in total. The material chosen is elastoplastic to introduce nonlinear behaviour and the beam is pinned on the edges. Each span has three concentrated loads, two on the first span and one on the second span. The first concentrated load has a force of Q while the second and third have βQ and αQ, respectively. For simplicity we discretized the beam in thirteen positions, one for each meter of the beam. So, six positions for each span, with position 0 and 12 being the pinned edges and position 6 the middle support.

Various examples were then computed for this scenario where we varied the position and magnitude of the loads. Excluding the support positions, the first two loads could take any position on the first span and the third load could take any position on the second span. The magnitude of the loads varied from 10kN to 3000kN. β and α could take the values 0.5, 1 or 2. We simulated in ABAQUS, 70.000 random examples inside the domain of the scenario. We extracted values of Section Curvature, Bending Moment and Rotation at the end of simulation. In our datasets, we had information about the positions of the loads, their magnitudes and the values of Section Curvature, Bending Moment and Rotation in each one of the thirteen nodes of the beam.

Non Linear 3-Span Beams

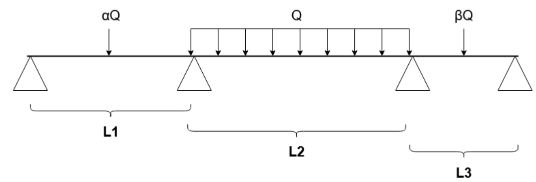

Figure 8: Scenario for the Non-Linear 3-Span Beams

We then escalated into a 3-span scenario, this would be the hardest scenario we would try to predict with NN. Represented in Figure 8, our beams were composed by 3 spans that could vary in size from 1 to 10 meters, the material was elastoplastic and the edge supports were pinned. One of the spans had a uniform load while the other two had a concentrated load in the middle, the span that had the uniform load could be in the middle or on the edges, the uniform load had a magnitude of Q while the the other two had magnitudes of α Q and β Q respectively. To simplify and to able our NN to predict all the possible examples, we discretized the whole beam in 31 positions, 11 for each span, sharing the edges with the next one.

To create the dataset we simulated around 70.000 random examples. In each example we varied which span had the uniform load (middle or edge), the size of each span, (from 1 to 10 meters), the magnitude of the load Q(from 10kN to 3700kN) and β and α that could take the values 0.5, 1 or 2. Structure responses such as Rotations, Section Curvatures and Bending Moments, as well as input information such as the load types, magnitude and positions and the beam sizes were then extracted to build the datasets to train the neural networks.

4.4 Neural Network Models

Having created all the datasets for all the scenarios developed, all that is left is to create the neural networks, train them with the datasets for each scenario and evaluate the predictions.

Simple MLP Network

The first NN created was a simple NN, represented in Figure 8. This NN was only used for the linear examples, it takes as input 12 nodes, 11 representing the 11 positions of the beam with values either 0 or 1 depending on the existence of a load and another node to represent the size of the beam.

This Network has 3 hidden layers with 64, 32 and 24 neurons respectively. As for the output layer, 11 nodes are computed representing the values of deflection or bending moment in each of the 11 positions of the beam.

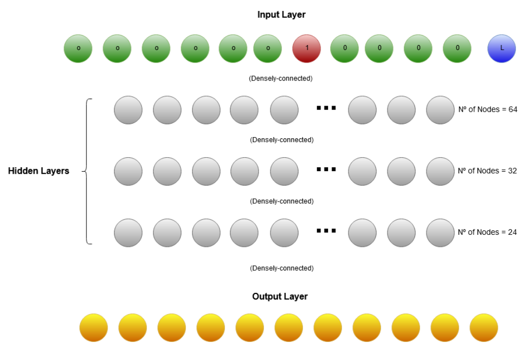

Figure 9: First simple NN tested for Linear examples

Complex MLP/CNN Network

We then decided to test a different NN architecture, because in all our scenarios we had n equally distant points to represent our beam, so a valid argument could be made to use CNN for its uniformly distributed weights and transition invariance on the nodes. This network is a more complex NN (represented in Figure 9, for the 2-span non linear scenario), it only takes as the input layer the nodes that represent the positions of the beam (this number varied depending on the scenario), then goes through Hidden Convolution Layers before we concatenate the rest of the information like the size of the beams or magnitudes of loads, depending on the scenario. The next layers are normal densely connected, before giving the output with the value of of the structure response in each one of the positions. All these layers had ReLu as the activation function, that was proven to be the one with the fastest convergence. Also, the loss function with best convergence was Mean Squared Error, and the best optimizer was Adam because of the adaptive learning rate.

This architecture was used for all the linear and nonlinear scenarios and it was the one that obtained the best results.

Single Output MLP Network

Nevertheless, we still tried another architecture. represented in Figure 10, the different approach here is that we would only give one input of position and obtain the output result for that given position, instead of predicting the whole beam like before. The hidden layers on this network were five and all normal densely connected they all also had ReLu as activation function.

Prediction Methodology

To predict the plastic hinges locations and the maximum load supported, we trained 2 Neural Networks with the same architecture, one with the Section Curvature dataset and the other with the Bending Moment dataset. When receiving a new beam design, our model uses the Section Curvature NN model to predict the positions where the plastic hinges will occur, to find it it simply simulates with a very high value of load in the input and the model predicts where the structure would break, therefore that is where the plastic hinges occurred. After knowing the plastic hinges positions, it will use the Bending Moment model to predict the yield moment and the forces necessary to make the structure collapse on the plastic hinges. For that, it must predict the bending moment vs rotation graphs, so it calls the NN model multiple times to predict the same beam scenario but with incremented loads and plot the bending moment until the slope of the graph turns null, the load that originated that point is the maximum value of load.

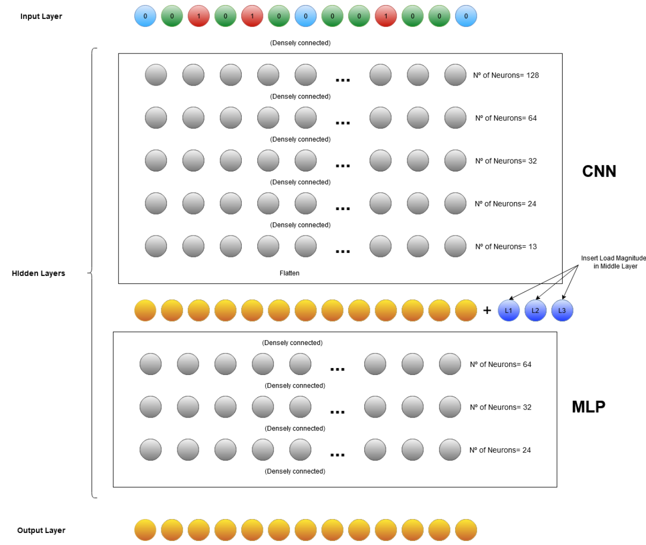

Figure 10: Complex MLP/CNN Network developed

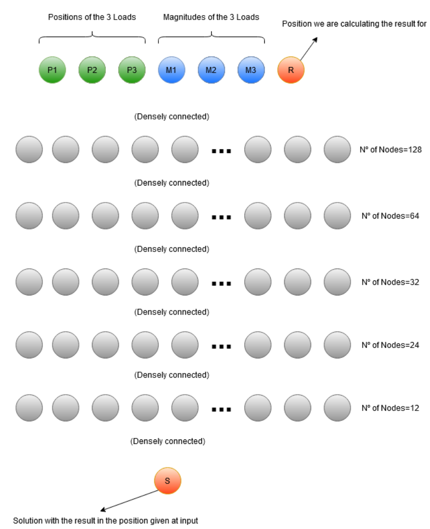

Figure 11: Single Output Neural Network Architecture

5 Results and Evaluation

We compared the network architectures developed in the several scenarios considered in this work, we then show some predictions on the most relevant structure responses and evaluate our method of finding the maximum load supported by the beam.

5.1 Linear Single Span Beam

In this scenario, results were successful with the one concentrated load and with the uniform load. The complex NN with the convolutional layers (Figure 10) performed better, achieving the lowest error rate faster. In figure 12 we can see an example of a Deflection prediction in this scenario that is quite accurate.

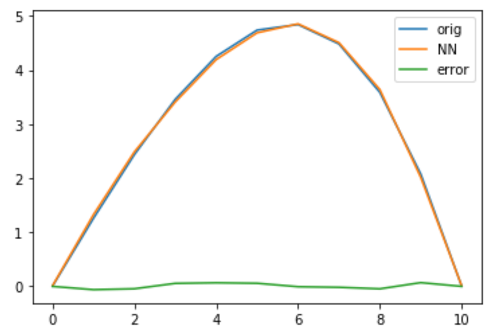

Figure 12: Predicting Deflection of Concentrated Load on 1-Span Beam

5.2 Linear 2-Span Beams

This Scenario was the most complex of the linear ones. Again, the complex NN with the convolutional layers (Fig.10) performed better, achieving the minimum error rate in less iterations. The reason behind this increased performance is probably exactly those convolution layers that converge better than normal dense layers when we give the nodes in a form representing the structure of our beam and therefore, the training is improved significantly.

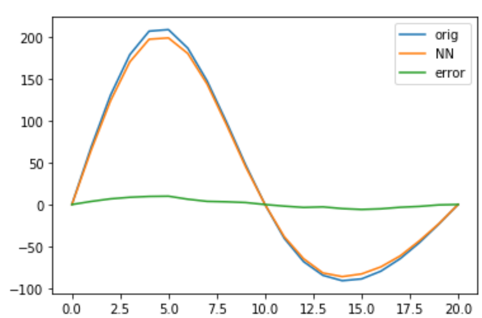

Figure 13: Predicting Deflection of Concentrated Load on 2-Span Beam

A prediction with this NN can be observed in figure 13. Even though all these tests demonstrated were simple and solving these linear scenarios with traditional methods also takes a very short amount of time,it still proves that Neural Networks are effective and capable of predicting very accurately structure responses of beams.

5.3 Nonlinear 2-Span Beams

In this scenario we tested the complex CNN/MLP Network (figure 10) and the single output MLP Network (figure 11) and compared the results. We can observe comparisons of predictions in figures 14 and 15. We can observe that the CNN/MLP architecture still performed slightly better in these non linear beam designs.

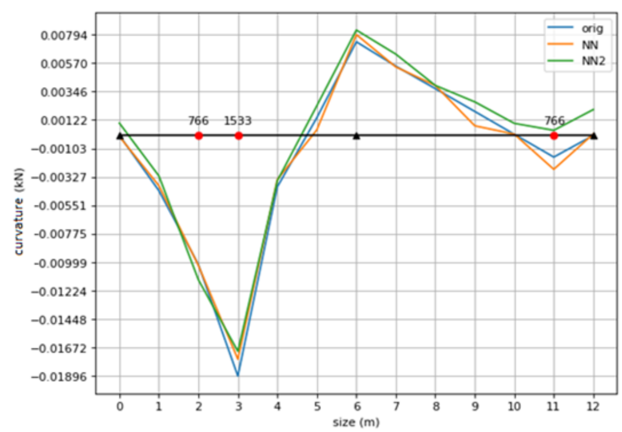

Figure 14: Comparison between the two NN architectures in predicting a Section Curvature example

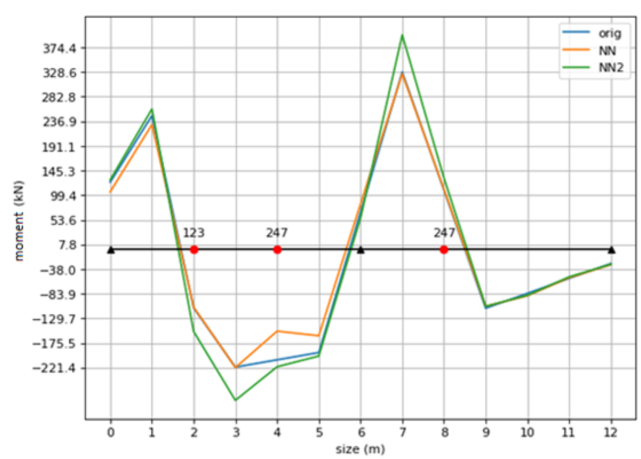

Figure 15: Comparison between the two NN architectures in predicting a Bending Moment example

5.4 Nonlinear 3-Span Beams

Finally in the last scenario with the 3-span beams, we only tested the CNN/MLP Network, that obtained some convincing results even in this complex scenario with many variables to generalize. Examples of structure responses predictions in this scenario are seen in figures 16, 17 and 18.

Being able to predict section curvatures, bending moments and rotations in the non linear scenarios meant we could then try to predict the plastic hinges positions and the loads necessary to form them.

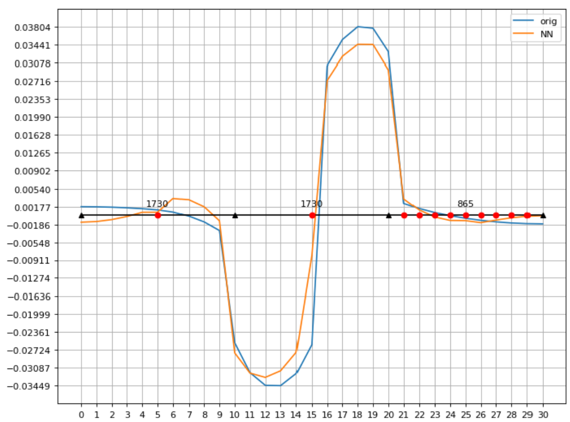

Figure 16: NN prediction of Rotation of 3-Span Beam, Sizes: 2m, 6m, 1m

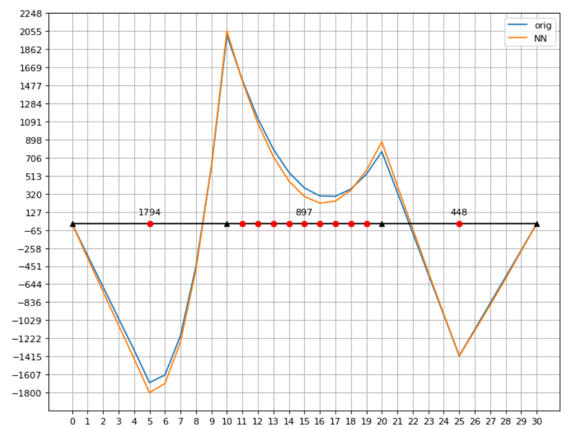

Figure 17: NN prediction of Bending Moment of 3-Span Beam, Sizes: 6m, 3m, 4m

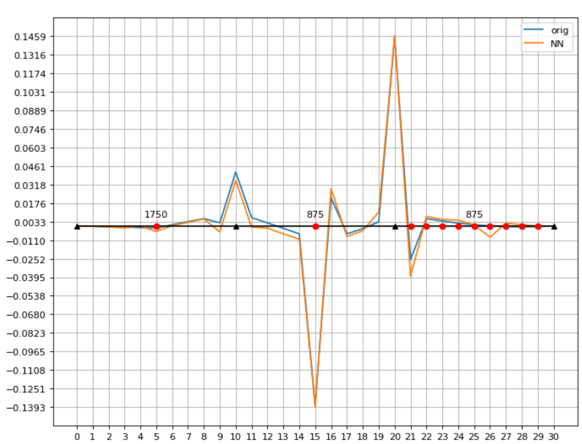

Figure 18: NN prediction of Section Curvature of 3-Span Beam, Sizes: 7m, 6m, 3m

5.5 Maximum Load Supported Predictions

To exemplify our method of finding out the maximum load supported by a beam we take an example of the 2-span beam scenario with $\alpha = \beta = 1$. It starts by plotting the section curvature graph when giving a value of load high enough to observe the plastic hinges points, as we can see in Figure 19 it would be positions 2 and 6 since those are the points where the section curvature value spikes.

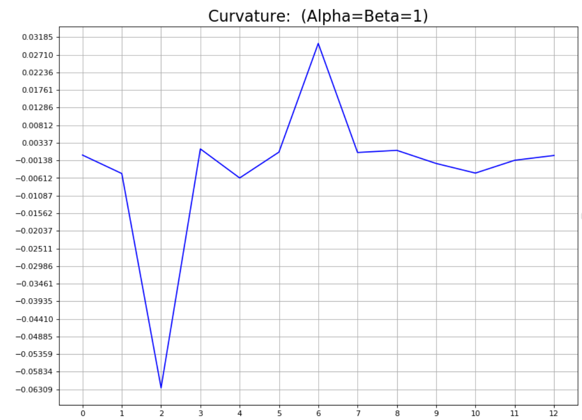

Figure 19: NN prediction of Section Curvature 2-Span Beam with $\alpha = \beta = 1$

Then we calculate the rotation and bending moment graphs for the same scenario, multiple times with incremented loads. We simulated the same beam with values of load from 25kN to 1975kN, increasing 50kN in each subsequent simulation, we did not go denser here, say 1kN in each simulation, for the sake of the clarity of the plot to be showed in this document. Then we can finally take the values of bending moment and rotation on the plastic hinges identifies before and plot them together to create the graph in blue in figure. We identify the maximum possible load one the point where the absolute value of bending moment lowers and stop the simulation there. The graph in orange in the same figure is the original beam computed by ABAQUS and we can observe our prediction is close. The real value was 1564kN and our model predicted 1625kN. Keeping in mind that in our final model our incrementations are denser, we increment load by just 1kN in each step, instead of 25kN like in this example we can get even more accurate predictions.

Our program simulates around three thousand different incremented loads for a given scenario, then computes and outputs the result in around four and a half seconds. The following table shows some scenarios executions times comparisons between ABAQUS FEM based simulation and our NN model predictions.

6 Conclusions

Planning and building a civil engineering structure is a serious and complex project, designing a safe architecture for the structure is of course an essential part to avoid risks. Nowadays, conventional methods to verify the safety of a complex and non linear structure can be very time expensive. Neural

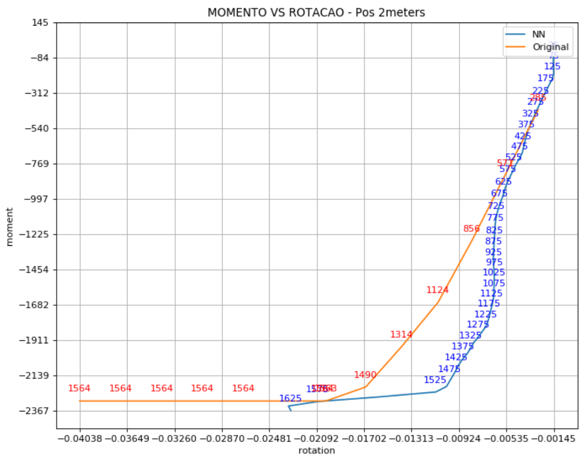

Figure 20: Comparison between real and Final Model prediction of Moment vs Rotation of 2-Span Beam with $\alpha = \beta = 1$

Table 1: Beam Scenarios execution times

Scenario	Simulations	ABAQUS	NN
Linear 2-Span Beam	1	27s	0.001s
Linear 2-Span Beam	20	541s	0.002s
Non Linear 2-Span Beam	1	27s	0.001s
Non Linear 2-Span Beam	20	541s	0.005s
Non Linear 3-Span Beam	1	28s	0.001s
Non Linear 3-Span Beam	20	565s	0.005s

Network approaches can help to reduce this computational time.

With this work we proved the potential of neural network models by predicting approximations of solutions for beam structures designs in 30 000 less time than with a traditional method.

6.1 Future Work

One thing to improve in our models is the capacity to generalize more different and diverse beam designs. Obtaining new training data to incorporate different designs, and then test the prediction results on them, would be the first step to improvement. Likely, would also be necessary to modify the NN architecture, since now the beam architecture has to be represented in a less constant way.

Another interesting goal for the project initiated here would be to incorporate it inside the FEM model. Our model can offer a fast approximation of structure responses for an easier and faster modeling at a preliminary design phase. However, it could also be possible to incorporate our model with the FEM to replace the more complex and computational expensive functions with the NN.

Not so much an improvement to our model, but more of an alternative with higher potential to solve the same problem. It could be possible to use the neural network models to learn and predict the partial differential equations solutions that have to be solved for FEM in non linear models that are computational expensive and significantly increase the time efficiency.

References

[Al-Aradi *et al.*, 2018] Ali Al-Aradi, Adolfo Correia, Danilo Naiff, Gabriel Jardim, and Yuri Saporito. Solving nonlinear and high-dimensional partial differential equations via deep learning. *arXiv preprint arXiv:1811.08782*, 2018.

[Chaudhary *et al.*, 2007] Sandeep Chaudhary, Umesh Pendharkar, and Ashok Kumar Nagpal. Bending moment prediction for continuous composite beams by neural networks. *Advances in Structural Engineering*, 10(4):439–454, 2007.

[Fausett, 2008] Laurene Fausett. Fundamentals of neural networks architectures, algorithms and applications. 2008.

[Kaczmarek and Szymańska, 2016] Mateusz Kaczmarek and Agnieszka Szymańska. Application of artificial neural networks to predict the deflections of reinforced concrete beams. *Studia Geotechnica et Mechanica*, 38(2):37–46, 2016.

[Lee *et al.*, 2018] Seunghye Lee, Jingwan Ha, Mehriniso Zokhirova, Hyeonjoon Moon, and Jaehong Lee. Background information of deep learning for structural engineering. *Archives of Computational Methods in Engineering*, 25(1):121–129, 2018.

[Lopez *et al.*, 2008] R Lopez, Eva Balsa-Canto, and Eugenio Oñate. Neural networks for variational problems in engineering. *International Journal for Numerical Methods in Engineering*, 75(11):1341–1360, 2008.

[Nwankpa *et al.*, 2018] Chigozie E Nwankpa, Winifred Ijomah, Anthony Gachagan, and Stephen Marshall. Activation functions: Comparison of trends in practice and research for deep learning. 2018.

[Pedro *et al.*, 2019] Juan B Pedro, Juan Maroñas, and Roberto Paredes. Solving partial differential equations with neural networks. 2019.

[Salehi and Burgueno, 2018] Hadi Salehi and Rigoberto Burgueno. Emerging artificial intelligence methods in structural engineering. *Engineering structures*, 171:170–189, 2018.