Sayyed Mohsen Vazirizade

23398312

smvazirizade@email.arizona.edu

INFO 521

Introduction to Machine Learning

Project

Using Artificial Neural Network for Predicting Response of the Structure

Introduction describing the method

Artificial Neural Network (ANN) in terms of machine learning is inspired by a rudimentary and simplistic model of the biological neural network[1]. The following figure shows and overview of a neuron in human brain[2].

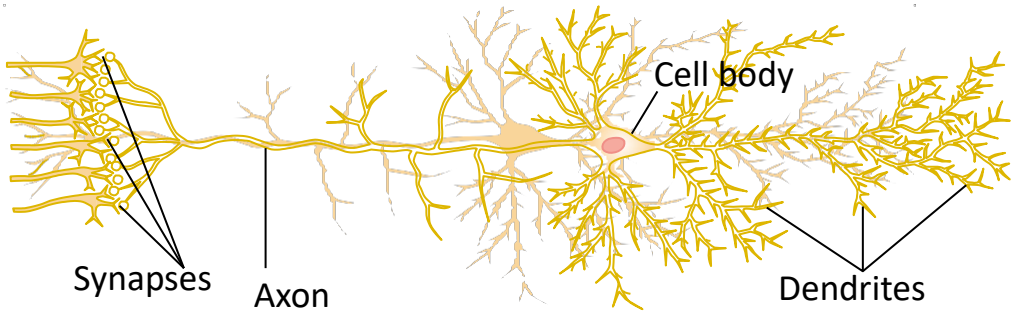

Figure 1 An overview of a neuron in human brain

On the other side, the movement of each floor relative to the upper and lower floors is of importance in a building structure. There are many mathematical based relationships for simulating the response of the building to a white noise excitation or earthquake record. However, rarely do we know the exact mass matrix, damping ratios, and stiffness matrix of the structure while these mathematical methods highly depend on these values. In this regard, ANN is used to predict the behavior of the structure based on the previous behavior of the structure. A sufficient number of elements (neural) in A neural network can model dynamic systems and addressing non-linear dynamic problems with decent accuracy. In other words, it is very common to record the movement of a structure by using sensors; therefore, the acceleration, velocity, or displacement history of the structure is accessible. Nevertheless, the engineering properties of the structure frame are unknow. Using time history response and employing an ANN, the response of the structure to earthquake or white noise is imitated. There is some other research related to this topic [3]–[7] to name but a few; nonetheless, in this study we aim to minimize the number of the inputs to our model and compare the performance of the ANN.

Procedure

ANN is using simple mathematical relationships to find the connection between input and output. The following figure shows a very common ANN. Each of these circles is a Neuron, having an input and output values as well as a transfer function—usual the transfer function is the same for all neurons, and a collection of neurons is called a Layer–usually, a layer's neurons are not connected to each other.

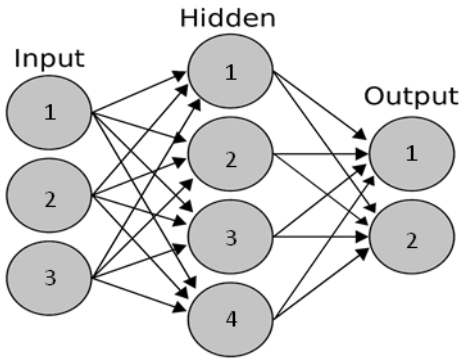

Figure 2 An overview of a ANN.

The mathematical statement of a ANN for *i*th neuron in the *j*th layer is as follow:

$$a^{j}{}_{i} = f\,(x) = f\left(\sum_{k=1}^{m} w^{j}{}_{k,i}\, a^{j-1}{}_{k} + b^{j}{}_{i}\right) \qquad (1)$$

Where

a^{j}_{i}: the output value of that neuron, which would be input values of the *j*+1th layer.

a^{j-1}_{k}: the output value of *k*th neuron in the *j*-1th layer, which is the input value of *j*th layer

m: the number of data as input for this neuron, which usually equals the number of neurons in the *j*-1th layer,

i: the number of the *j*th layer's neurons

$w^{j}_{k,i}$: the weight parameter for the connection of the neuron *i* in the *j*th layer to the neuron *k* in the *j*-1th layer

b^{j}_{i}: the bias value of *i*th neuron in the *j*th layer, and

f: the transfer function

In this study, the output of our ANN is the response of the structure at the current time in a certain floor, and the input is the current and past response of the structure in other floors and history of the response of the structure in that certain floor. Our network is trained during the time and the validation process is a live process. By doing so, we can imitate the response of the structure. The following figure elaborates the mentioned process. The green rectangles are the input data and the transparent one is the output data. The signals in this study are supposed to be acceleration in y-axis and time in x-axis, and the aforementioned rectangles are moving as a window in time.

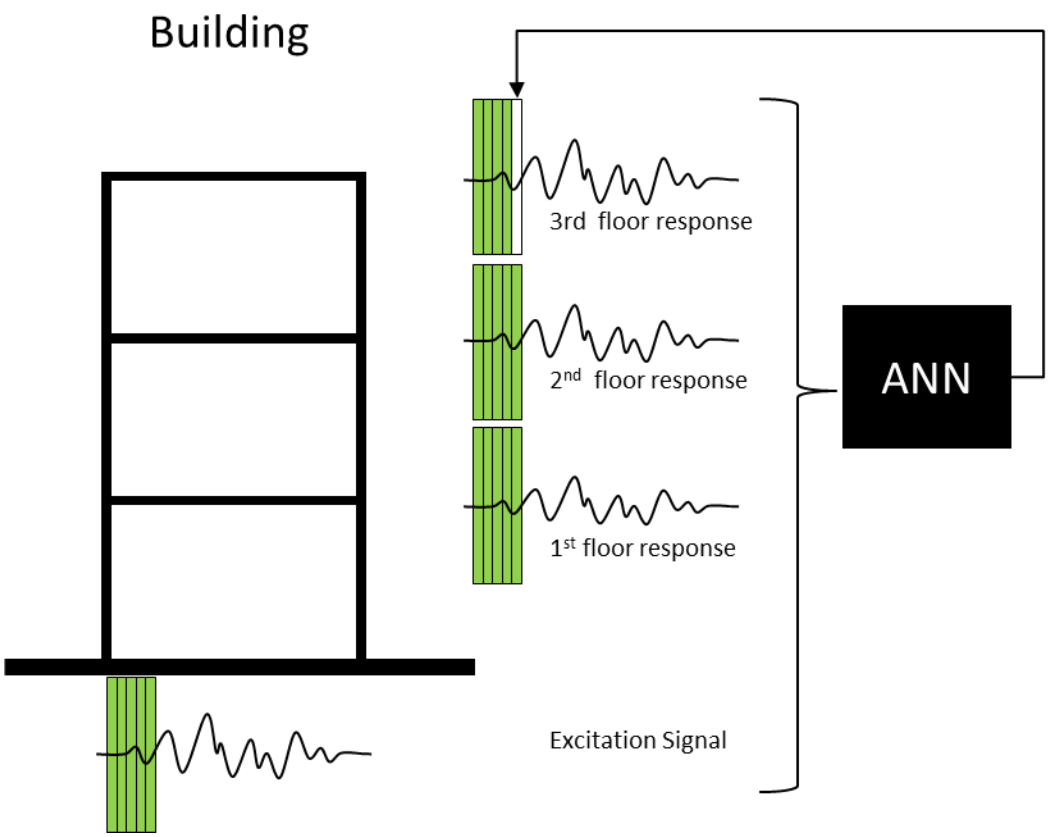

Figure 3 Schematic view of the procedure

Matlab is used for generating figure and the ANNs. Furthermore, OpenSees is used to model our finite element frame structure and generate the structural response to the intended excitation. The format of the files for OpenSees are Tcl. The required script platform has been written previously, and Matlab call it and feed it with new values and collect the outputs and drawing all the figures. All the scripts are attached. On the following, there are three examples. The first example aims to show how an ANN works. The second example is a single-story frame building to show to what extent an ANN can imitate the real response of the structure and which type of time-series of ANNs is more suitable for this goal. The third one, which is the most important one as well, asserts that ANNs in this manner can show the exact time of sudden changes in the stiffness of the structure.

Example 1

This example aims to show this method is applicable to a simple sequential data. We use a single layer network which embrace just a neuron, the simplest network possible. The architecture of this ANN is shown in the figure below.

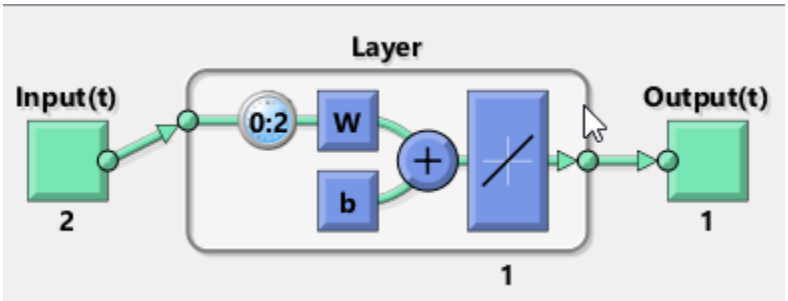

Figure 4 Architecture of a single neuron in single layer network

The goal of this ANN is to predict feature values based on the previous data. First, 100 random data are generated in a sequence. We generate x and y will be calculated automatically.

$$y_i = y_{i-1} + 2x_{i-1} + 4x_{i-2}$$

This ANN uses past data to predict future data; the following figure shows the architecture of data. The ANN uses current x, and past x in t-1 and t-2 step, as well as pas Y from t-1 through t-3. To recap, the number of inputs are 6 and just one output.

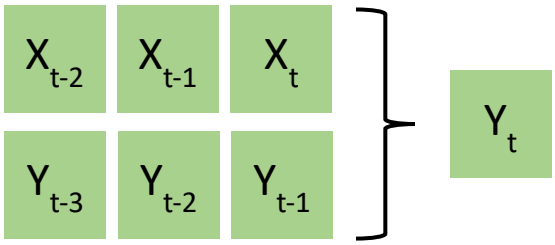

Figure 5Architecture of input and output data

For the aforementioned problem, we used half of the generated data (red line) for trading and the other half for test (orange line) Figure 6. And, the figure below displays the error in percentage. It shows our network is working precisely.

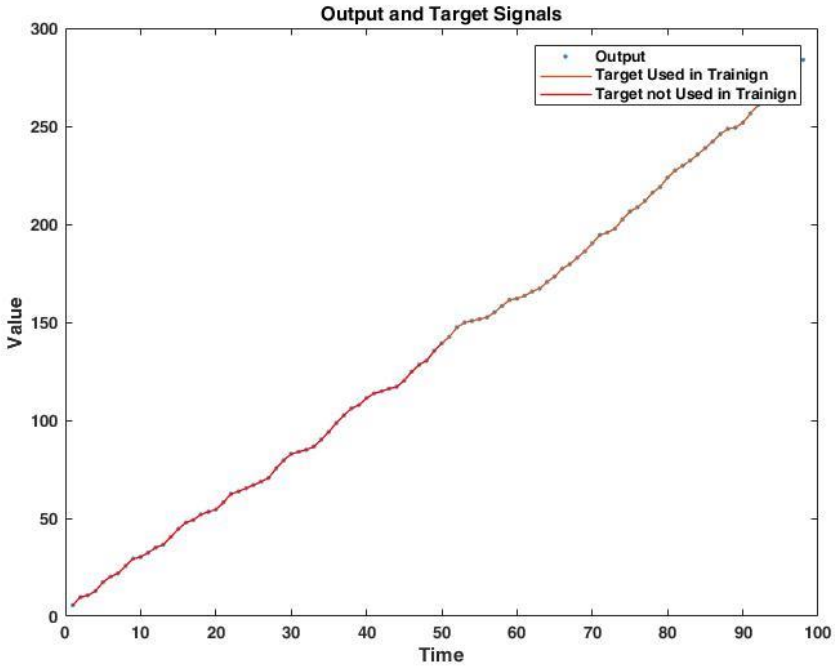

Figure 6 Comparing predicted values and actual ones

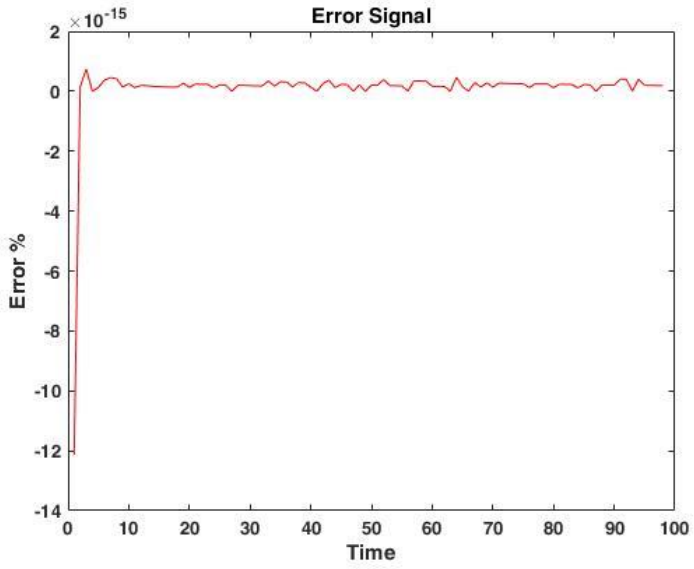

Figure 7 The difference between predicted and actual value respect to actual value

As earlier explained, for this simple ANN we just need 6 weights and 1 bias. It is worth mentioning that the transfer function is considered linear. The figure below shows the calculated values in our network, as we expected.

$W_1=2$ $X_t \times w_1$
$w_2=4$ $X_{t-1} \times w_2$
$w_3=0$ $Y_{t-2} \times w_3$
$w_4=1$ $Y_{t-1} \times w_4$
$W_5=0$ $Y_{t-2} \times w_5$
$W_6=0$ $Y_{t-3} \times w_6$

$+b=Y_t$
$b=4.1017e-15$

Figure 8 Results summary

Setting aside the following part, The script is available in folder example1.

```
clc;clear
%%
%Genreating Data
ExternalInput=rand(1,100);
Feedback(1:2)=0;
for i=3:length(ExternalInput)
Feedback(i)=2*ExternalInput(i)+4*ExternalInput(i-1)+1*Feedback(i-1);
end
%%
END=round(length(ExternalInput)*0.5)                    %value for keeping
some data for test
```

```
Delay=0 ;                                                   %The gap between
input and output, default Delay=0, it means X9n) is effective in y(n)
Span=2;                                                     % How much data you
wanna use as history,
                                                            % For example 1
means, just current data(0) and one data before
                                                            % If yoiu want use
Feedback as input, it automatically will be shifted one step, cause you can
use the same Feedback step as input and output

%%
%This is function for making the data as the way we want
[X,T,Xi] = DATANN2(ExternalInput,Feedback,Span, Delay);
%%
% Design one linear layer with one neuron so we have b1 and w1 to wn which
% n is the number of inputs
net = newlind(X(1:END),T(1:END),Xi);
view(net)
Y = net(X,Xi);
sprintf('The following shows weights')
sprintf('x0 y0 x(-1) y(-1) ...')
net.iw{1,1}
sprintf('The following shows biases')
net.b{1,1}
%%
%%figures
figure
plot(cell2mat(Y),'.');hold on
plot(cell2mat(T),'LineWidth',1)
plot(cell2mat(T(1:END)),'r','LineWidth',1);hold on
xlabel('Time');
ylabel('Value');
title('Output and Target Signals');
legend('Output','Target Used in Trainign','Target not Used in Trainign');
figure
E = (cell2mat(T)-cell2mat(Y))./cell2mat(T);
plot(E,'r')
hold off
xlabel('Time');
ylabel('Error %');
title('Error Signal');
%%
function [X,T,Xi] = DATANN2(ExternalInput,Feedback,Span, Delay)
%Delay=0  ;                                                   %The gap between
input and output, default Delay=0, it means X9n) is effective in y(n)
%Span=1;                                                     % How much data you
wanna use as history,
                                                            % For example 1
means, just current data(0) and one data before
                                                            % If yoiu want use
Feedback as input, it automatically will be shifted one step, cause you can
use the same Feedback step as input and output
ExternalInput = con2seq([ExternalInput',[0 Feedback(1:end-1)]']');
Feedback=con2seq(Feedback);
%if Span(1)==0 && length(Span)==1;
for i=1:Span
    Xi(i) = ExternalInput(i);
```

```
end
X = ExternalInput(Span+1:(end-Delay));
T = Feedback(Span+1+Delay:end);
XXX=(cell2mat(X))';
YYY=cell2mat(T)';
sprintf('The following shows data during the time, vertical axes shows
passing time, first column is x, second column is y(-1) and nex column is y')
[XXX,YYY]
end
```

Example 2

In this example, 3 different ANNs are applied to the response of the single-story shear building. The structure excited by earthquake (acc0) and we record the response of the floor as acceleration (acc1). Figure 9 display schematic view of the model

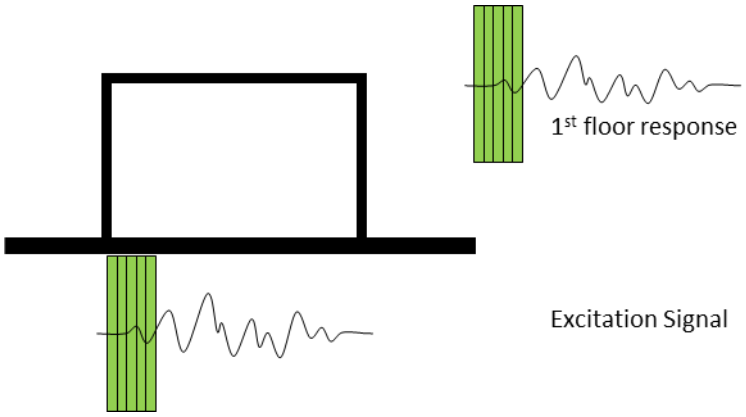

Figure 9 Schematic view of the model

Tabas earthquake record, RSN143_TABAS_TAB-L1.AT2, is extracted from Pacific Earthquake Engineering Research Center (PEER) ground motion database and used for the excitation. The height of this frame is 3m and the span is 4m. The frame considered as a shear building and is modeled in OpenSees. The material is defined as steel01 with yielding stress of 2.4e8 N/m^2 and 2.e11 N/m^2 as the modulus of elasticity. The first period of the structure is also 0.258 s. In order to be in linear behavior region of the elements, the excitation record is scaled with 0.25. Figure 10 indicates the ground motion in the first row, and displacement response, acceleration response, and changes in stiffness in row 2 trough 4, respectively.

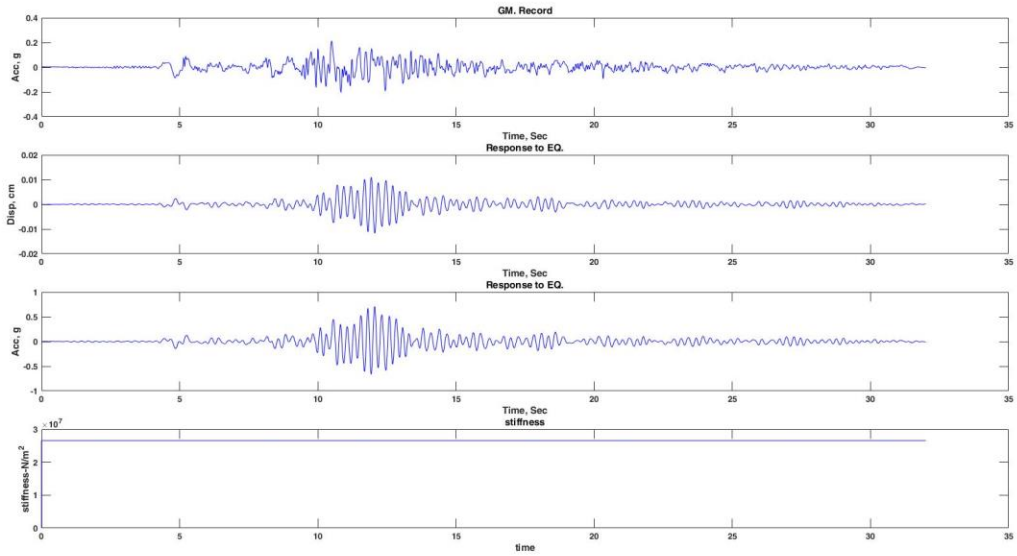

Figure 10 Ground excitation, displacement response, acceleration response, and changes in stiffness

Three different networks are used, Timedelaynet, narnet, narxnet. They are all from the same family with small differences in architecture and type of input values. The following figure shows the architecture of each of them from left to right, respectively. Furthermore, the second row shows the real response (blue line) and the predicted response by our ANN (red line) and the third row shows the error. As it can bee seen there is no significant error in narnet and narxnet models; they predict the behavior of the structure precisely. In all of these three ANNs, a hidden layer with 10 neurons and an output single-neuron layer are employed. The training algorithm is Levenberg-Marquardt (trainlm). For the hidden layer the transfer function is tangant-sigmoid (tansig), and the delay gap is 5 steps. All these assumptions are summarized in the figure below which shows the architecture of each ANN. One more thing that deserves to be mentioned is that in this example we did not use all the data for training. Indeed, the vertical line shows where the training process finishes and after that biases and weights of our ANNs don't change.

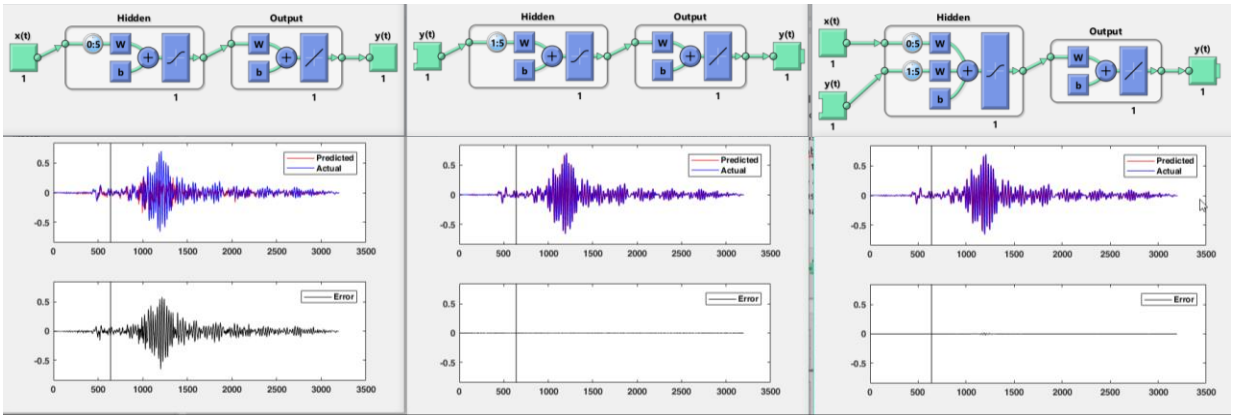

Figure 11 Architecture, predicted response and error for Timedelaynet, narnet, narxnet

All the scripts are used in this example are in a file under the name of example 2. Because the scripts including modeling FEM they are not copied here, and all the comments are on the own file.

Example 3

Any changes in structural stiffness of the structure beget changes in structural response. In terms of safety of the structure, assessing changes in structural stiffness of the structure is of importance. The main goal of this project is assessing whether any changes occur in structure. According to the result of the previous example we use a narxnet ANN. The structure is a 2-story frame, and the response of each floor is recorded. The following figure indicates changes in stiffness of each floor; there is a sudden change in the column of the first floor at second 14.

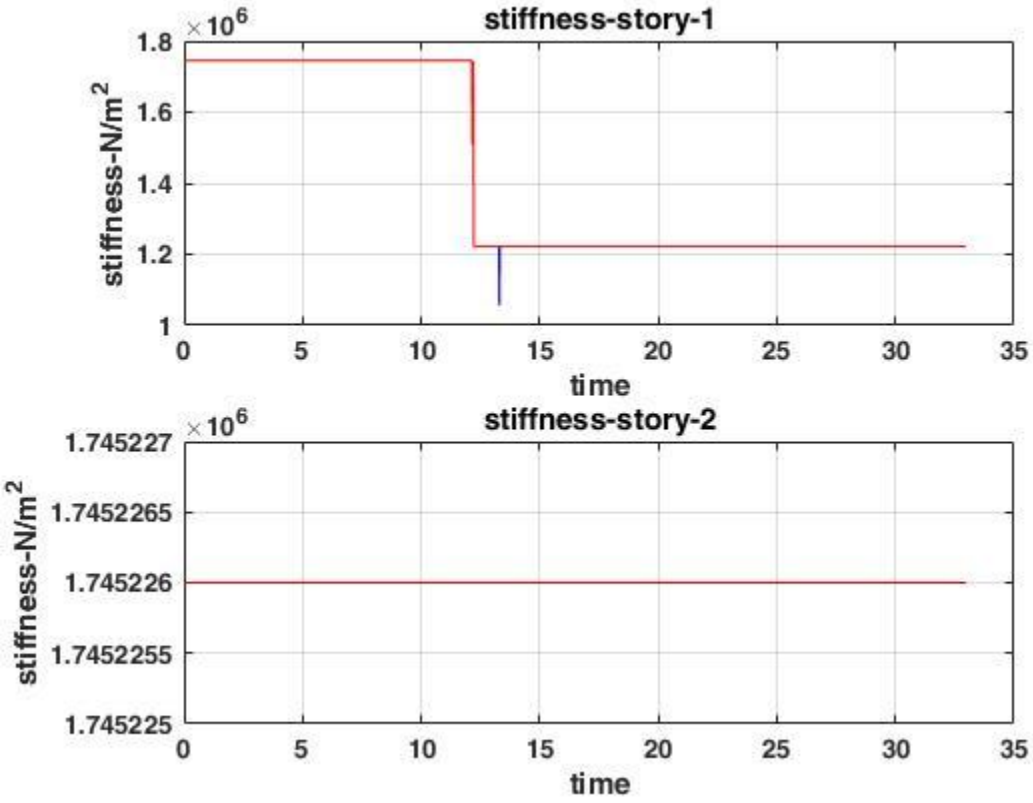

Figure 12Stifness of each floor during the time

Figure 13 display relative displacement of each floor to the ground level. As it can be seen, due to the nonlinear behavior of the structure we have some residual displacement.

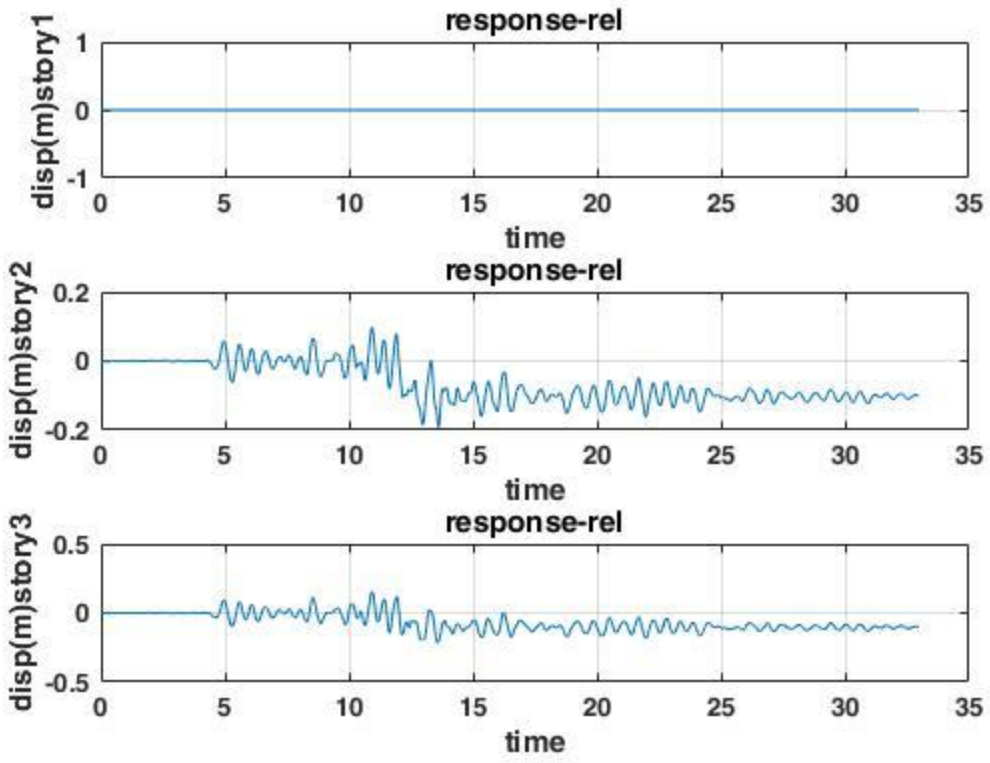

Figure 13 Displacement of each floor

The following figure shed light on the relationship between the scripts in file example 3; the Matlab scripts are in blue while the OpenSees script is in green. First, we run Matlab code Model6.m which generates the input file required by OpenSees script Model1-2.tcl. Afterwards, it calls Model1-2.tcl to run the model using input data stored in folder data and data1. Then, using the output of Model1-2.tcl it prepares the for our ANN model. As the next step we have to run the NN_Openseesdata script. It calls the data from previous step and train the ANN and draw the remaining figure.

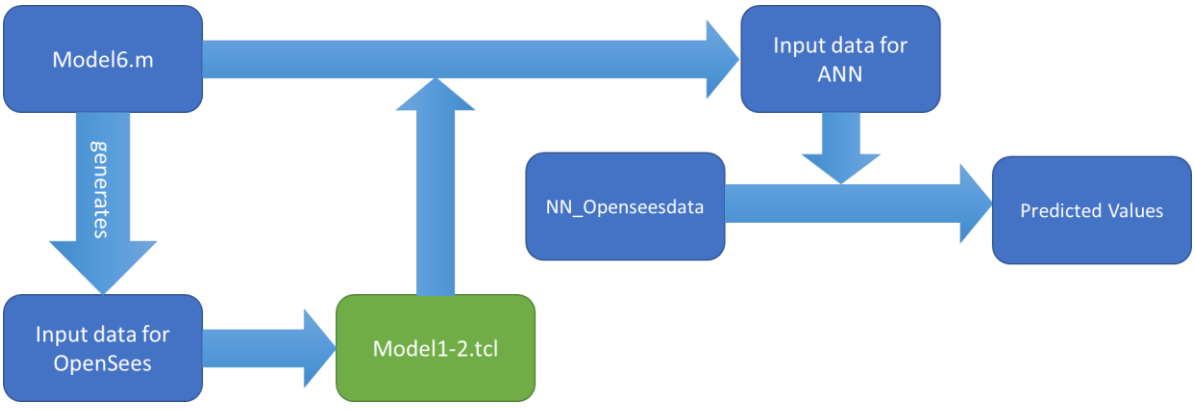

Figure 14 Schematic view of the model

Figure 15 shows the architecture of our ANN. It uses the acceleration data for story0 (ground motion) and story2 (the last floor) as well as the previous steps of story1 (the first floor) to predict the response of the structure for the first floor at the current time.

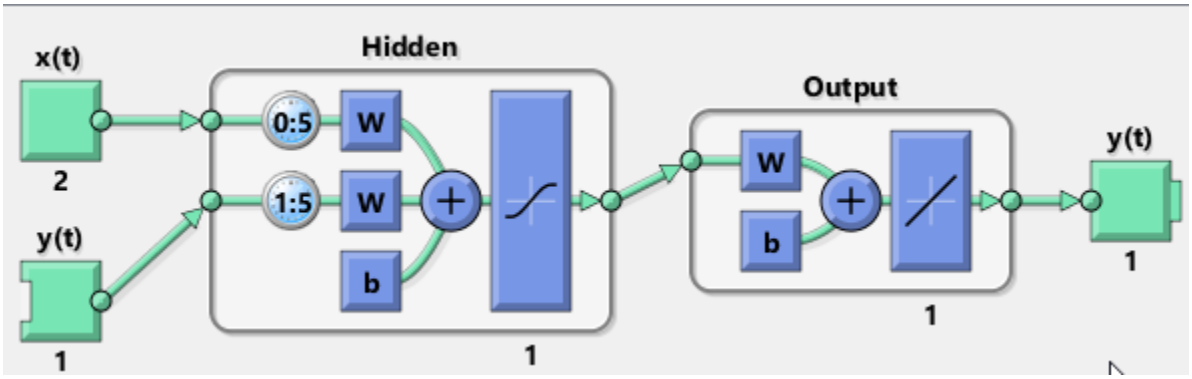

Figure 15 Architecture of narxnet

The figure below indicates how precise is our ANN. And the vertical line shows when the training process finishes, and the values of biases and weights become constant. There is a huge error exactly at the time the structure undergoes changes in stiffness. In other words, because the structure has changed, the ANN can not predict the answer precisely. Furthermore, even after that, the error is a little more than the error before the stiffness change.

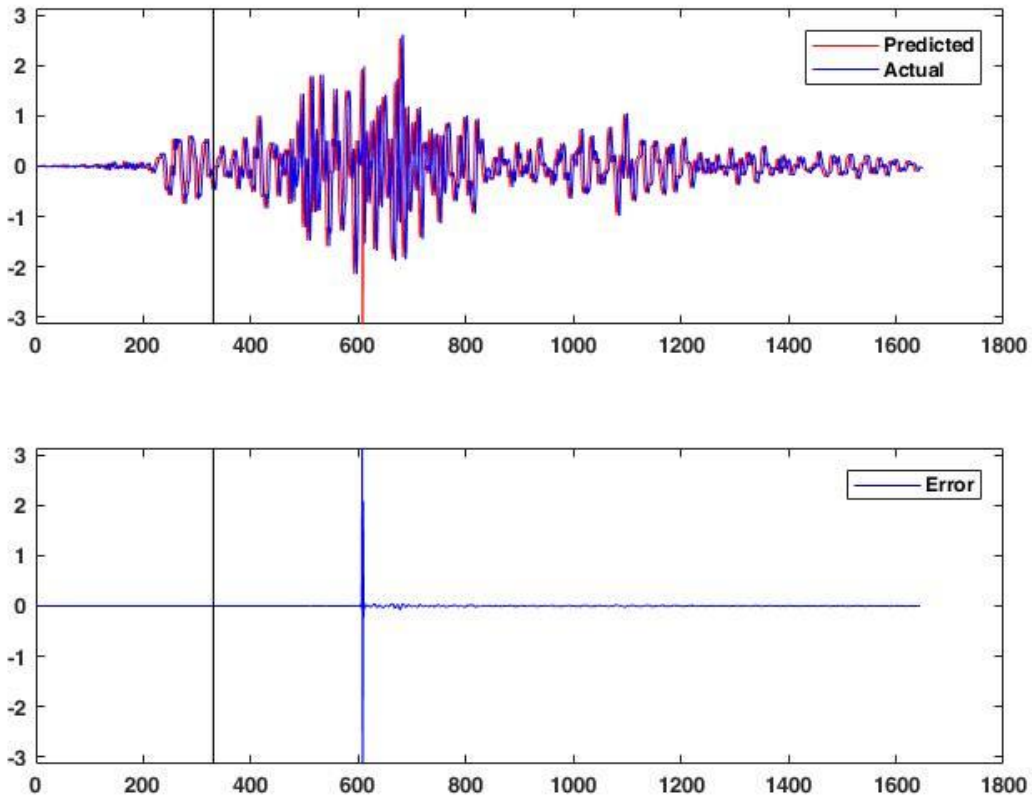

Figure 16Predicted values and errors

Evaluation and Results

Our goal is to evaluate of using ANN as a fruitful approach for live predicting the behavior of the structure, in particular when we don't have precise information of engineering parameters of the structure, including mass, stiffness, and damping. Due to the no-stationarity of the earthquake record it is usually not an easy task. Furthermore, we don't have any information about the mass, weight or damping of the structure. In this regard, ANN comes to play to predict the behavior of the structure that we don't have that much of information about it. The first example is just a rudimentary example about ANN. However, the two other examples are more advanced and show the application of ANNs in civil engineering and health monitoring of the structure. Moreover, thanks to the fastness of this method, it can be considered as live procedure for monitoring the structure.

References

[1] C. M. Bishop, *Neural Networks for Pattern Recognition*. Oxford University Press, Inc., 1995.

[2] A. C. Guyton and J. E. (John E. Hall, *Textbook of medical physiology*. Elsevier Saunders, 2006.

[3] S. M. Vazirizade, S. Nozhati, and M. A. Zadeh, "Seismic reliability assessment of structures using artificial neural network," *J. Build. Eng.*, vol. 11, 2017.

[4] S. Suresh, S. N. Omkar, R. Ganguli, and V. Mani, "Identification of crack location and depth in a cantilever beam using a modular neural network approach," *Smart Mater. Struct.*, vol. 13, no. 4, pp. 907–915, Aug. 2004.

[5] X. Fang, H. Luo, and J. Tang, "Structural damage detection using neural network with learning rate improvement," *Comput. Struct.*, vol. 83, no. 25–26, pp. 2150–2161, Sep. 2005.

[6] B. Xu, Z. Wu, G. Chen, and K. Yokoyama, "Direct identification of structural parameters from dynamic responses with neural networks," *Eng. Appl. Artif. Intell.*, vol. 17, pp. 931–943, 2004.

[7] H. Adeli, "Neural networks in civil engineering: 1989--2000," *Comput. Civ. Infrastruct. Eng.*, vol. 16, no. 2, pp. 126–142, 2001.