**INSTITUTO TECNOLOGICO Y DE ESTUDIOS SUPERIORES DE OCCIDENTE**

Imagen que contiene Logotipo

Descripción generada automáticamente

**MASTER IN DATA SCIENCE**

**REPORT #3:**

**Model Selection, Theoretical Framework and Scenario Study**

Advisors: PhD Jorge Arturo Pardiñas Mir, PhD Riemann Ruiz Cruz

Presents: Engineer Adrián Ramos Pérez

adrian.ramos@iteso.mx

April 25th, 2022

CONTENT TABLE

[Description of the model 3](#_Toc103639604)

[Cost Function 4](#_Toc103639605)

[Learning Algorithm 4](#_Toc103639606)

[Experiment description 4](#_Toc103639607)

[Preliminary Results 5](#_Toc103639608)

[Partial conclusions 7](#_Toc103639609)

[Next steps 7](#_Toc103639610)

[Bibliography 8](#_Toc103639611)

# Description of the model

The proposed model is a Time Delay Neural Network which consists of a dense neural network whose input is a signal delayed n number of lags.

For the problem we aim to solve, the input X is a matrix containing several delays of the speech signal convolved with the impulse response. The output vector Y is the original signal prior to have been affected by the room’s response. The neural model objective is thus try to learn the room behavior and fit it to the data.

Interfaz de usuario gráfica

Descripción generada automáticamente

Figure 1. Signal path

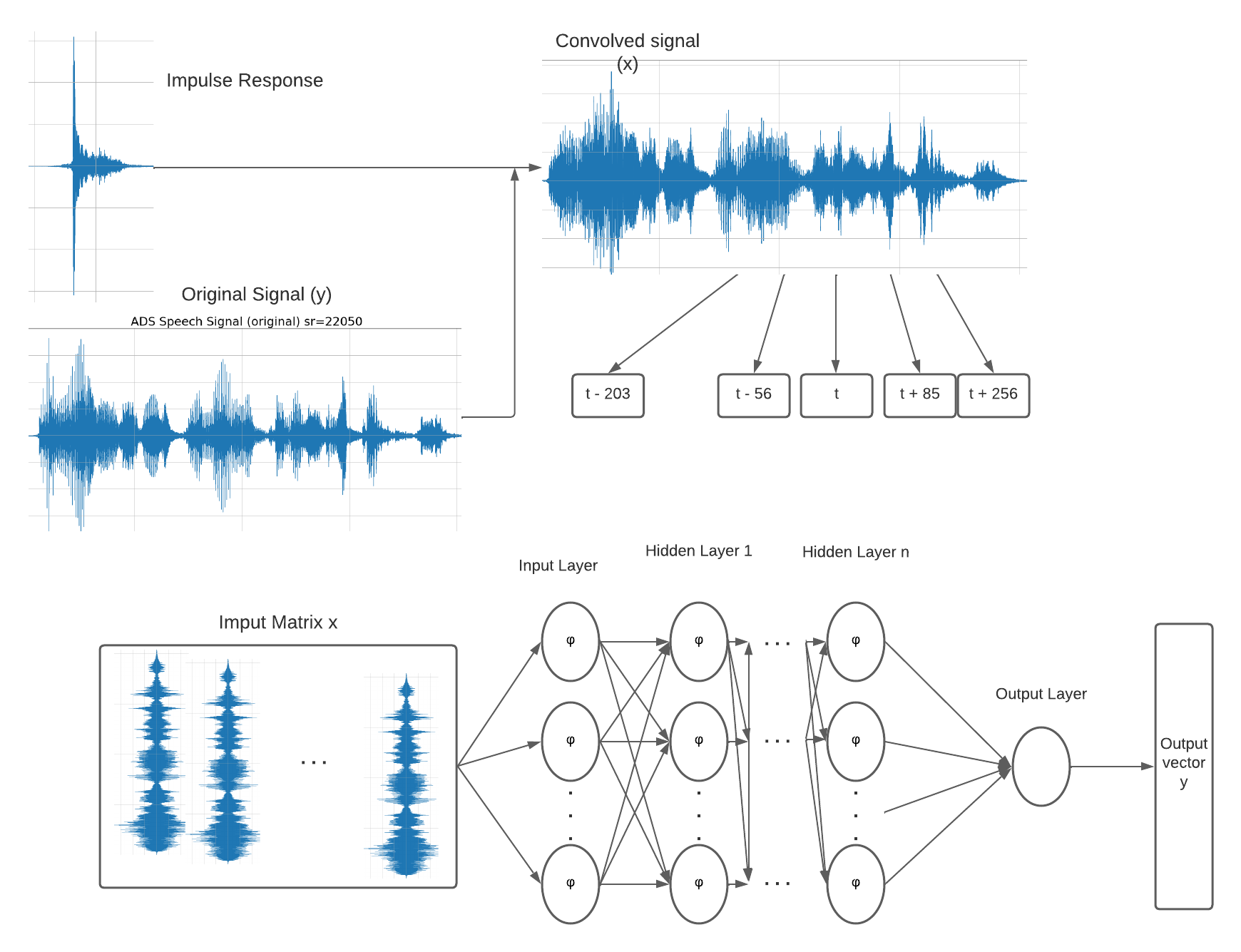


Figure 2. Neural Network with proposed Time-Delay arquitecture

## Cost Function

The objective function or cost function defines a model: is the aim of the model to minimize this function and then when it reaches its minimum value, the neural model is considered to have already learned behavior of data to be fitted.

For a Regression problem the cost function is the mean square error:

## Learning Algorithm

Learning algorithm determines how the objective function is going to be optimized.

Gradient descent is used as the training algorithm because it is the one implemented in Keras library for Python, although Levenberg-Marquart has proven to be better but is extensive computationally.

ADAM or Adaptive Moment Estimation optimizer has been used as it is at the time the best algorithm to optimize and help descent gradient converge.

## Experiment description

Several runs of the model have been performed, running the model while varying input matrix of lags and the structure of the network adding layers and varying number of neurons per hidden layer.

# Preliminary Results

Several runs of the model have been performed, running the model while varying input matrix of lags and the structure of the network adding layers and varying number of neurons per hidden layer. A noticeable difference can be spotted and listened to while the number of lags lies between 1250 and 2500. When it is above 7500 it comes down again.

However, gradient descent is not the best option regarding learning algorithms, so others might need to be tested.

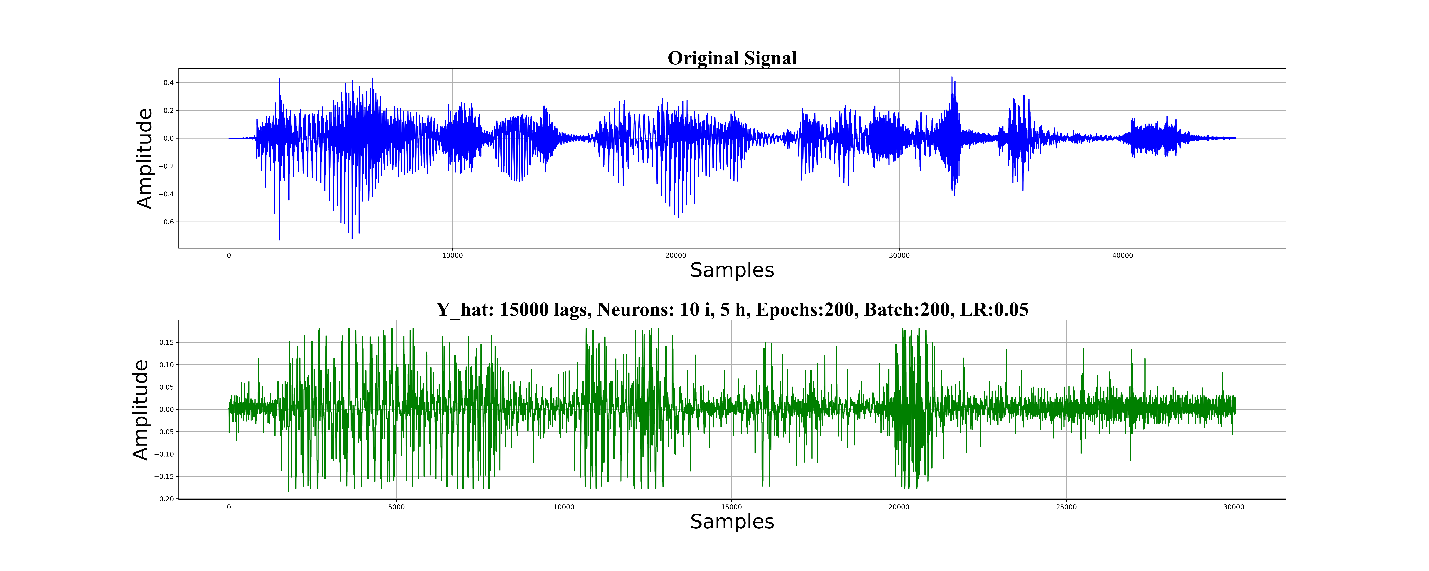
In the next figure, the evolution of the learning process of the model can be appreciated:

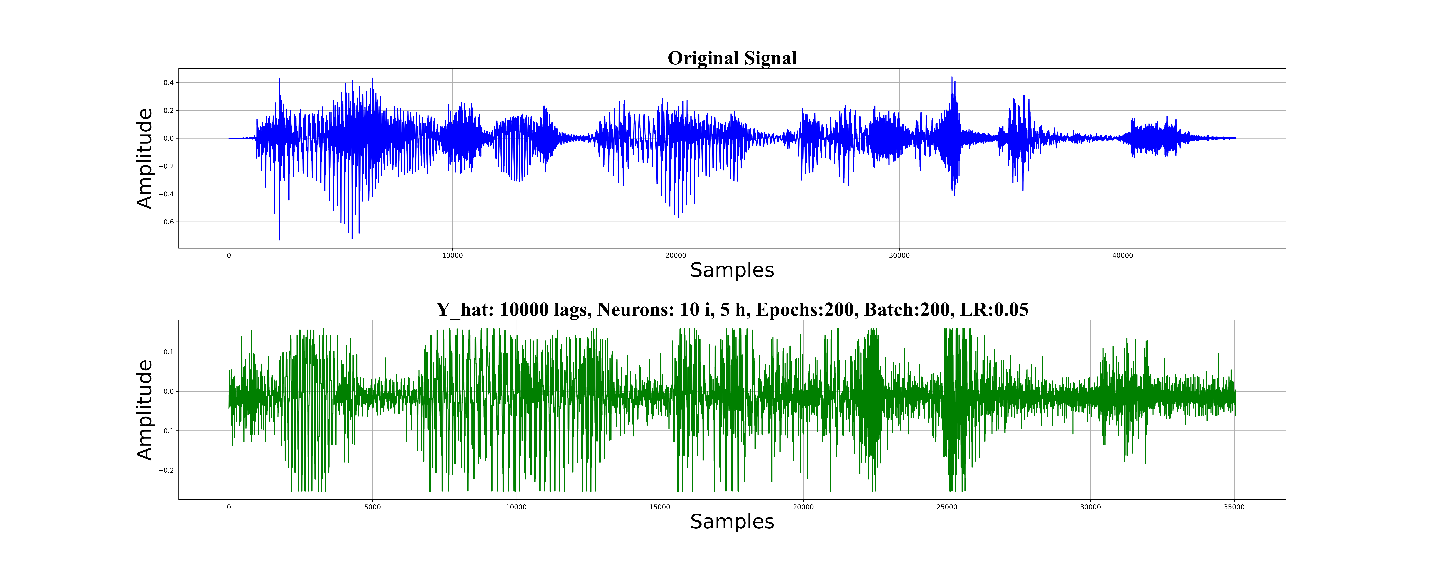
Gráfico

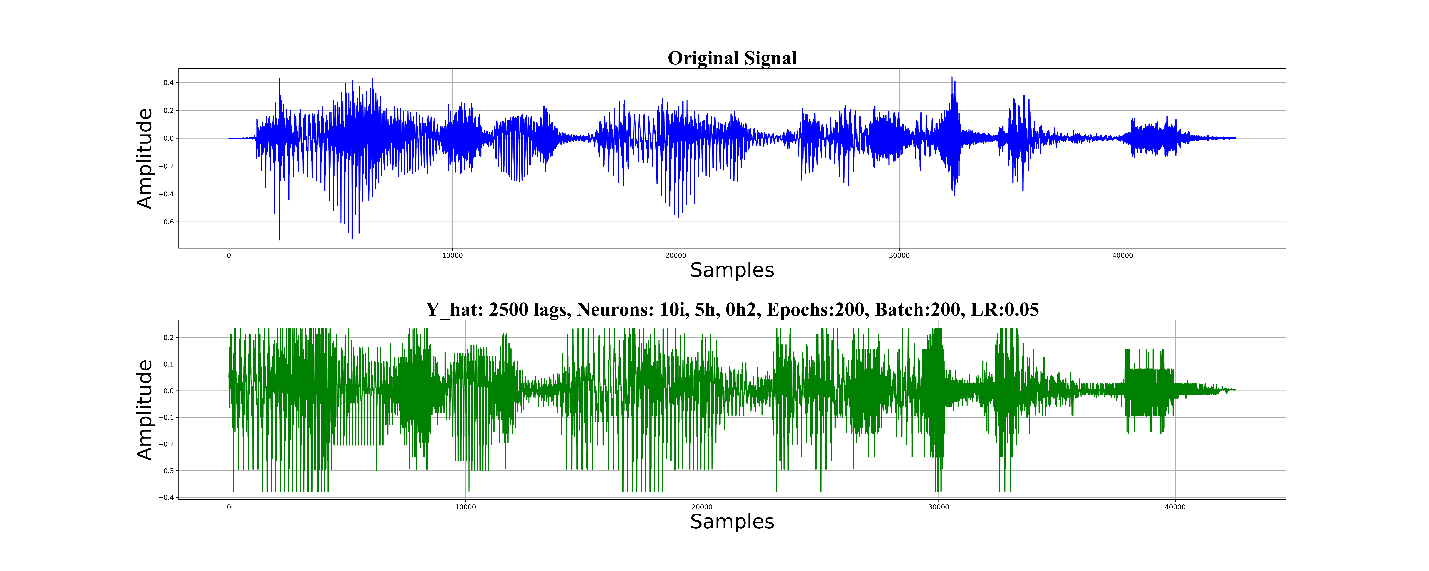
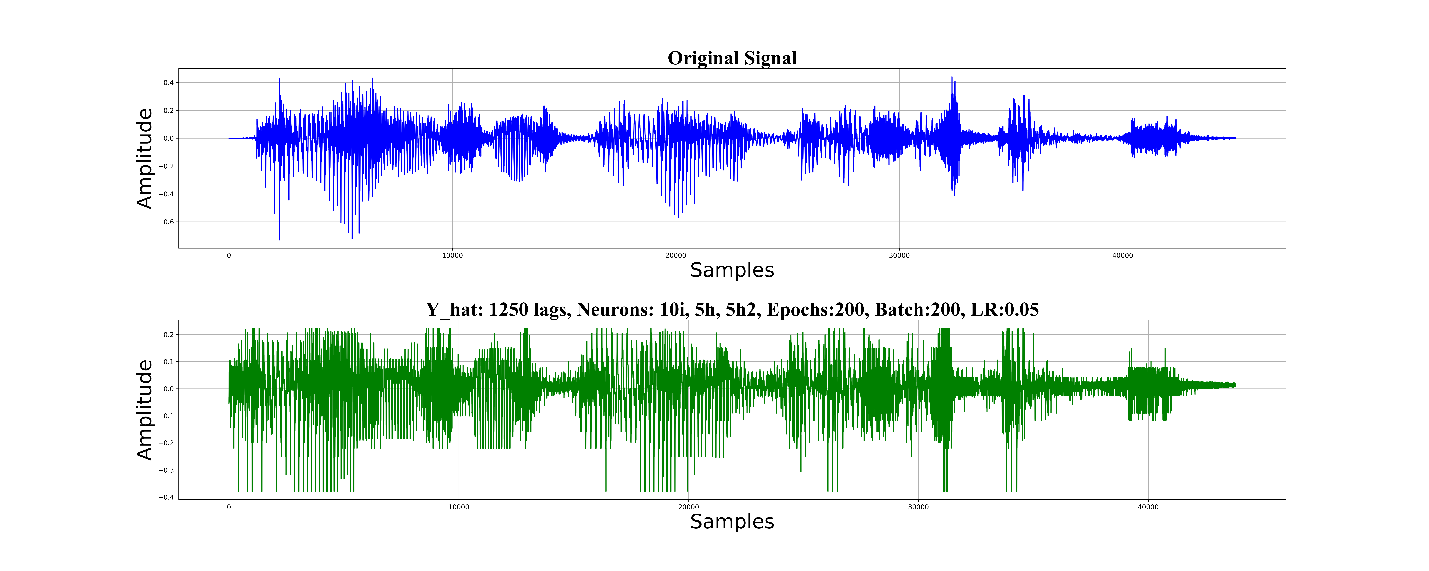
Descripción generada automáticamente

Ilustración 1. Loss function and Mean-squared Error through learning process.

Variation of predicted signal can be seen on the next figures:

Imagen que contiene Interfaz de usuario gráfica

Descripción generada automáticamente



In the next table, the performance comparison between some of the more representative of the conducted test can be appreciated, paying special attention to R2 metric and number of lags and interchangeable neurons:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Lags** | **Hidden Layer 1 Neurons number** | **Hidden Layer 1 Neurons number** | **Hidden Layer 1 Neurons number** | **Learning rate** | **Epoch** | **Batch size** | **Momentum** | **R2** | **Comments** |
| 1250 | 10 | 5 | 0 | 0.5 | 200 | 200 | 0.8 | 0.7571 |  |
| 2500 | 10 | 5 | 0 | 0.5 | 200 | 200 | 0.8 | 0.8473 | Best |
| 2500 | 10 | 5 | 5 | 0.5 | 200 | 200 | 0.8 | 0.8430 | Better |
| 7500 | 10 | 5 | 0 | 0.5 | 200 | 200 | 0.8 | 0.4009 |  |
| 7500 | 25 | 5 | 0 | 0.5 | 200 | 200 | 0.8 | 0.7390 | Better |
| 10000 | 10 | 5 | 0 | 0.5 | 200 | 200 | 0.8 | 0.6554 |  |
| 15000 | 10 | 5 | 0 | 0.5 | 200 | 200 | 0.8 | 0.5777 | Worse |

# Partial conclusions

The number of lags seems very relevant, it is needed to define the optimum number of lags while prevent overfitting.

More lags not necessarily means better performance.

The model behaviour seems promising, despite the results has not been the best yet. More parameter testing need to be done.

Other approaches using neural networks can be followed, as using spectrograms and Convolutional Neural Networks, but they need to be explored yet and compared to the current TDNN model.

# Next steps

Several next steps are still required to fit the best solution to the problem. Mainly they are as follow:

* Augment sampling frequency to 44.1 kHz.
* Determine optimum number of lags without overfitting.
* Find cause of clipping.
* Experiment with another learning algorithms, like Levenberg-Marquart
* Try ti predict using another test signals (generalize).
* Make a performance comparison against other models.
* Hyperparameter tunning
* Automatize tests.

# Bibliography

[1] Cecchi, S; Carini, A.; Spors, S. Room Response Equalization-A Review. Appl. Sci. 2018,8,16.