**INSTITUTO TECNOLÓGICO Y DE ESTUDIOS SUPERIORES DE OCCIDENTE**

Imagen que contiene Logotipo

Descripción generada automáticamente

**MAESTRÍA EN CIENCIA DE DATOS**

**REPORTE #2:**

**Análisis descriptivo de los datos y análisis comparativo del modelo**

Director: Dr. Jorge Arturo Pardiñas Mir

Presenta: Ing. Adrián Ramos Pérez

adrian.ramos@iteso.mx

November 22nd, 2021

TABLA DE CONTENIDO

[1. Análisis Descriptivo de los Datos 3](#_Toc89330807)

[1.1 Obtención de la Respuesta al Impulso 3](#_Toc89330808)

[2. Análisis Comparativo de Modelos 3](#_Toc89330809)

[Función de activación 4](#_Toc89330810)

[Regularización 4](#_Toc89330811)

[Convolutional Neural Network 4](#_Toc89330812)

[Particle Swarm Optimization 5](#_Toc89330813)

[Cálculo del Error y Eficiencia del Modelo 5](#_Toc89330814)

[Bibliografía 6](#_Toc89330815)

# Análisis Descriptivo de los Datos

Para entrada del sistema o modelo se usarán múltiples grabaciones con la respuesta al impulso unitario de un cuarto a ecualizar, acondicionado mínimamente en sus paredes para que su reflexividad sea baja (menor a un 36%) y permita hacer invertible la respuesta al impulso al ser de fase mínima [1].

El número de respuestas al impulso depende del número de fuentes y número de micrófonos. Para los experimentos se utilizarán dos fuentes (dos monitores de referencia de estudio) y un solo micrófono.

El vector de entrada está conformado por las muestras de las respuestas al impulso concatenadas.

El vector de salida está conformado por los coeficientes de un filtro FIR.

## Obtención de la Respuesta al Impulso

La respuesta al impulso del cuarto se obtendrá mediante un barrido senoidal del toolbox de MATLAB para procesamiento digital de señales. Este barrido senoidal aumenta su frecuencia con el tiempo en el rango de frecuencias seleccionado, que en este caso será de 20 Hz a 22 kHz. La señal es grabada y se utilizará la deconvolución para obtener la respuesta al impulso de el tono senoidal barrido.

## Tipo de Filtro Seleccionado (sección tentativa)

Filtro FIR (Finite Impulse Response): no tiene retroalimentación, lo que garantiza una respuesta finita al impulso.

* Se prefiere por ser de fase lineal y no alterar la fase de la señal.
* Eficiencia computacional. (Menos cálculos que un IIR, que pueden ser omitidos).
* Algunas respuestas no son prácticas para implementar con FIR, hay qué explorar esto para los casos en que sea más óptimo un IIR.

Filtro IIR (Infinite Impulse Response)

* Baja latencia para aplicación de control en tiempo real.
* Menor número de coeficientes.

# Análisis Comparativo de Modelos

Dentro de los algoritmos de *machine learning* existen dos enfoques de aprendizaje: ***supervisado*** y ***no supervisado***.

En el ***aprendizaje supervisado*** se cuenta con datos de entrada así como un vector de salida que sirve para retroalimentar a un modelo y definir el error, el objetivo es entrenar un modelo con features y labels de entrenamiento, para posteriormente usar nuevos features y predecir etiquetas.

En el ***aprendizaje no supervisado***, no se cuenta con variable de salida , ni con etiquetas, sólo con los datos de entrada de los cuales se pretende aprender la estructura de la data y encontrar un patrón o patrones que pueden resultar de interés.

Por la naturaleza del problema y que se necesita ecualizar un cuarto que se considera un sistema no estacionario, hay diversos métodos desde los que se puede abordar. Sin embargo es conveniente tomar el cuarto como un Sistema Lineal Invariante en el Tiempo

## Baseline techniques

Se utilizarán dos métodos tradicionales de machine learning para contrastar los resultados que se obtengan con algoritmos de Deep learning u otros.

### Support Vector Machine (SVM)

Las máquinas de soporte vectorial permiten resolver de manera eficiente problemas de optimización convexa

Kernel de base Radial

* Scikit learn API para la implementación.

## Convolutional Neural Network (CNN)

El enfoque tradicional de Deep learning son las redes neuronales

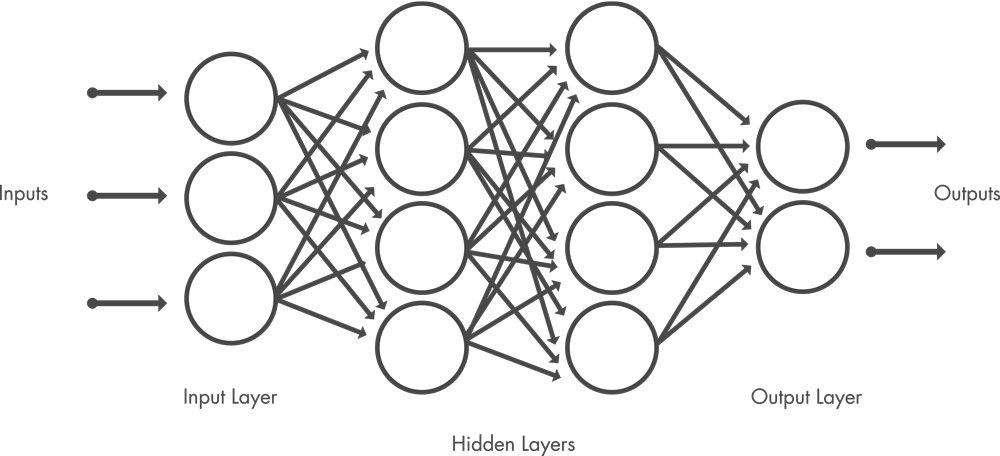


Ilustración 1. Arquitectura de una Red Neuronal densa

Entrenar a una red neuronal es un problema de optimización donde una función de pérdida es minimizada por propagación hacia atrás del error a través de la red neuronal.

**Existe una curva deseada en la respuesta al impulso**, la que se toma como la salida ideal y contra la que se compara la diferencia de la curva de respuesta en frecuencia alcanzada, **lo que representa la función de coste a minimizar**.

### Arquitectura de la Red

La arquitectura de la red consiste en capas convolucionales, capas de pooling

### Función de activación

Es importante que la función de activación sea no lineal porque en caso contrario el modelo se vuelve un simple sistema lineal, y la función de activación no lineal permite al modelo capturar precisamente no linealidades complejas, así como efectos de interacción.

Se implementará la función de activación ReLU (Rectified Linear Unit)

### Regularización

Es necesaria para prevenir el sobreajuste u overfitting del modelo a los datos de entrenamiento, para poder generalizarlo posteriormente.

Algunos métodos comunes de regularización son:

* Ridge
* Lasso
* Dropout

Para el presente proyecto se utilizará regularización por dropout, que consiste en descartar nodos con base los pesos sinápticos de las redes neuronales.

## 2.2.4 Cálculo del Error y Eficiencia del Modelo

El error medio cuadrado está dado por:

## Particle Swarm Optimization

Este algoritmo permite probar distintas soluciones candidatas. Con él se pueden así mismo obtener los coeficientes de un filtro que permita ecualizar la respuesta del cuarto

# Bibliografía

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

[2] Audio Equalization – Deep Neural Networks. Appl. Sci. 2020, 10, 2843.