



UNIVERSIDAD DE CHILE  
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS  
DEPARTAMENTO DE INGENIERÍA MECÁNICA

**DESARROLLO DE MODELOS BASADOS EN REDES NEURONALES PARA  
LA PREDICCIÓN DE BAND GAPS EN PANELES DE METAMATERIALES**

MEMORIA PARA OPTAR AL TÍTULO DE INGENIERA CIVIL MECÁNICA

**PAULA CAROLINA MONTSERRAT RAMÍREZ SALAS**

PROFESORA GUÍA:  
VIVIANA MERUANE NARANJO

PROFESOR CO-GUÍA:  
RAFAEL RUÍZ GARCÍA

COMISIÓN:  
RUBÉN FERNÁNDEZ URRUTIA

Este trabajo ha sido parcialmente financiado por los proyectos Núcleo Milenio en Soft  
Smart Mechanical Metamaterials y Fondecyt 1210442

SANTIAGO DE CHILE  
2021

RESUMEN DE LA MEMORIA PARA OPTAR  
AL TÍTULO DE INGENIERA CIVIL MECÁNICA  
POR: PAULA RAMÍREZ SALAS  
FECHA: AGOSTO 2021  
PROF. GUÍA: VIVIANA MERUANE NARANJO

## **DESARROLLO DE MODELOS BASADOS EN REDES NEURONALES PARA LA PREDICCIÓN DE BAND GAPS EN PANELES DE METAMATERIALES**

Los metamateriales son materiales artificiales cuyas propiedades provienen directamente de la posición y geometría que ocupan sus componentes dentro de su estructura. Con metamateriales en paneles para control de vibraciones, se pueden lograr bandas de frecuencias (band-gaps) en que las vibraciones son suprimidas por el panel. Esto es de gran interés en casos donde se quiere evitar la vibración de paneles debido a una excitación externa.

La evaluación de los band gaps en elementos finitos es muy costosa computacionalmente, esto hace que no sea factible implementar algoritmos de optimización tradicionales para maximizar el band gap a una cierta frecuencia, por ejemplo. Debido a esto es que se plantea como objetivo entrenar modelos de redes neuronales para predecir band gaps en paneles de metamateriales. En ese mismo sentido, se desprende también la generación de la base de datos para el entrenamiento y la definición del modelo de redes neuronales que mejor se ajuste al problema. Las redes neuronales corresponden a un algoritmo de aprendizaje automático de rápida evaluación, algo que se podría utilizar, posteriormente, para la optimización topológica de los metamateriales.

La ejecución del presente trabajo comprendió dos ejes: la generación de las bases de datos y el desarrollo de modelos de redes neuronales. La generación de bases de datos lo suficientemente robustas, se realizó a partir de los modelos numéricos de distintos paneles diseñados en trabajos previos relacionados al tema y siguiendo 2 enfoques para el problema de predicción. En tanto, el desarrollo de modelos contempla la implementación de distintas arquitecturas que serán evaluadas para seleccionar la que obtenga los mejores resultados en la predicción de band-gaps.

Como resultados, se determina que modelos basados en redes neuronales logran predecir exitosamente el ancho y frecuencia central de los band-gaps. En particular, se aprecia que los modelos predicen más fácilmente estas propiedades en paneles que poseen resonadores internos en sus núcleos.

*A mi mamá, Chilita, Mimi y Jainij.*

# Agradecimientos

Quiero iniciar agradeciendo a las personas que no alcanzaron a estar conmigo para celebrar el término de esta etapa en mi vida. Muchísimas gracias a mis abuelos, a mis tíos Juan y Miriam, y a mi padrino Héctor, por acompañarme y estar siempre conmigo. Los llevo siempre en mi corazón.

Creo que no me alcanzan las palabras para agradecerle por todo a mi mamá. Sin su apoyo, enseñanzas, ánimos, oraciones y llamadas despertándome para ir a clases, no hubiese llegado a esta instancia. Simplemente, gracias por existir y siempre creer en mí. Gracias por ser tú.

Infinitas gracias a mi madrina Lidia, por todas las velitas que ha prendido y todas las oraciones realizadas en estos 25 años, las que me acompañaron en todos los procesos que permitieron llegar en donde estoy ahora. De igual manera, le agradezco a mi tía Jimena, por los buenos deseos y estar siempre atenta a cómo estoy.

Agradecer también a las personas que he conocido en Beauchef. A mis amigos de mecánica, Byron, Carlota, Cristóbal, Emiliano, Javier, Nathaniel, Ralf, Sebastián y Víctor, gracias por estar cuando lo necesité, por escucharme y por siempre hacerme reír. Agradecerle a Heinrich, robándole las palabras de su tesis, porque es verdad que sin tu apoyo incondicional (sobretudo emocional), esta tesis no podría haberse realizado. Gracias por todos estos años.

Por último, agradecerle a la profesora Viviana por sus consejos y recomendaciones para la realización de este trabajo. Gracias por estar siempre disponible para escucharme y ayudarme.

# Tabla de Contenido

<b>1. Introducción</b>	<b>1</b>
1.1. Antecedentes Generales	1
1.2. Objetivos	2
1.2.1. Objetivo General	2
1.2.2. Objetivos Específicos	2
1.3. Alcances	2
<b>2. Antecedentes</b>	<b>3</b>
2.1. Metamateriales	3
2.2. Estructuras celulares periódicas	3
2.3. Band Gaps Fonónicos	5
2.4. Aprendizaje Automático	6
2.5. Redes Neuronales	7
2.5.1. Tipos de redes neuronales	9
2.5.2. Tipos de problemas	9
2.5.3. Entrenamiento y pruebas	11
2.5.4. Métodos de regularización	11
2.5.5. Preprocesamiento de los datos	13
2.5.6. Selección de hiperparámetros	13
2.5.7. Métricas de evaluación	13
<b>3. Formulación</b>	<b>15</b>
3.1. Geometrías	15
3.1.1. Estructura 1	15
3.1.2. Estructura 2	16
3.1.3. Estructura 3	17
3.2. Cálculo de band-gaps	17
<b>4. Metodología</b>	<b>20</b>
4.1. Generación de las bases de datos	20
4.1.1. Enfoque I: Predicción de propiedades del band-gap	20
4.1.2. Enfoque II: Predicción del diagrama de bandas	21
4.2. Desarrollo de modelos de redes neuronales	22
<b>5. Resultados</b>	<b>24</b>
5.1. Enfoque I	24

5.1.1. Estructura 1	24
5.1.2. Estructura 2	26
5.1.3. Estructura 3	28
5.2. Enfoque II	29
5.2.1. Estructura 2	30
5.2.2. Estructura 3	31
<b>6. Discusión y Análisis</b>	<b>34</b>
6.1. Enfoque I	34
6.2. Enfoque II	35
<b>7. Conclusiones</b>	<b>37</b>
7.1. Trabajos Futuros	37
<b>Bibliografía</b>	<b>39</b>
<b>Anexo A. Códigos</b>	<b>42</b>

# Índice de Tablas

3.1. Propiedades mecánicas de los materiales utilizados en cada estructura en estudio.	15
5.1. Registro de desempeño de los modelos para la predicción del ancho de banda del band-gap en Estructura 1.	25
5.2. Registro de desempeño de los modelos para la predicción de la frecuencia media del band-gap en Estructura 1.	25
5.3. Hiperparámetros del Modelo Fully Connected para la Estructura 1.	26
5.4. Hiperparámetros del Modelo CNN para la Estructura 1.	26
5.5. Registro de desempeño de los modelos para la predicción del ancho de banda del band-gap.	26
5.6. Registro de desempeño de los modelos para la predicción de la frecuencia media del band-gap.	27
5.7. Hiperparámetros del Modelo Fully Connected para la Estructura 2.	28
5.8. Hiperparámetros del Modelo CNN para la Estructura 2.	28
5.9. Registro de desempeño de los modelos para la predicción del ancho de banda del band-gap.	28
5.10. Registro de desempeño de los modelos para la predicción de la frecuencia media del band-gap.	29
5.11. Hiperparámetros del Modelo Fully Connected para la Estructura 3.	30
5.12. Hiperparámetros del Modelo CNN para la Estructura 3.	30
5.13. Registro de desempeño del modelo para la predicción de las propiedades del band-gap en Estructura 2.	30
5.14. Hiperparámetros para el modelo de la Estructura 2.	31
5.15. Registro de desempeño del modelo para la predicción de las propiedades del band-gap en Estructura 3.	32
5.16. Hiperparámetros para el modelo de la Estructura 3.	33

# Índice de Ilustraciones

2.2.1.Material 3D con diferentes periodicidades. <b>[1]</b> . . . . .	4
2.2.2.Ejemplos de distintas topologías de enrejados configuradas como núcleos de estructuras de paneles sándwich. <b>[2]</b> . . . . .	5
2.3.1.Diagrama de bandas. <b>[3]</b> . . . . .	6
2.4.1.Proceso de Aprendizaje Automático. . . . .	7
2.5.1.Una neurona artificial que calcula una suma ponderada de sus entradas y luego aplica una función escalón. <b>[4]</b> . . . . .	8
2.5.2.Arquitectura de una red neuronal. <b>[5]</b> . . . . .	8
2.5.3.Arquitectura típica de una red neuronal convolucional. <b>[4]</b> . . . . .	9
2.5.4.En la clasificación binaria, la red neuronal tiene solo una salida para predecir 2 clases. En cambio, en la clasificación multiclase, se puede tener N salidas. <b>[6]</b> . . . . .	10
2.5.5.Modelo de regresión para la predicción del precio de casas. <b>[6]</b> . . . . .	10
2.5.6.Representación gráfica del método de dropout sobre una red neuronal estándar. <b>[7]</b> . . . . .	12
2.5.7.Regularización por early stopping. <b>[4]</b> . . . . .	12
3.1.1.Estructura 1. . . . .	16
3.1.2.Estructura 2: Sistema tipo voladizo en masa . . . . .	16
3.1.3.Estructura 3: Sistema tipo enrejado con masas suspendidas . . . . .	17
3.2.1.Ejemplo de estructura con periodicidad infinita. A la izquierda se muestra su simetría reflectiva con respecto a una recta en $45^\circ$ que pasa por el centro. A la derecha se presenta la Zona Irreducible de Brillouin. . . . .	18
3.2.2.Diagrama de bandas con la presencia de un band-gap . . . . .	18
4.1.1.Diagrama de flujo para la primera etapa del Enfoque I . . . . .	21
4.1.2.Diagrama de flujo para la primera etapa del Enfoque II . . . . .	22
4.2.1.Diagrama de flujo para el enfoque I . . . . .	23
4.2.2.Diagrama de flujo para el enfoque 2 . . . . .	23
5.1.1.Comparación entre el ancho de banda real y el predicho, según los modelos entrenados con la Estructura 1. . . . .	25
5.1.2.Comparación entre la frecuencia media real y la predicha, según los modelos entrenados con la Estructura 1. . . . .	25
5.1.3.Comparación entre el ancho de banda real y el predicho, según los modelos entrenados con la Estructura 2. . . . .	27
5.1.4.Comparación entre la frecuencia media real y la predicha, según los modelos entrenados con la Estructura 2. . . . .	27



5.1.5.Comparación entre el ancho de banda real y el predicho, según los modelos entrenados con la Estructura 3. . . . .	29
5.1.6.Comparación entre la frecuencia media real y la predicha, según los modelos entrenados con la Estructura 3. . . . .	29
5.2.1.Comparación entre el ancho de banda real y el predicho para la Estructura 2	31
5.2.2.Comparación entre la frecuencia media real y la predicha para la Estructura 2	31
5.2.3.Comparación entre el ancho de banda real y el predicho para la Estructura 3	32
5.2.4.Comparación entre la frecuencia media real y la predicha para la Estructura 3	32



# Capítulo 1

## Introducción

### 1.1. Antecedentes Generales

Los metamateriales, estructuras compuestas artificiales con propiedades materiales exóticas, han surgido como una nueva frontera de la ciencia que involucra la física, la ciencia de los materiales, la ingeniería y la química. Su arquitectura es diseñada de forma inteligente ya que, a diferencia de los materiales convencionales, las propiedades de los metamateriales provienen directamente de la posición y la geometría que ocupan sus componentes.

El interés en la investigación de metamateriales explotó a principios de los años 2000, cuando los científicos comenzaron estudiar los efectos de las ondas electromagnéticas sobre este tipo de materiales. Los resultados experimentales llevaron a concluir que la existencia de periodicidad de las estructuras, inhibe la propagación de ondas electromagnéticas.

En investigaciones posteriores con propagación de ondas mecánicas, se concluye que la periodicidad en los metamateriales conduce a interferencias constructivas o destructivas dependiendo de la frecuencia de las ondas. Esto produce rangos de frecuencias en las que se permite que las ondas mecánicas se propaguen o se bloqueen en cualquier dirección admisible. A esto último se le denomina *Band Gap Fonónico* y su determinación es gran interés en la actualidad ya que con ellos se permitiría controlar y manipular ondas acústicas y elásticas que se propagan por este tipo de materiales.

Es aquí que las nuevas herramientas computacionales juegan un rol importante en este campo de investigación. El aprendizaje automático surge como el estudio de algoritmos computacionales que mejoran automáticamente a través de la experiencia, es decir, que van aprendiendo de los datos sin estar programado explícitamente para hacerlo. El uso de algoritmos de aprendizaje automático podría ser crucial en la predicción de estos band gaps, no solo determinando su existencia en determinados paneles sino que también indicando su frecuencia promedio y ancho de la banda.

## 1.2. Objetivos

### 1.2.1. Objetivo General

Entrenar algoritmos de machine learning para la predicción de band gaps en paneles de metamateriales.

### 1.2.2. Objetivos Específicos

- Generar las bases de datos para el entrenamiento a partir de modelos de elementos finitos de paneles de metamateriales.
- Determinar los parámetros de entrada y salida relevantes de los modelos de redes neuronales para la formulación del problema.
- Comparar y analizar de los resultados las diferentes arquitecturas de redes neuronales.
- Definir el mejor modelo de redes neuronales para la predicción de band-gaps.

## 1.3. Alcances

Para la realización del presente trabajo, se cuenta con los modelos numéricos de los paneles de metamaterial a analizar gracias a trabajos de título previos enfocados en este campo de investigación. Estos corresponden a estructuras bidimensionales de tipo enrejado.

Además, se limita al uso de redes neuronales como algoritmo de aprendizaje automático y metamateriales en materiales mecánicos con vibraciones.

# Capítulo 2

## Antecedentes

### 2.1. Metamateriales

Los metamateriales comprenden un nuevo campo de investigación interdisciplinaria, que se ocupa de materiales compuestos artificiales diseñados para mostrar propiedades físicas que superan (o complementan) a las disponibles en la naturaleza. El prefijo griego *meta* en su nombre, indica la noción de 'después' o 'más allá' [8]. Esto está relacionado con la comprensión de que los metamateriales exhiben propiedades que generalmente se consideran que van más allá lo que se espera encontrar en materiales convencionales.

Ejemplos de propiedades tan inusuales incluyen la generación de rangos de frecuencia donde una vibración es incapaz de transmitirse (band gaps fonónicos) e índices de refracción negativos. En los metamateriales, las propiedades provienen directamente de la posición y geometría que ocupan sus componentes dentro de su estructura, no así de su composición. A diferencia de los materiales compuestos tradicionales, en los metamateriales las inserciones de las segundas fases ocupan posiciones estratégicas, definiendo patrones que se repiten de forma periódica. Esto permite obtener propiedades globales diferentes a las propiedades propias de cada componente individual [9]. La presencia de periodicidad en estos materiales permite la descripción intrínseca, basada en celda unitaria, de propiedades dinámicas y características de propagación de ondas, así como orden en la introducción de los elementos de resonancia local [10].

Desde su aparición a finales de los años 90, los metamateriales han atraído mucha atención de la comunidad científica de todo el mundo. Esto debido a que este tipo de materiales pueden minimizar la respuesta vibratoria, algo que los hace ideales en aplicaciones de aislación acústica y protección de vibraciones. Por otro lado, en ellos es posible maximizar la respuesta en una área específica, evitando que la vibración se propague hacia otras zonas.

### 2.2. Estructuras celulares periódicas

Las estructuras celulares periódicas son materiales que se obtienen del teselado de una celda unitaria que comprende un arreglo de vigas o barras, las que pueden representar en su

interior periodicidad de las celdas en una dimensión, dos dimensiones (tipo panel) o en tres dimensiones (enrejados o lattices). Valencia [1] ejemplificó a través de legos la periodicidad de un material. Las figuras 2.1 (a)-(c) muestran un material tridimensional con periodicidad en una, dos y tres dimensiones. Tales materiales periódicos están completamente descritos por un enrejado y por una celda unitaria elemental. A su vez, el enrejado está definido por un conjunto de vectores base (Fig. 2.1 (d)), que permiten la construcción del material a través de la aplicación sucesiva de operaciones de traslación de la celda unitaria según el vector  $\mathbf{a}$  del enrejado. (lego azul).

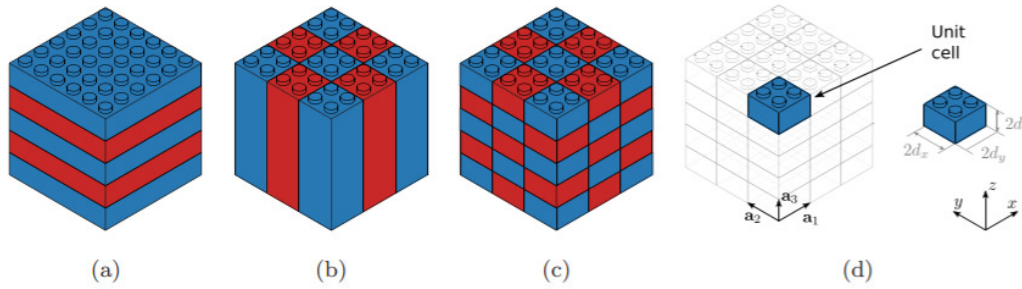


Figura 2.2.1: Material 3D con diferentes periodicidades. [1]

Los materiales celulares se utilizan principalmente en aplicaciones en las cuales se necesita tener una baja densidad, en adición con una alta rigidez y resistencia, como también en el control de la transferencia de calor y una mayor absorción de energía mecánica [11]. Alcanzar las propiedades descritas anteriormente, depende de factores relacionados directamente con la celda, tales como su topología, número, tamaño, cantidad de uniones de elementos y espesor de los elementos, como también el proceso de fabricación del material y las condiciones de borde y carga.

Entre las posibles aplicaciones estructurales que poseen este tipo de materiales, se cuenta su uso como núcleo de paneles tipo sándwich. Un panel sándwich es una estructura que tiene 2 placas delgadas de material rígido en los bordes, las cuales rodean núcleo de material celular de baja densidad [12]. Este núcleo debe tener la rigidez suficiente tanto para mantener la distancia entre las placas, como también para evitar el desplazamiento entre ellas ante la flexión del panel. Por otra parte, la función de las placas es resistir los esfuerzos de tracción y compresión en el sándwich. La figura 2.2 muestra seis ejemplos de topologías celulares de enrejados utilizados como núcleos de paneles sándwich.

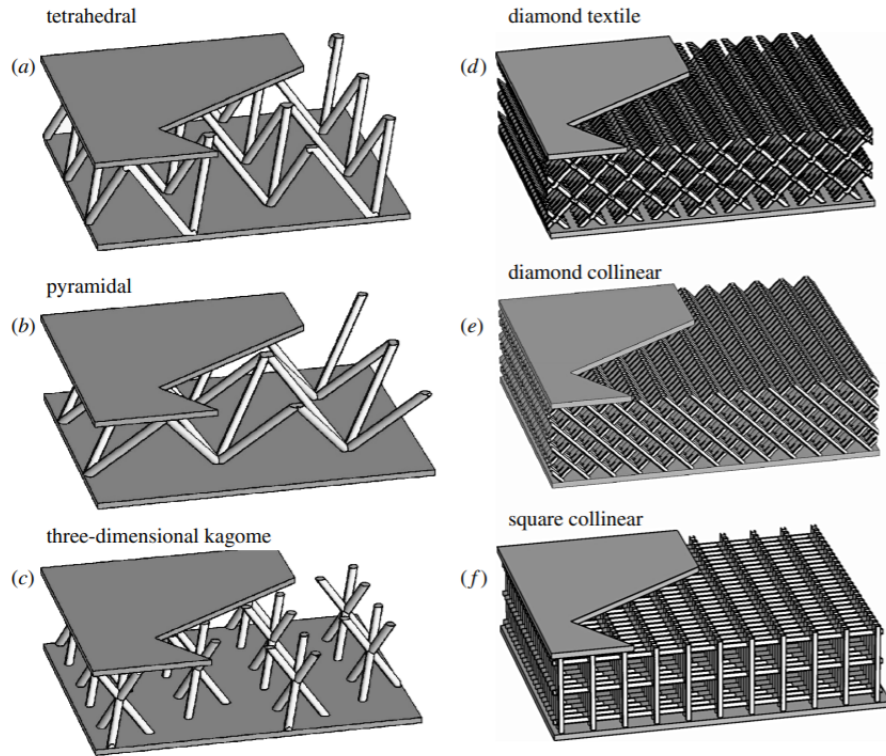


Figura 2.2.2: Ejemplos de distintas topologías de enrejados configuradas como núcleos de estructuras de paneles sándwich. [2]

### 2.3. Band Gaps Fonónicos

La periodicidad en los metamateriales conduce a interferencias constructivas o destructivas dependiendo de la frecuencia de las ondas. Cuando las ondas se dispersan dentro del panel, las ondas son fuertemente dispersadas de un material a otro y se producen ondas secundarias en cada interfaz entre ellos. Las ondas secundarias se reflejan hacia atrás e interfieren entre sí de forma constructiva o destructiva. Esto produce rangos de frecuencias en las que se permite que las ondas originales se propaguen (bandas de paso) o en las que las ondas originales se bloquean en una dirección (bandas de parada) o se bloquean en cualquier dirección. A esto último se le denomina *Band Gap Fonónico* [13] [14]. El trabajo realizado por Sigalas [15] en 2005 habla de la existencia de estos band gaps, concluyendo que eran un efecto combinado de la difracción de Bragg y de Scattering de las ondas, los cuales son mecanismos activados por la geometría y periodicidad del material.

Así, los band gaps fonónicos están estrechamente relacionados con la geometría y la frecuencia de las ondas. Como resultado, la determinación de estos band gaps es de gran interés ya que éstos permiten controlar y manipular ondas acústicas y elásticas que se propagan por este tipo de material.

Los band gaps fonónicos se pueden identificar a través de diagramas de banda, que representan gráficamente las superficies de dispersión de la onda. Para la construcción de estos diagramas, se necesita que el material esté previamente definido a partir de un vector de

ondas  $\mathbf{k}$ , el cual entrega información de la repetición de la celda, en particular, su forma y simetría [16]. Entonces, el diagrama consiste en graficar los valores propios o frecuencias naturales en función de un vector de ondas.

En la figura 2.3 se puede observar un diagrama de bandas típico. Este diagrama corresponde al de un conjunto de cilindros paralelos idénticos sumergidos en agua, el cual fue sometido a ondas propagándose en la dirección normal a los cilindros [3]. El eje de las abscisas contiene las direcciones de propagación de onda y el eje de las ordenadas las frecuencias con que la onda se propagaría en cada dirección. En dicho diagrama, se puede identificar la presencia de un band gap del orden de 100[Hz] (región en rojo). Entonces, una onda con un rango de frecuencias que se encuentre en dentro de esa región, no cuenta con direcciones admisibles para su propagación.

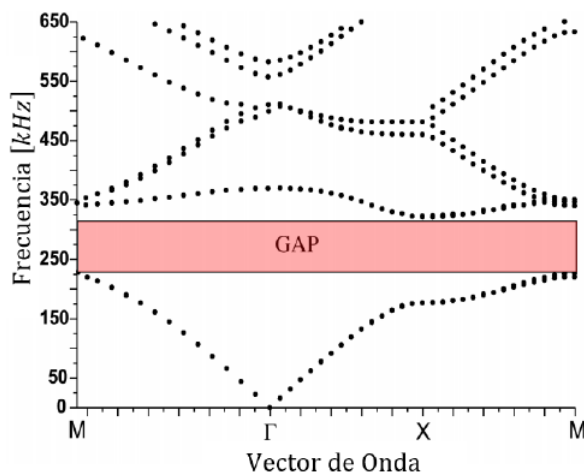


Figura 2.3.1: Diagrama de bandas. [3]

## 2.4. Aprendizaje Automático

Un algoritmo es una secuencia de instrucciones que se llevan a cabo para transformar un input a un output. Para poder resolver un problema en un computador se necesita un algoritmo, pero hay ocasiones en que no se cuenta con alguno que permita dar solución a ciertas tareas. A pesar de aquello, lo que sí se tienen son datos y la intuición indica que lo que se podría hacer es *aprender* con ellos.

El aprendizaje automático, comúnmente referido como *Machine Learning*, es una rama de la Inteligencia Artificial que consiste en programar computadores para optimizar cierto criterio de rendimiento utilizando una base de datos o experiencias pasadas. Géron[4] habla de una definición más general, indicando que es un campo de estudio que le da a los computadores la habilidad de aprender sin ser explícitamente programado para ello. El aprendizaje automático se basa en un conjunto de algoritmos que pueden procesar una amplia variedad de datos que normalmente son difíciles de procesar a mano.



Esta disciplina encuentra su inspiración en la biología, ya que la máquina, representada por algoritmos, imita técnicas y comportamientos de aprendizaje humano o animal desde una perspectiva de pensamiento y respuesta. La mayor ventaja de incorporar técnicas de aprendizaje automático es el aspecto de automatización de enriquecer la base de conocimientos con técnicas de autoaprendizaje con una mínima intervención humana en el proceso.

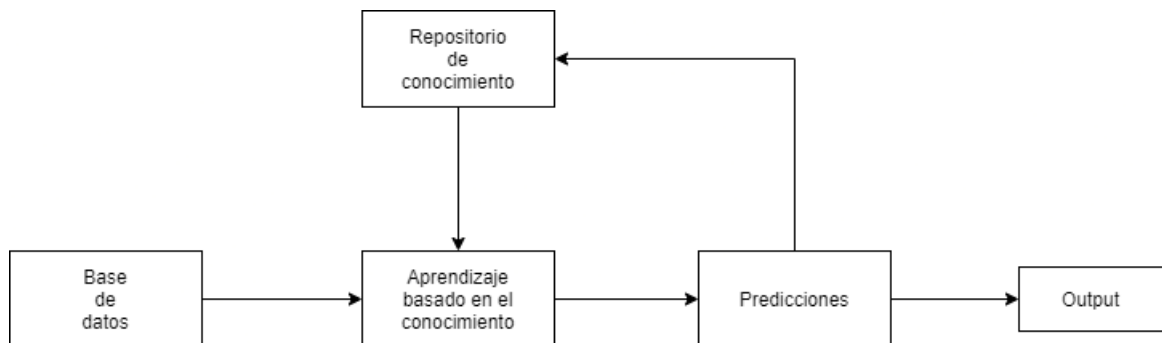


Figura 2.4.1: Proceso de Aprendizaje Automático.

## 2.5. Redes Neuronales

El cerebro es un dispositivo de procesamiento de información muy inteligente y eficiente. Tanto así que supera incluso los más recientes productos ingenieriles en casi todos los dominios. Parece lógico, entonces, mirar la arquitectura del cerebro en busca de inspiración sobre cómo construir una máquina inteligente.

En aprendizaje automático, las redes neuronales (ANN) se inspiran en las redes de neuronas biológicas que se encuentran en el cerebro. Al aplicar algoritmos que imitan los procesos de las neuronas reales, se puede hacer que la red *aprenda* a resolver muchos tipos de problemas. Las ANN son el núcleo del Deep Learning. Son versátiles, potentes y escalables, haciéndolas ideales para abordar problemas de aprendizaje automático grandes y complejos, tales como clasificar miles de millones de imágenes, potenciar el reconocimiento de voz servicios o aprendiendo a vencer al campeón mundial en el juego de Go.

A modo de entender cómo funcionan las ANN, en la figura 2.5 se observa una representación gráfica de la arquitectura más simple de una red neuronal artificial, una que contine solo una neurona. Las entradas y la salida son números y cada conexión de entrada está asociado con un peso. La unidad calcula una suma ponderada de sus entradas ( $z = w_1 \cdot x_1 + w_2 \cdot x_2 + \dots + w_n \cdot x_n = x^T \cdot w$ ), luego aplica una función de activación a esa suma (función escalón en este caso) y genera el resultado:  $h_w(x) = \phi(z) = \text{step}(z)$ , donde  $z = x^T w$ .

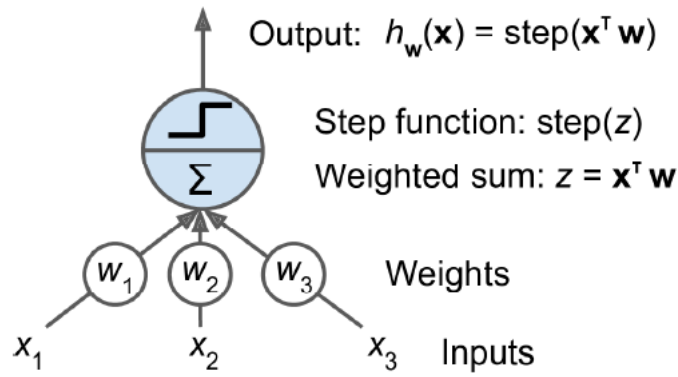


Figura 2.5.1: Una neurona artificial que calcula una suma ponderada de sus entradas y luego aplica una función escalón. [4]

La arquitectura de la figura 2.5.1, contempla una capa con una neurona, la cual está conectada a todas las entradas. Cuando todas las neuronas en una capa están conectadas a cada neurona en la capa anterior, a la capa se le conoce como una *fully connected layer* o *dense layer*.

Entonces, una ANN se basa en una colección de neuronas conectadas, donde cada conexión puede transmitir una señal a otras neuronas. Una neurona artificial que recibe una señal la procesa y puede señalar a las neuronas conectadas a ella. Normalmente, las neuronas se agregan en capas, y cuando una ANN contiene una pila profunda de capas ocultas, se denomina red neuronal profunda (DNN). En la figura 2.6 se observa una arquitectura típica de una red neuronal profunda.

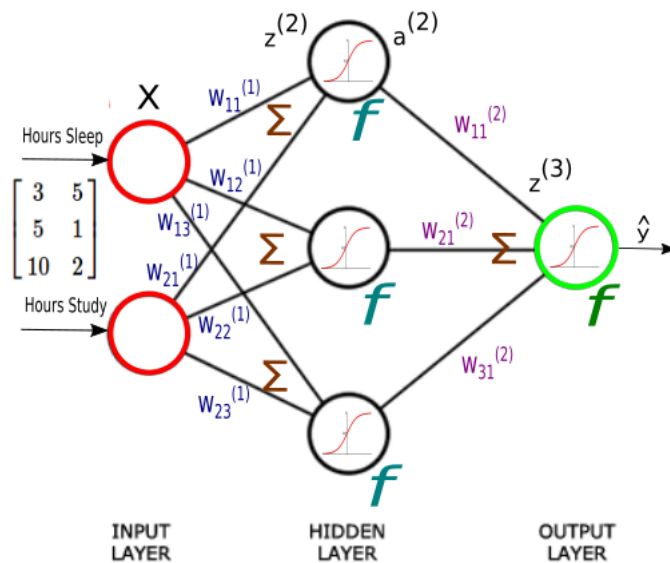


Figura 2.5.2: Arquitectura de una red neuronal. [5]

Durante muchos años, investigadores buscaron una manera de entrenar redes neuronales. Sin embargo, el éxito llegó recién en el año 1986 [17], con la publicación de una revolucionaria investigación que introducía el algoritmo de entrenamiento de *backpropagation*. Básicamente, éste consiste en un descenso de gradiente, utilizando una técnica eficiente para calcular los gradientes automáticamente. Entonces, para cada instancia de entrenamiento, el algoritmo de backpropagation primero hace una predicción (forward pass) y mide el error. Luego, pasa por cada capa en reversa para medir la contribución de error de cada conexión (reverso pass), y finalmente ajusta los pesos de conexión para reducir el error (Gradient descent pass). Una vez que tiene estos gradientes, simplemente realiza un paso de descenso de gradiente regular, y todo el proceso se repite hasta que la red converge a la solución.

### 2.5.1. Tipos de redes neuronales

Tal como se describía anteriormente, una red neuronal fully connected, consta de una serie de capas completamente conectadas que conectan cada neurona en una capa con cada neurona en la otra capa. La principal ventaja de las redes fully connected es que son *agnósticas de la estructura*, es decir, no es necesario realizar suposiciones especiales sobre la entrada, lo que hace que este tipo de redes se apliquen de manera muy amplia. Sin embargo, estas no son las únicas redes que existen.

Existe otro tipo de redes llamadas Redes Neuronales Convolucionales (CNN), las que se utilizan principalmente en el reconocimiento de imágenes. Las arquitecturas de CNN asumen explícitamente que las entradas son imágenes [18], lo que permite codificar ciertas propiedades en la arquitectura del modelo. Una CNN simple es una secuencia de capas y cada capa de una CNN transforma un volumen de activaciones en otro a través de una función diferenciable. Se utilizan tres tipos principales de capas para construir la arquitectura CNN: capa convolucional, capa pooling y capa fully connected. Esta combinación de capas hace que el input entregado a las capas siguientes se haga cada vez más pequeño, conforme se va avanzando en red, como también más profundo. Es por esto que, en comparación con las redes neuronales fully connected, el principal beneficio de usar CNN es la reducción de los parámetros que se deben aprender [19].

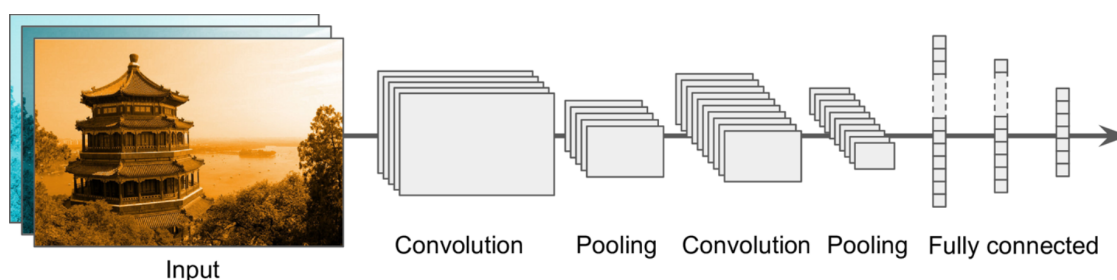


Figura 2.5.3: Arquitectura típica de una red neuronal convolucional. [4]

### 2.5.2. Tipos de problemas

La relevancia de utilizar algoritmos de aprendizaje de máquinas, por sobre otras técnicas tradicionales, radica en obtener solución a problemas que se consideran complejos ya sea por la gran cantidad de datos a analizar, la gran cantidad de ajustes o la gran cantidad de

requisitos que tienen en la entrada. Algunas de los problemas a resolver más comunes para este tipo de algoritmos son:

## Clasificación

Un problema de clasificación se produce cuando un objeto debe asignarse a alguno de los grupos (clases) que fueron generados con anterioridad en función de un conjunto de atributos observados [20]. Es por esto que, en este tipo de problemas, los datos que son alimentados a la red neuronal, están divididos en clases. Es decir, se utiliza un dataset previamente etiquetado. Un ejemplo común de clasificación, es determinar el estado de salud de componentes mecánicos como engranajes, cuyas clases serían *saludable* y *dañado*.

Los problemas de clasificación pueden ser binarios o multiclase. Tal como indica su nombre, la clasificación binaria es una tarea de clasificación entre 2 clases y, la clasificación multiclase, una tarea de clasificación de más de 2 clases. Entonces, los modelos de redes neuronales difieren en la cantidad de salidas que tienen, como se muestra en la figura 2.5.4.

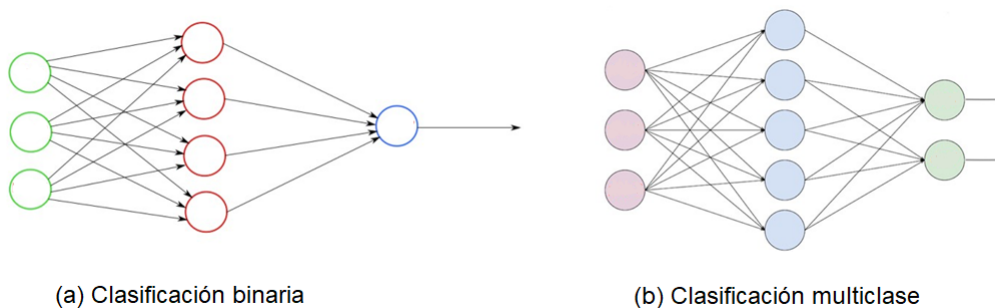


Figura 2.5.4: En la clasificación binaria, la red neuronal tiene solo una salida para predecir 2 clases. En cambio, en la clasificación multiclase, se puede tener N salidas. [6]

## Regresión

Otro problema típico a resolver es la predicción de un valor numérico objetivo, como el precio de una casa, dado un set de características llamadas predictores. Es decir, se centra en la relación entre una variable dependiente y una o más variables independientes [20]. A esto se le llama regresión. En este tipo de modelos, la red neuronal tendrá un nodo de salida para cada valor continuo que se quiera predecir, como se muestra en la figura 2.5.5.

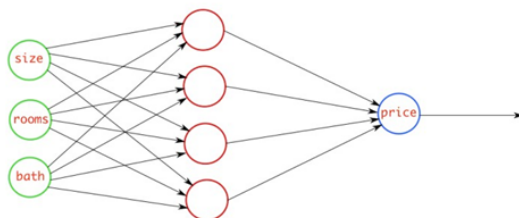


Figura 2.5.5: Modelo de regresión para la predicción del precio de casas. [6]

### 2.5.3. Entrenamiento y pruebas

Una de las cosas que los humanos hacen seguido es sobregeneralizar. Como por ejemplo, cuando compran un producto y a los pocos días deja de funcionar, tienden a pensar que todos los productos de esa marca en específica son malos. Lamentablemente, las máquinas pueden caer en lo mismo si no se tiene cuidado, produciéndose lo que se llama *sobreajuste*.

El sobreajuste ocurre cuando se entrena un modelo y se evalúa su desempeño con los mismos datos. El modelo simplemente repite las etiquetas de las muestras que acaba de ver, obteniendo una puntuación perfecta, pero falla en predecir datos no vistos con anterioridad. Para evitar este problema, una práctica común es separar los datos disponibles en un conjunto para entrenar (conjunto de entrenamiento) y otro para evaluar (conjunto de testeo).

Siguiendo lo anterior, evaluar un modelo sería bastante simple: utilizar un conjunto de testeo. Sin embargo, al evaluar diferentes hiperparámetros para los modelos, aún existe el riesgo de que el conjunto de testeo se sobreajuste. Esto sucede porque los parámetros se pueden seleccionar hasta que el estimador funcione bien para ese set de datos. Una solución a esto es la validación de exclusión: retener una parte del conjunto de entrenamiento para evaluar varios modelos candidatos y seleccionar el mejor. Este nuevo conjunto se denomina conjunto de validación.

Entonces, el modelo se entrena con los datos de entrenamiento y se realizan pruebas con distintas configuraciones evaluando en el set de validación. Cuando se encuentra la mejor configuración, se hace la evaluación final en el conjunto de prueba.

### 2.5.4. Métodos de regularización

Las redes neuronales profundas generalmente tienen de miles a millones de parámetros. Esto hace que tengan bastante libertad para poder adaptarse a una gran variedad de datasets distintos. Sin embargo, esto hace que la red esté susceptible a sobreajustar el conjunto de entrenamiento. Es por esto que se necesita una técnica llamada regularización.

La palabra regularizar significa hacer que las cosas sean regulares o aceptables. La regularización es el término estadístico general para una operación matemática que limita la memorización al mismo tiempo que promueve el aprendizaje generalizable. Las regularizaciones son técnicas que se utilizan para reducir el error al ajustar una función de forma adecuada en el conjunto de entrenamiento dado y evitar el sobreajuste.

#### Dropout

Dropout es una forma de regularización que elimina aleatoriamente una parte de las neuronas que ingresan a una capa fully connected. En palabras simples: en cada paso del entrenamiento, cada neurona (incluida la neuronas de entrada, pero excluyendo las neuronas de salida) tiene una probabilidad  $p$  de ser abandonada temporalmente, lo que significa que se ignorará por completo durante este paso del entrenamiento, pero puede estar activa durante el siguiente paso. A este parámetro  $p$  se le llama *dropout rate*.

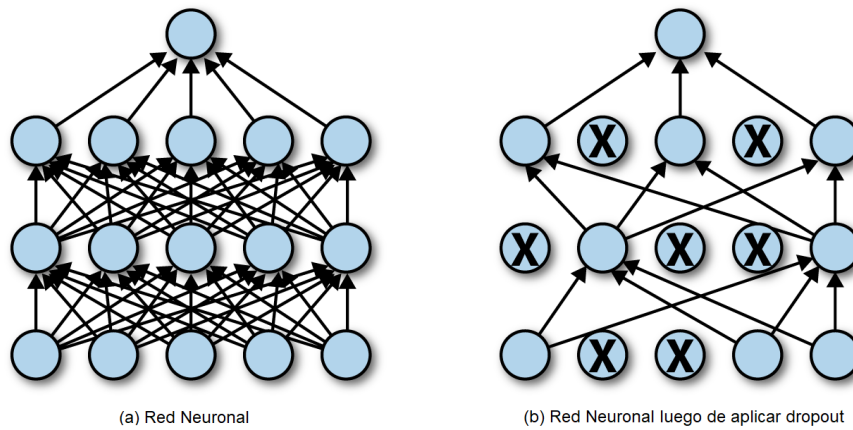


Figura 2.5.6: Representación gráfica del método de dropout sobre una red neuronal estándar. [7]

La finalidad de este método de regularización es evitar la *co-adaptación*. La co-adaptación surge cuando una neurona de una red profunda ha aprendido una representación útil, haciendo que otras neuronas más profundas de la red aprendan rápidamente a depender de esa neurona en particular para obtener información. Este proceso hace que la red sea frágil, ya que la red dependerá excesivamente de las características aprendidas por esa neurona.

### Early Stopping

Tal como se mencionó anteriormente, las redes neuronales tienden a memorizar todo lo que se les presenta ante ellas. Entonces, es útil realizar un seguimiento del rendimiento de la red en un conjunto de validación y detener la red cuando el rendimiento en este conjunto de validación comience a disminuir. A esto se le llama *early stopping*.

Para entender mejor este método, se observa lo que sucede en la figura 2.5.7. La figura muestra la variación del error de predicción (RMSE) sobre un modelo de redes neuronales, a medida en que pasan las épocas. Se puede ver que, después de un tiempo, el error sobre el conjunto de validación deja de disminuir y comienza a subir. Esto indica que el modelo comienza a sobreajustarse a los datos del conjunto de entrenamiento. Con *early stopping*, solo se detienen el entrenamiento tan pronto el error en el conjunto de validación alcanza el mínimo.

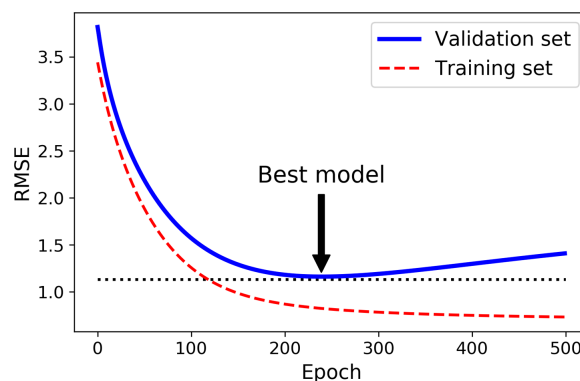


Figura 2.5.7: Regularización por early stopping. [4]

## 2.5.5. Preprocesamiento de los datos

Tal como se habló anteriormente, los modelos de redes neuronales aprenden a relacionar variables de entrada con variables de salida, pero puede ocurrir que la escala y distribución de los datos utilizados sean diferentes para cada variable. Es decir, se podrían tener diferentes unidades para las variables de entrada (metros, kilos y horas, por ejemplo), lo que significa que las variables tienen diferentes escalas. Esto deriva en un aumento en la dificultad del problema que se modela y, eventualmente, resultados deficientes.

Para evitar lo anterior, se utiliza el método de **normalización mínima-máxima**, previo al entrenamiento de los modelos y así obtener mejores resultados. Este método crea versiones escaladas y desplazadas de los conjuntos de datos, para realizar una comparación más cuidadosa de las variables. La normalización es un cambio de escala de los datos del rango original para que todos los valores estén dentro del rango de 0 y 1. Por lo mismo, requiere que se pueda ser capaz de estimar con precisión los valores observables mínimo y máximo. Entonces, un valor es normalizado como sigue [21]:

$$x' = \frac{x - \min(x)}{\max(x) - \min(X)} \quad (2.1)$$

## 2.5.6. Selección de hiperparámetros

Los hiperparámetros son las variables que determinan la estructura de una red neuronal y las variables que determinan cómo se entrena esta red. Son parámetros que no se aprenden directamente dentro de los modelos y que se deben especificar antes del entrenamiento.

Dentro de los hiperparámetros **relacionados a la estructura de la red** se encuentran: número de capas ocultas, número de neuronas por capa, tasa de dropout y función de activación. Mientras que, como hiperparámetros **relacionados al entrenamiento del modelo**, se tienen: tasa de aprendizaje, número de épocas, batch size.

El proceso de selección de hiperparámetros puede llegar a ser un poco tedioso por la cantidad de combinaciones posibles a evaluar. Es por ello que existen estrategias para automatizar este proceso como lo son:

- **Grid Search:** se evalúan todas las combinaciones de parámetros dentro de ciertos rangos.
- **Randomized Search:** se evalúan un número fijo de combinaciones seleccionadas aleatoriamente.

## 2.5.7. Métricas de evaluación

En el área del aprendizaje automático, existen varias métricas para la evaluación del rendimiento de los algoritmos. En particular, para los modelos que resuelven problemas de regresión, se utiliza el **error cuadrado medio** y el **error absoluto medio** como función de pérdida [22], el **coeficiente de determinación**  $R^2$  para cuantificar el desempeño en la predicción de las etiquetas.

El coeficiente de determinación  $R^2$ , el cual determina la calidad del modelo para replicar los resultados y la proporción de variación de los resultados que puede explicarse por el modelo [23] [24]. Un modelo será excelente mientras más cercano a 1 sea su  $R^2$ . Este coeficiente se calcula como [25]:

$$R^2 = \frac{\sum_i (\hat{y}_i - \bar{y})^2}{\sum_i (\bar{y}_i - \hat{y})^2} \quad (2.2)$$



# Capítulo 3

## Formulación

En este capítulo se presentarán los lineamientos necesarios para la generación de los conjuntos de datos que alimentan los modelos de redes neuronales. Se comienza por presentar las estructuras en estudio y los parámetros de diseño definidos para este trabajo. Luego, se aborda la obtención de los diagramas de banda y el cálculo de las características del band-gap que se busca predecir: ancho de banda y frecuencia central media.

### 3.1. Geometrías

El creciente interés en el estudio de los metamateriales es la inspiración detrás del análisis de distintos núcleos de paneles tipo sándwich. Por lo mismo, es que el presente trabajo aborda el estudio de 3 tipos de celdas unitarias para este tipo de estructuras, cuyas diferencias radican en los parámetros de diseño definidos para ellas. Los materiales empleados para las siguientes estructuras corresponden a aluminio y tungsteno, descritos en la tabla 3.1, debido a su alto contraste en las propiedades mecánicas.

Tabla 3.1: Propiedades mecánicas de los materiales utilizados en cada estructura en estudio.

Propiedad	Aluminio 6061 (Material 1)	Tungsteno (Material 2)
Módulo de elasticidad	68.9[GPa]	411[GPa]
Densidad	2700[kg/m <sup>3</sup> ]	19300[kg/m <sup>3</sup> ]
Coefficiente de Poisson	0.33	0.28

#### 3.1.1. Estructura 1

La primera estructura proviene del trabajo de título desarrollado por L.Quinteros [16]. Corresponde a un enrejado que posee el mismo número de celdas  $n_c$  tanto en  $x$  como en  $y$ , el que se muestra en la figura 3.1.1. En este caso, se puede observar que tiene un número de celdas igual a 4. El largo de la celda estará determinado por  $n_c$  y será calculado con la expresión  $L_c = 0,025n_c[m]$ .

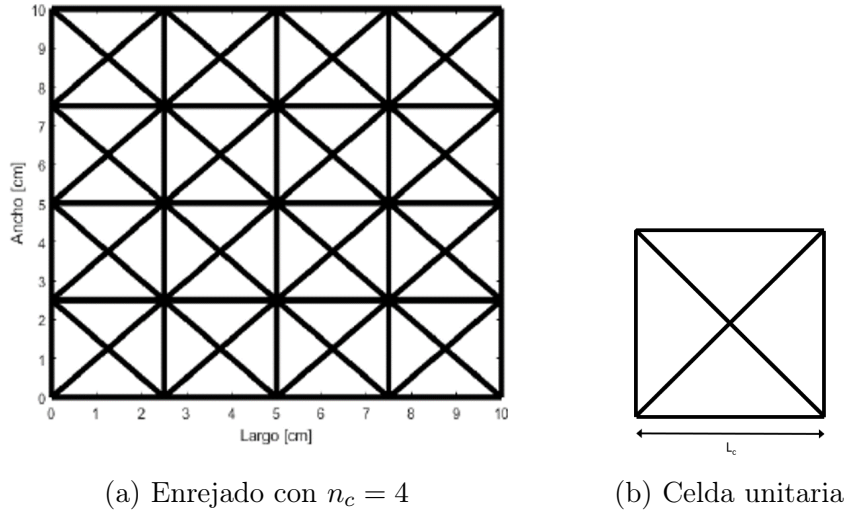


Figura 3.1.1: Estructura 1.

Esta estructura tiene 2 variables de decisión por cada elemento, siendo una de ellas la que controla el material y la otra la sección transversal, teniéndose un vector con 24 variables de entrada para los modelos de redes neuronales. Al tratarse de un elemento de barra, para el cálculo de la sección transversal solo interesa su área, la que se define como una sección circular con diámetro  $4[mm]$  en la sección más angosta y  $8[mm]$  en la más ancha.

### 3.1.2. Estructura 2

El diseño elegido para la estructura 2 está conformado por una matriz y resonadores internos, la que también proviene de un trabajo de título [26] relacionado al área de metamateriales. Esta celda unitaria cuenta con una geometría cuadrada con vigas en voladizo en su interior, obteniéndose un sistema tipo voladizo en masa. El esquema de la celda unitaria se presenta en la figura 3.1.2. Los parámetros de diseño para este caso corresponden al largo de celda  $L$ , largo de barras  $L_v$ , material de barras, radios de barras y las masas puntuales, teniéndose un vector de 15 variables de entrada para los modelos de redes neuronales.

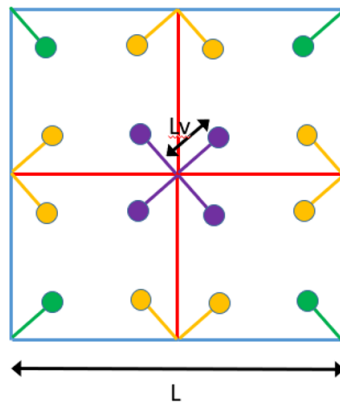


Figura 3.1.2: Estructura 2: Sistema tipo voladizo en masa

En la figura, además, se observa que la celda unitaria se descompone en otras cuatro celdas interiores, cada una formada por un marco cuadrado de vigas unidas rígidamente, y por un conjunto de 4 voladizos que se encuentran empotrados en las esquinas del marco.

### 3.1.3. Estructura 3

La tercera estructura en estudio es una aplicación entre las primeras 2 estructuras antes descritas. El diseño de esta estructura consiste en una matriz enrejada con resonadores internos. El esquema de la celda unitaria se presenta en la figura [3.1.3](#). Los parámetros de diseño para este caso corresponden al largo de celda  $L$ , material de barras, radios de barras y las masas puntuales, teniéndose un vector de 15 variables de entrada para los modelos, al igual que el caso anterior.

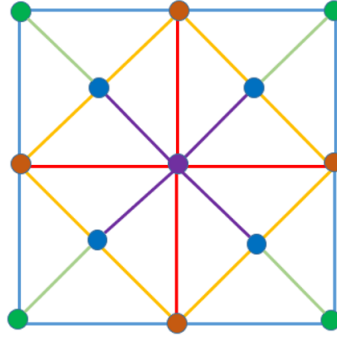


Figura 3.1.3: Estructura 3: Sistema tipo enrejado con masas suspendidas

Observando la figura anterior, se puede apreciar que ésta también se descompone en 4 celdas interiores, cada una formada por un marco cuadrado de vigas unidas rígidamente y masas concentradas en las vigas en las cuales están interconectadas.

## 3.2. Cálculo de band-gaps

Tal como se menciona anteriormente, el diagrama de bandas entrega información sobre la propagación de las ondas mecánicas sobre el material. Para entender cómo se obtiene, se observa la figura [3.2.1](#), la que muestra una estructura con periodicidad infinita que tiene simetría reflectiva con respecto a una recta en  $45^\circ$  que pasa por el centro, y a su derecha, se presenta la zona irreducible de Brillouin (ZIB), que es la primera zona de Brillouin reducida por todo el grupo de simetrías presente en red del material. Este gráfico indica el camino del vector de ondas  $\mathbf{k}$ , el que es definido a partir del perímetro del camino cerrado que se recorre alrededor de la ZIB. En éste, se puede ver que el recorrido comienza en  $\Gamma$ , pasando a  $X$ , luego  $M$  y retorna a  $\Gamma$ .

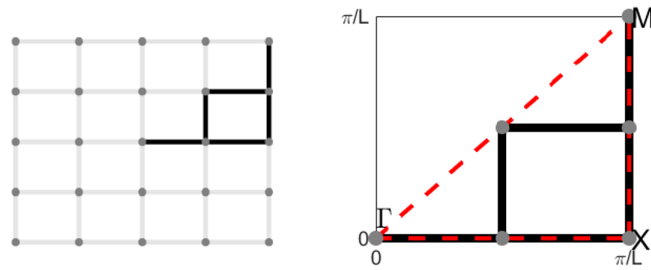


Figura 3.2.1: Ejemplo de estructura con periodicidad infinita. A la izquierda se muestra su simetría reflectiva con respecto a una recta en  $45^\circ$  que pasa por el centro. A la derecha se presenta la Zona Irreducible de Brillouin.

A partir de este vector de ondas  $\mathbf{k}$ , se obtiene una lista de valores propios, que al ser ordenada de forma ascendente, permite hacer el seguimiento de los valores propios y observar sus posiciones específicas en la lista. Se define  $w_n(k)$  como el término que indica la evolución de la frecuencia natural, asociada al valor propio ubicado en la  $n$ -ésima posición del listado. Entonces, en el diagrama de bandas se muestran las frecuencias naturales (bandas) calculadas a lo largo del vector de ondas.

El band-gap se produce cuando la diferencia entre 2 bandas de frecuencia adyacentes es positiva. La figura 3.2.2 muestra el diagrama de bandas de un material que tiene un band-gap entre las bandas 6 y 7.

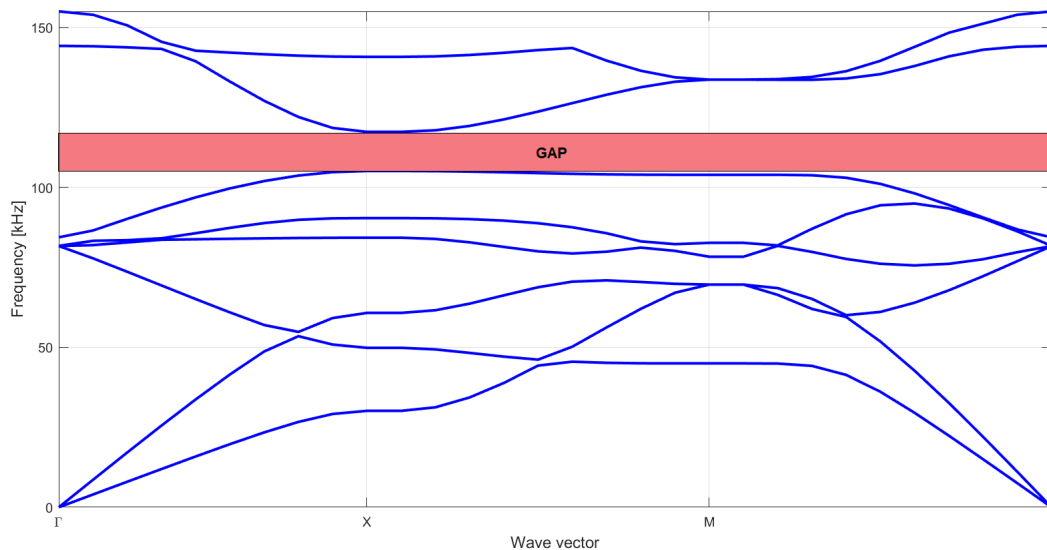


Figura 3.2.2: Diagrama de bandas con la presencia de un band-gap

Naturalmente, el ancho del band-gap sería la distancia entre ambas bandas. Generalizando, el ancho de un band-gap se calcula como la distancia entre el mínimo valor de la banda  $n + 1$  y el máximo valor de la banda  $n$ :

$$\Delta\omega = \text{Min } \omega_{n+1}(k) - \text{Max } \omega_n(k) \quad (3.2.1)$$

Siguiendo esa misma línea, la frecuencia central media del band-gap es simplemente:

$$\omega_t = \frac{\text{Min } \omega_{n+1}(k) + \text{Max } \omega_n(k)}{2} \quad (3.2.2)$$

# Capítulo 4

## Metodología

En este capítulo se abordarán las dos etapas en que se organizó el desarrollo de este trabajo. La primera de ellas guarda relación con la obtención de las bases de datos que alimentan los modelos de redes neuronales. En tanto, la segunda etapa contempla la construcción e implementación de los modelos de redes neuronales que resuelven el problema de predicción de band-gaps en paneles de metamateriales.

### 4.1. Generación de las bases de datos

Las bases de datos utilizadas en el entrenamiento de algoritmos, se generan a partir de modelos numéricos de paneles de metamateriales. Tal como se mencionó en la sección anterior, las estructuras en estudio fueron definidas en trabajos anteriores relacionados a esta área, por lo que ya se cuenta con los modelos numéricos de estos paneles. Entonces, lo que sigue es generar los diagramas de bandas para cada topología de panel, a partir de múltiples combinaciones de los parámetros de diseño. La Estructura 1 contempla 24 parámetros de diseño, por lo que se busca generar un conjunto robusto de vectores de 24 elementos. Del mismo modo, para las Estructuras 2 y 3, se generan muestras de vectores de 15 elementos.

Para construir dichos conjuntos de datos, se trabaja con dos enfoques distintos para la predicción de band-gaps: predicción de propiedades del band-gap y predicción del diagrama de bandas. Ambos enfoques buscan asociar a los parámetros de diseño de las celdas anteriores, con un valor de salida que permite, eventualmente, predecir el band-gap fonónico. Luego, los conjuntos de datos para cada topología se dividen en conjuntos de datos de entrenamiento, testeo y validación.

#### 4.1.1. Enfoque I: Predicción de propiedades del band-gap

El objetivo de este enfoque es la implementación de modelos de redes neuronales para la predicción directa del ancho y la frecuencia central media del band-gap, a partir de los parámetros de diseño definidos para cada estructura. Es decir, se tiene un problema con un determinado set de características en la entrada y dos salidas de valores continuos, por lo que se necesita un modelo de regresión para su resolución.

El input de los modelos a implementar, corresponde a una base de datos de vectores que contiene varias combinaciones de los parámetros de diseño, que permiten la presencia de al menos un band-gap las estructuras descritas. Para la obtención las posibles combinaciones de parámetros, se utiliza *Latin Hypercube Sampling* [27] como técnica de muestreo. Esta técnica consiste en evaluar puntos de manera aleatoria con una distribución uniforme al interior del dominio de diseño, con el objetivo de obtener una muestra lo suficientemente representativa.

Para la construcción de esta base de datos, se debe comenzar verificando si en cada combinación de parámetros obtenida, se produce band-gap. Esto se realiza bajo la premisa que hay band-gap cuando:

$$\text{Min } \omega_{n+1}(x) > \text{Max } \omega_n(x) \quad (4.1.1)$$

Cumpliendo lo anterior, se calcula del ancho y frecuencia media del band-gap (salidas), y se registran estos valores con su respectivo punto de diseño (entradas). Esto se realiza reiterativamente, hasta obtener una base de datos lo suficientemente robusta para alimentar el algoritmo. Para este enfoque, se trabaja con 3 bases de datos, una generada por cada estructura en estudio.

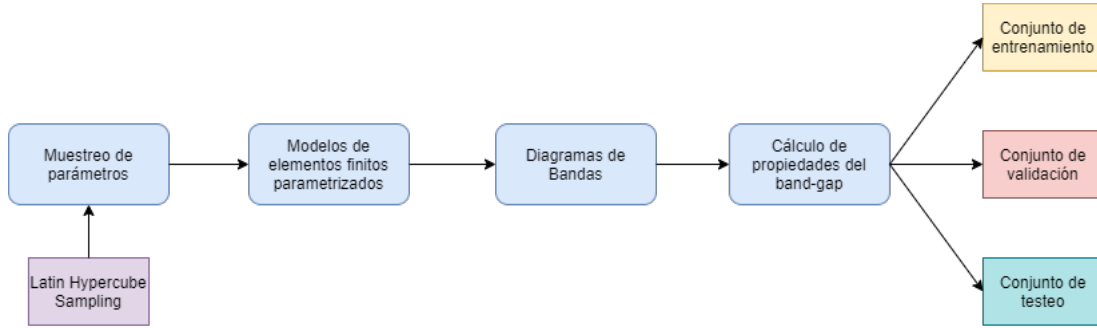


Figura 4.1.1: Diagrama de flujo para la primera etapa del Enfoque I

#### 4.1.2. Enfoque II: Predicción del diagrama de bandas

A diferencia de lo anterior, este enfoque toma un camino un poco más largo para poder predecir el band-gap del material. Este enfoque se centra en la implementación de modelos de redes neuronales que predigan los componentes principales de las bandas de frecuencia de estos diagramas, para luego reconstruirlos y así calcular las características del band-gap.

Para obtener estos conjuntos de datos, se comienza por guardar la información de las bandas de frecuencias de los modelos numéricos de los paneles en estudio. La información de estas matrices se comprime utilizando uno de los métodos de reducción de parámetros conocidos, llamado *Kernel PCA*, lo que permite que cada banda necesite solo de un componente principal para reconstruirse de forma perfecta. Entonces, cada combinación de parámetros de diseño del material, es asociada a las bandas de frecuencia de su diagrama de bandas. Este enfoque trabaja con 2 bases de datos, generadas a partir del muestreo de las estructuras 2 y 3.

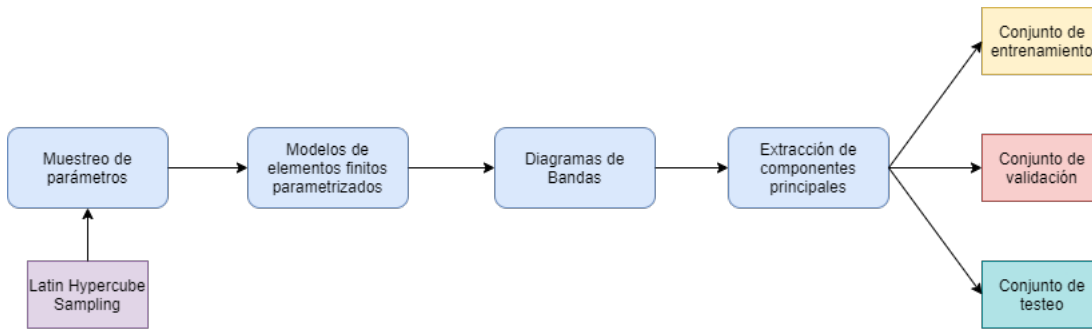


Figura 4.1.2: Diagrama de flujo para la primera etapa del Enfoque II

## 4.2. Desarrollo de modelos de redes neuronales

Esta etapa comprende la generación de las distintas arquitecturas para los modelos de redes neuronales, la cual comienza enumerando los recursos computacionales necesarios para su desarrollo. Se utiliza Python como lenguaje de programación y las librerías de código abierto Scikit-learn y Keras. Keras es una biblioteca de redes neuronales capaz de ejecutarse sobre TensorFlow y está especialmente diseñada para posibilitar la experimentación en más o menos poco tiempo con redes de aprendizaje aprofundo, siendo fácil de usar y amigable para el usuario. En relación a los requerimientos para la ejecución de códigos, se utiliza Google Colaboratory para ejecución en la nube y Spyder para la ejecución de manera local. El computador utilizado tiene un procesador Intel(R) Core(TM) i5 de 8va generación, 8GB de memoria RAM y GeForce MX 150 GPU.

Teniendo en cuenta lo anterior, se procede con la programación de los algoritmos. Se trabaja con la arquitectura clásica de una red neuronal profunda utilizada en problemas de predicción, como la que se muestra en el siguiente pseudocódigo.

---

### Algorithm 1: Modelo de Redes Neuronales Profundas

---

**Input:** Set de parámetros de diseño de la estructura en estudio

**Output:** Características del band-gap/Componentes principales de bandas de frecuencia

- 1 Cargar sets de datos
  - 2 Normalizar sets de datos
  - 3 Inicializar modelo de Keras
  - 4 Configurar capa de entrada
  - 5 Agregar capas ocultas
  - 6 Configurar capa de salida
  - 7 Compilar modelo
  - 8 Entrenar modelo
  - 9 Obtener predicciones sobre el conjunto de testeo
  - 10 Evaluar rendimiento del modelo
- 

Los modelos de predicción programados para este trabajo siguen este mismo esquema, ya que lo diferencia un modelo de otro, son las entradas y salidas, el tipo de problema que resuelven (clasificación o regresión) y los hiperparámetros elegidos. Dado que existen una



gran cantidad de combinaciones distintas para las funciones e hiperparámetros de los modelos, previo al entrenamiento se implementa una estrategia de búsqueda aleatoria (Random Search) para encontrar las mejores combinaciones. Se define que la búsqueda se realice sobre la cantidad de neuronas por capa, función de activación, tasa de Dropout y tasa de aprendizaje.

Como se mencionó anteriormente, se tienen 2 enfoques para resolver el problema original de predicción. La entrada es la misma (parámetros de diseño), pero difieren en las salidas, ya que los modelos del enfoque I predicen ancho de banda y frecuencia central media del band-gap (2 salidas), y los modelos del enfoque II predicen las bandas de frecuencia del diagrama (8 salidas). Por esta misma razón, es que para el enfoque II se tienen pasos adicionales antes de escoger el modelo de redes neuronales final. El flujo de trabajo para ambos enfoques se muestran en las figuras 4.2.1 y 4.2.2 .

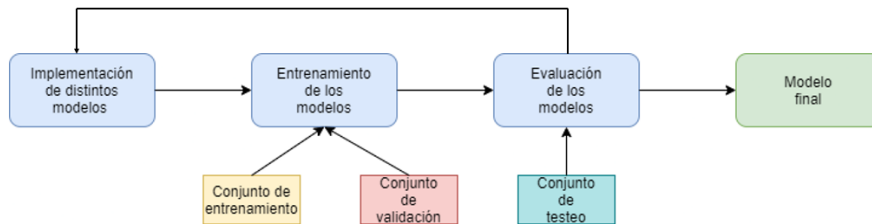


Figura 4.2.1: Diagrama de flujo para el enfoque I

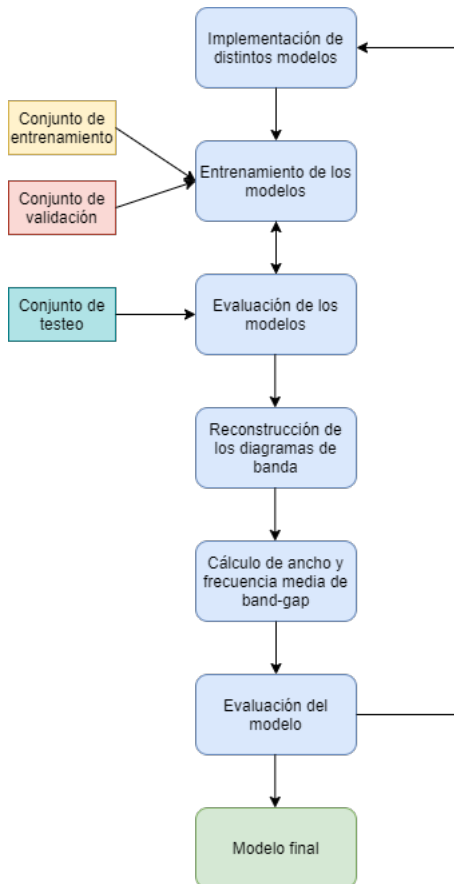


Figura 4.2.2: Diagrama de flujo para el enfoque 2

# Capítulo 5

## Resultados

En esta sección se presentan los resultados obtenidos del entrenamiento de los distintos modelos, según los enfoques presentados anteriormente y las distintas bases de datos generadas de los modelos numéricos de las estructuras. Además, se presentan las combinaciones de hiperparámetros utilizados para cada modelo y las comparaciones de desempeño entre ellos.

### 5.1. Enfoque I

En este enfoque, se programaron 2 tipos de modelos de redes neuronales para cada estructura: uno con capas fully connected y otro con capas convolucionales (CNN). Si bien, es conocido el uso de CNN para el reconocimiento de imágenes, también se puede aplicar CNN con análisis de datos de regresión. Para cada uno de ellos, se realizó una búsqueda aleatoria de hiperparámetros previa al entrenamiento, definiendo neuronas por capa, función de activación, tasa de Dropout y tasa de aprendizaje. Además, del uso de Dropout como técnica de regularización, se agrega ruido Gaussiano a la capa de entrada y Early Stopping en el entrenamiento. La arquitectura de los modelos se presenta en la sección de Anexos de este documento.

Los valores de entrada a los modelos son los parámetros de diseño de las estructuras, mientras que las salidas corresponden a las 2 propiedades del band-gap: ancho de banda y frecuencia media. A continuación, se muestran los resultados del desempeño para la predicción del ancho y frecuencia media del band-gap de cada estructura.

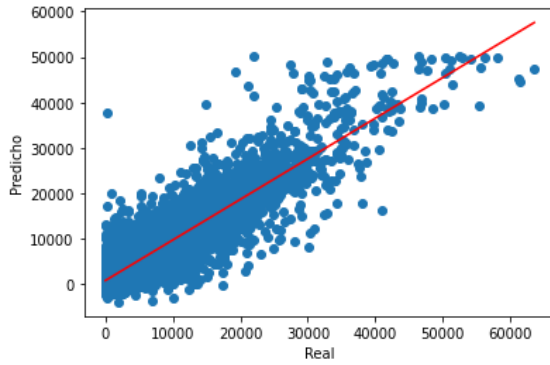
#### 5.1.1. Estructura 1

La base de datos obtenida a partir de la Estructura 1, contiene 26943 muestras, de las cuales se tiene 16165 en el conjunto de entrenamiento y 5389 los conjuntos de validación y testeo. Estos set de datos son los mismos en cada modelo, esto para hacer comparable el desempeño entre ellos. Además, se tienen 24 entradas para los modelos, de los cuales 12 corresponden a parámetros del material (binarios) y los otros 12 corresponden a parámetros de área, un arreglo de valores continuos entre 0 y 1.

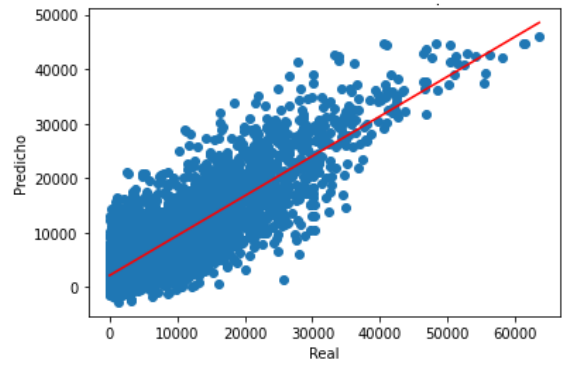
Se registra un tiempo de entrenamiento de 5 minutos para el modelo Fully Connected y de 10 minutos para el modelo con capas convolucionales (CNN).

Tabla 5.1: Registro de desempeño de los modelos para la predicción del ancho de banda del band-gap en Estructura 1.

Modelo	R2 promedio	Desviación Estándar
Fully Connected	0.786	0.0013
CNN	0.723	0.008



(a) Modelo Fully Connected.

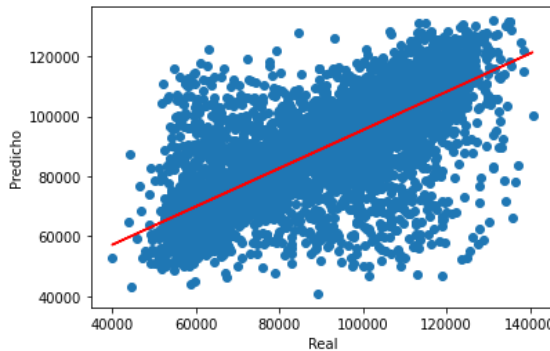


(b) Modelo CNN.

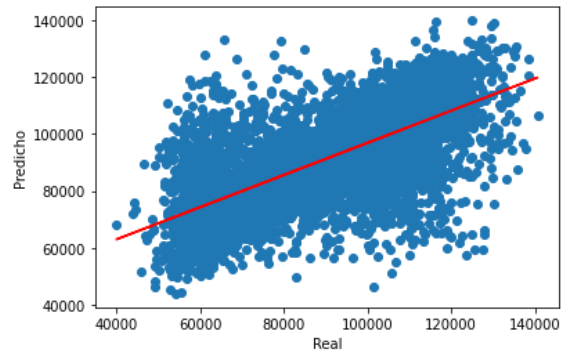
Figura 5.1.1: Comparación entre el ancho de banda real y el predicho, según los modelos entrenados con la Estructura 1.

Tabla 5.2: Registro de desempeño de los modelos para la predicción de la frecuencia media del band-gap en Estructura 1.

Modelo	R2 promedio	Desviación Estándar
Fully Connected	0.508	0.014
CNN	0.436	0.012



(a) Modelo Fully Connected.



(b) Modelo CNN.

Figura 5.1.2: Comparación entre la frecuencia media real y la predicha, según los modelos entrenados con la Estructura 1.

Tabla 5.3: Hiperparámetros del Modelo Fully Connected para la Estructura 1.

<b>Modelo Fully Connected</b>	
<b>Hiperparámetro</b>	<b>Valor</b>
No. capas ocultas	9
Neuronas por capa	64-64-64-128-128-128-256-256-256
Función de activación	Tangente hiperbólica
Tasa de Dropout	0.2
Tasa de aprendizaje	0.001
Tasa de ruido Gaussiano	0.1

Tabla 5.4: Hiperparámetros del Modelo CNN para la Estructura 1.

<b>Modelo CNN</b>	
<b>Hiperparámetro</b>	<b>Valor</b>
No. capas convolucionales	4
Filtros por capa convolucional	64-64-128-128
Función de activación	Tangente hiperbólica
No. capas densas	3
Neuronas por capa densa	96-96-96
Tasa de Dropout	0.1
Tasa de aprendizaje	0.001
Tasa de ruido Gaussiano	0.1

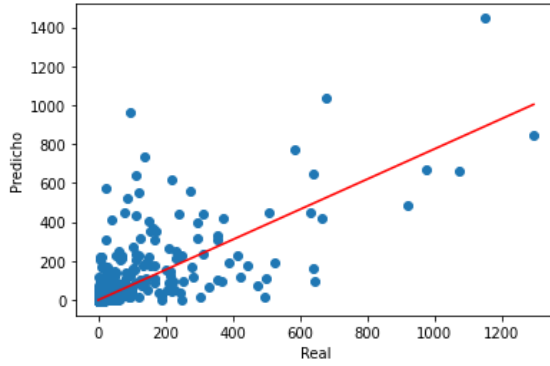
### 5.1.2. Estructura 2

La base de datos obtenida a partir de la Estructura 2, contiene 11242 muestras, de las cuales se tiene 6744 en el conjunto de entrenamiento y 2249 los conjuntos de validación y testeo. Además, se tienen 15 entradas para los modelos, de los cuales 5 son parámetros del material (binarios), 3 las masas puntuales (binarios), 5 los radios de barras y 2 correspondientes a los largos de celda  $L$  y largo de barras  $L_v$ .

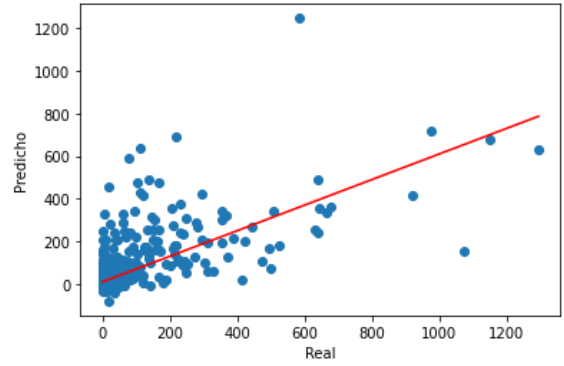
Se registra un tiempo de entrenamiento de 1 minuto para el modelo Fully Connected y de 5 minutos para el modelo con capas convolucionales (CNN).

Tabla 5.5: Registro de desempeño de los modelos para la predicción del ancho de banda del band-gap.

<b>Modelo</b>	<b>R2 promedio</b>	<b>Desviación Estándar</b>
Fully Connected	0.519	0.015
CNN	0.455	0.024



(a) Modelo Fully Connected.

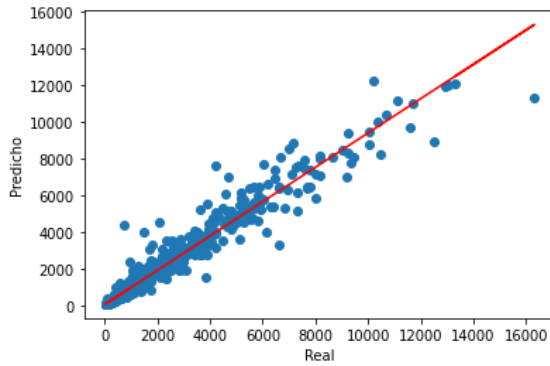


(b) Modelo CNN.

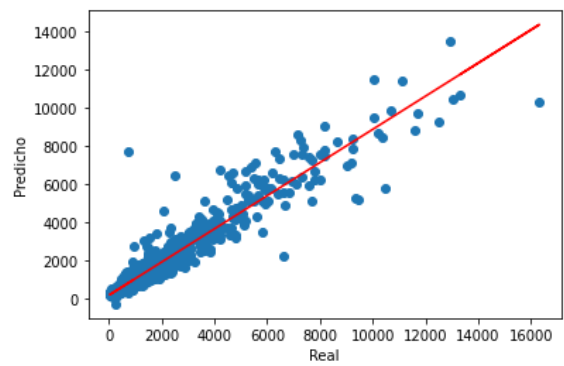
Figura 5.1.3: Comparación entre el ancho de banda real y el predicho, según los modelos entrenados con la Estructura 2.

Tabla 5.6: Registro de desempeño de los modelos para la predicción de la frecuencia media del band-gap.

Modelo	R2 promedio	Desviación Estándar
Fully Connected	0.952	0.006
CNN	0.915	0.002



(a) Modelo Fully Connected.



(b) Modelo CNN.

Figura 5.1.4: Comparación entre la frecuencia media real y la predicha, según los modelos entrenados con la Estructura 2.

Tabla 5.7: Hiperparámetros del Modelo Fully Connected para la Estructura 2.

<b>Modelo Fully Connected</b>	
<b>Hiperparámetro</b>	<b>Valor</b>
No. capas ocultas	6
Neuronas por capa	128-128-96-96-128-128
Función de activación	Sigmoide
Tasa de Dropout	0.1
Tasa de aprendizaje	0.1

Tabla 5.8: Hiperparámetros del Modelo CNN para la Estructura 2.

<b>Modelo CNN</b>	
<b>Hiperparámetro</b>	<b>Valor</b>
No. capas convolucionales	4
No. filtros por capa convolucional	96-96-96-96
Función de activación	Tangente hiperbólica
No. capas densas	2
Neuronas por capa densa	96-96
Tasa de Dropout	0.1
Tasa de aprendizaje	0.001

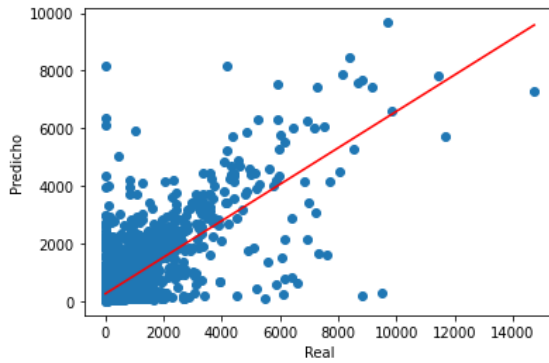
### 5.1.3. Estructura 3

La base de datos obtenida a partir de la Estructura 3, contiene 11160 muestras, de las cuales se tiene 6696 en el conjunto de entrenamiento y 2232 los conjuntos de validación y testeo. Además, se tienen 15 entradas para los modelos, de los cuales 5 son parámetros del material (binarios), 4 las masas puntuales (binarios), 5 los radios de barras y 1 correspondiente al largo de celda  $L$ .

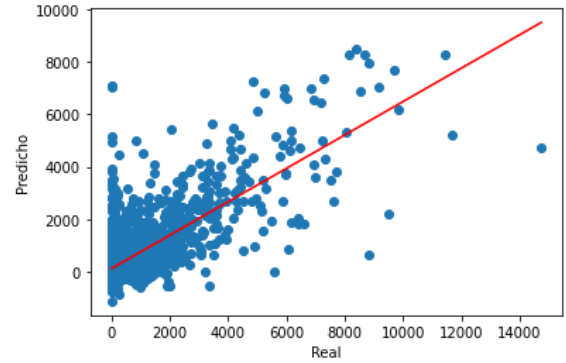
Se registra un tiempo de entrenamiento de 1 minuto para el modelo Fully Connected y de 4 minutos para el modelo con capas convolucionales (CNN).

Tabla 5.9: Registro de desempeño de los modelos para la predicción del ancho de banda del band-gap.

<b>Modelo</b>	<b>R2 promedio</b>	<b>Desviación Estándar</b>
Fully Connected	0.556	0.004
CNN	0.531	0.025



(a) Modelo Fully Connected.

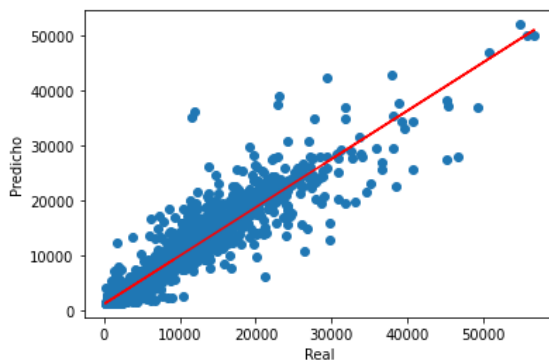


(b) Modelo CNN.

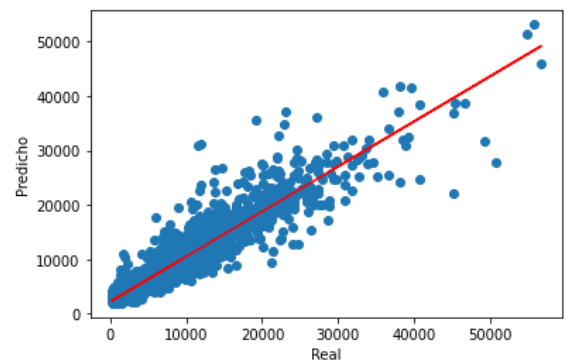
Figura 5.1.5: Comparación entre el ancho de banda real y el predicho, según los modelos entrenados con la Estructura 3.

Tabla 5.10: Registro de desempeño de los modelos para la predicción de la frecuencia media del band-gap.

Modelo	R2 promedio	Desviación Estándar
Fully Connected	0.857	0.0004
CNN	0.842	0.003



(a) Modelo Fully Connected.



(b) Modelo CNN.

Figura 5.1.6: Comparación entre la frecuencia media real y la predicha, según los modelos entrenados con la Estructura 3.

## 5.2. Enfoque II

Para este enfoque, solo se programó 1 modelo por cada estructura en estudio (Estructuras 2 y 3), el cual solo tiene capas fully connected. Se procede de este modo considerando los resultados del enfoque anterior, en donde no se presentan diferencias significativas entre el uso de un tipo de capa por sobre otra. Sin embargo, al igual que en el enfoque I, se realiza una búsqueda aleatoria de hiperparámetros previa al entrenamiento bajo las mismas definiciones anteriores, como también el uso de Dropout, ruido Gaussiano en la capa de entrada y Early Stopping como métodos de regularización.

Tabla 5.11: Hiperparámetros del Modelo Fully Connected para la Estructura 3.

<b>Modelo Fully Connected</b>	
<b>Hiperparámetro</b>	<b>Valor</b>
No. capas ocultas	6
Neuronas por capa	256-256-256-256-512-512
Función de activación	Tangente hiperbólica
Tasa de Dropout	0.1
Tasa de aprendizaje	0.0005
Tasa de ruido Gaussiano	0.01

Tabla 5.12: Hiperparámetros del Modelo CNN para la Estructura 3.

<b>Modelo CNN</b>	
<b>Hiperparámetro</b>	<b>Valor</b>
No. capas convolucionales	4
Filtros por capa convolucional	96-96-96-96
Función de activación	Tangente hiperbólica
No. capas densas	3
Neuronas por capa densa	96-96-96
Tasa de Dropout	0.1
Tasa de aprendizaje	0.001
Tasa de ruido Gaussiano	0.01

Cabe recordar que, tal como se indicó en el capítulo anterior, este enfoque mantiene los parámetros de diseño como entrada a los modelos, pero los valores de salida corresponden a los componentes principales de las 8 bandas de frecuencia del diagrama de bandas. Por lo tanto, se tiene un modelo con un total de 8 salidas, las que se utilizan para la reconstrucción de estos diagramas y calcular sus propiedades.

A continuación, se muestran los resultados del desempeño para la predicción del ancho y frecuencia media del band-gap de cada estructura, mediante el cálculo del Coeficiente de Determinación  $R^2$  y la desviación estándar.

### 5.2.1. Estructura 2

La base de datos es obtenida a partir de la Estructura 2 y contiene 11242 muestras, de las cuales se tiene 6744 en el conjunto de entrenamiento y 2249 los conjuntos de validación y testeo. Además, se mantienen los mismos 15 parámetros de diseño definidos para la estructura en el enfoque anterior. El tiempo de entrenamiento registrado para este modelo fue de 6[*min*]

Tabla 5.13: Registro de desempeño del modelo para la predicción de las propiedades del band-gap en Estructura 2

<b>Propiedad</b>	<b>R2 promedio</b>	<b>Desviación Estándar</b>
Ancho de banda	0.798	0.015
Frecuencia media	0.859	0.024



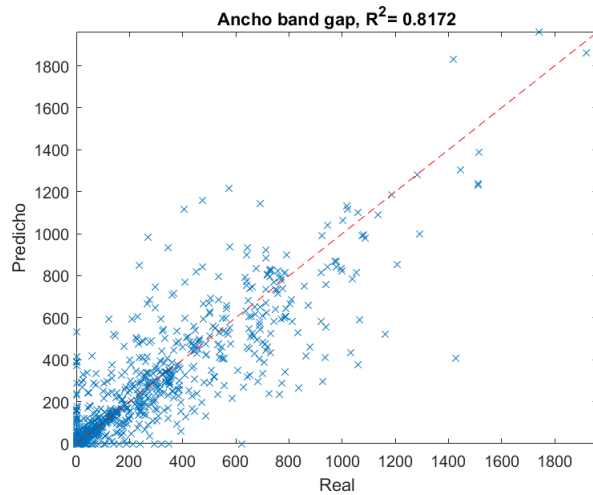


Figura 5.2.1: Comparación entre el ancho de banda real y el predicho para la Estructura 2

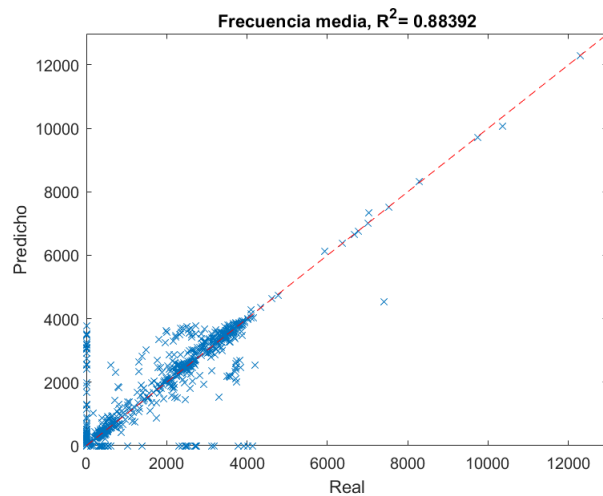


Figura 5.2.2: Comparación entre la frecuencia media real y la predicha para la Estructura 2

Tabla 5.14: Hiperparámetros para el modelo de la Estructura 2.

Hiperparámetro	Valor
No. capas ocultas	6
Neuronas por capa	290
Función de activación	Tangente hiperbólica
Tasa de Dropout	0.2
Tasa de aprendizaje	0.001
Tasa de ruido Gaussiano	0.01

### 5.2.2. Estructura 3

La base de datos es obtenida a partir de la Estructura 2 y contiene 11161 muestras, de las cuales se tiene 6696 en el conjunto de entrenamiento y 2233 los conjuntos de validación y

testeo. Además, se mantienen los mismos 15 parámetros de diseño definidos para la estructura en el enfoque anterior. El tiempo de entrenamiento registrado para este modelo fue de 4[*min*]

Tabla 5.15: Registro de desempeño del modelo para la predicción de las propiedades del band-gap en Estructura 3

Propiedad	R2 promedio	Desviación Estándar
Ancho de banda	0.871	0.00003
Frecuencia media	0.764	0.192

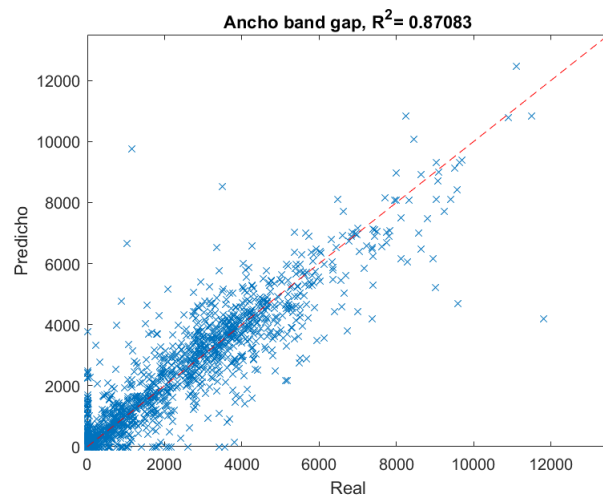


Figura 5.2.3: Comparación entre el ancho de banda real y el predicho para la Estructura 3

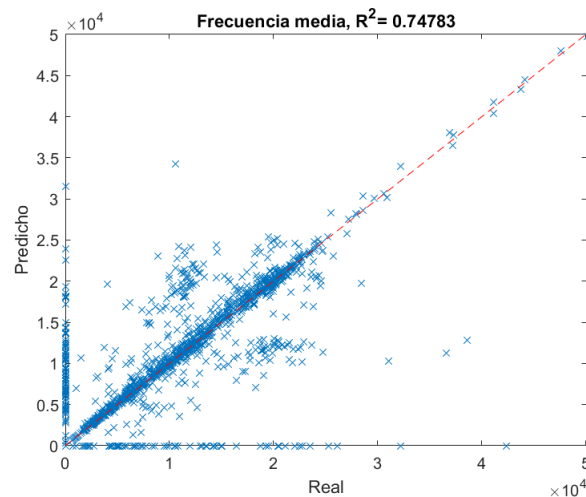


Figura 5.2.4: Comparación entre la frecuencia media real y la predicha para la Estructura 3

Tabla 5.16: Hiperparámetros para el modelo de la Estructura 3.

Hiperparámetro	Valor
No. capas ocultas	6
Neuronas por capa	219
Función de activación	Tangente hiperbólica
Tasa de Dropout	0.05
Tasa de aprendizaje	0.0005
Tasa de ruido Gaussiano	0.01

# Capítulo 6

## Discusión y Análisis

### 6.1. Enfoque I

Los resultados obtenidos dan cuenta de la capacidad de los modelos para predecir las propiedades del band-gap. En una mirada general, se observan diferencias significativas en la predicción de una propiedad respecto de la otra, como también la poca variación en el desempeño de los modelos Fully Connected y CNN.

Para la estructura 1, se observa de las figuras [5.1.2a](#) y [5.1.2b](#) que ambos modelos no logran ajustarse bien en la predicción de la frecuencia media del band-gap, ya que si bien las muestras predichas siguen la tendencia deseada, se alejan bastante de la línea de la regresión (recta en rojo). De hecho, se puede apreciar que estos modelos predicen varios valores bajos para la frecuencia media como valores altos, y viceversa. Se sabe que cuanto más cercano sea el valor de  $R^2$  a 1, mejor se ajustará el modelo. Por lo que es consistente el registro de un  $R^2$  promedio de 0,508 para el modelo Fully Connected y de 0,436 para el modelo CNN, valores que indican que los modelos solo logran explicar el 50,8 % y 43,6 % de la varianza de los datos, respectivamente. Por otra parte, se tiene que ambos modelos se ajustan bastante mejor en la predicción de ancho de banda, mostrando que se puede explicar al menos el 78 % de la varianza de los datos, si se utiliza el modelo Fully Connected. Esta mejora en el ajuste, se refleja en las figuras [5.1.1a](#) y [5.1.1b](#), donde se observan que las muestras predichas se concentran más sobre la recta de la regresión.

En la relación a la estructura 2, se tiene el caso contrario. Los modelos se ajustan mejor en la predicción de la frecuencia media que la predicción del ancho de banda. De hecho, se tiene un  $R^2$  de 0,952 para el modelo Fully Connected y de 0,915 para el modelo CNN, en la predicción de la frecuencia media, valores que son bastante más altos que el caso anterior. Este buen desempeño se muestra en las figuras [5.1.4a](#) y [5.1.4b](#), donde se observa que casi la totalidad de las muestras se alojan sobre la línea de la regresión. Sin embargo, el desempeño de ambos modelos en la predicción del ancho de banda es muy bajo, y se presenta una brecha más grande en la predicción de propiedades del band-gap con respecto al estructura anterior. En el caso más favorable, solo se logra una correlación de aproximadamente 52 % entre las variables.

Por último, para la estructura 3 también se registra un mejor ajuste de los modelos en la predicción de la frecuencia central, que en la predicción del ancho de banda. Solo que a diferencia del caso anterior, la brecha del  $R^2$  entre las predicciones disminuye. Se registra un aumento en el  $R^2$  para el ancho de banda, llegando a 0,556 con el modelo Fully Connected, y a 0,531 con el modelo CNN. Sin embargo, observando las figuras [5.1.5a](#) y [5.1.5b](#), se tiene que ambos modelos fallan en predecir valores pequeños ( $\sim 0$ ) del ancho de banda. Luego, con respecto a la predicción de la frecuencia media, se observa nuevamente que los datos se concentran bastante bien sobre la línea de la regresión. Pero, si se compara con la estructura 2, hay una disminución significativa en el  $R^2$ , llegando al 0,842 en el modelo CNN y 0,857 en el modelo Fully Connected.

Finalmente, en relación al tipo de capa utilizada para cada modelo, ya se mencionaba que no existen diferencias significativas en la predicción del band-gap. El desempeño del modelo Fully Connected fue levemente mejor sobre todas las estructuras, siendo mayor esta diferencia en los modelos entrenados con la Estructura 1. Esta diferencia en desempeño puede explicarse en la manera que los modelos interpretan los datos en la entrada [\[18\]](#). Las capas CNN asumen que lo que se les está entregando en la entrada son imágenes. En cambio, las capas fully connected son *estructuralmente agnósticas*, es decir, no realizan suposiciones de lo que se les provee en la entrada, por lo que se explica que sean suficientes y mejores para poder resolver este problema de predicción. Otro punto a favor del uso de capas fully connected, es el tiempo de entrenamiento medido, el cual no superó los 5 minutos para el caso donde se tenían más datos (Estructura 1). Esto se tiene porque estas capas son más paralelizables, es decir, evalúan al mismo tiempo 1 capa entera (con todas sus neuronas) para traspasarla a la capa siguiente.

## 6.2. Enfoque II

Este enfoque se trabajó solo utilizando bases de datos generadas a partir de las estructuras 2 y 3, por lo que se deja fuera de estos análisis y comparaciones, el desempeño obtenido anteriormente para la estructura 1. Además, siguiendo con lo presentado en la sección anterior, solo se implementaron modelos con capas fully connected.

Al igual que en el enfoque anterior, el modelo entrenado con la Estructura 2 es superior en la predicción de la frecuencia media, pero se obtienen mejores resultados a nivel general. Se registran  $R^2$  promedio de 0,798 y 0,859 para el ancho de banda y frecuencia media, respectivamente. Si bien, es bastante menor la precisión con que este modelo predice la frecuencia media en comparación con el Enfoque I, logra equilibrar la predicción de ambas propiedades del band-gap. Aún así, la gran debilidad de este modelo es que no logra predecir bien los valores pequeños ( $\sim 0$ ) para la frecuencia media.

Para el modelo entrenado con la Estructura 3, se tiene el caso contrario. La predicción del ancho de banda es bastante superior que para la frecuencia media, con valores de  $R^2$  promedio de 0,871 y 0,764, respectivamente. Además, se registra que la brecha entre las predicciones de las propiedades aumenta con respecto al modelo de la Estructura 2. Observando la [5.2.4](#), se tiene que para este modelo es difícil predecir valores pequeños para la frecuencia media, como también predice como 0 varios valores distintos de 0 para esta propiedad. Si se realiza la comparación con el enfoque anterior, es evidente que este modelo es mejor en la predicción,

ya que es más equilibrado en la predicción de ambas propiedades.

Los resultados obtenidos con este enfoque dan cuenta que hay menor pérdida de información en predecir las bandas de los diagramas, que en el cálculo de las propiedades de éstos. Sin embargo, es interesante el hecho que a pesar que el Enfoque II supone un problema más complejo que el Enfoque I, debido a que se cuenta con más información en la salida de los modelos, este segundo enfoque logra mejores resultados sobre la red neuronal. Por lo que, se puede decir que existe una función (que no es continua ni suave), que mapea uno por uno los vectores que describen la estructura con su respectivo diagrama de bandas, y que en algún punto, la red neuronal implementada pudo aproximar dicha función.

# Capítulo 7

## Conclusiones

En el presente trabajo se ha logrado de manera favorable el entrenamiento de modelos de redes neuronales para predecir band-gaps en paneles de metamateriales, lo que constituye algo potencialmente útil para el diseño de estructuras que supriman las vibraciones mecánicas debido a excitaciones externas. Los tópicos tratados en detalle en el marco teórico, proporcionan los fundamentos que permiten entender el problema de predicción y cálculo de las propiedades del band-gap.

En este estudio, se generaron 3 bases de datos para entrenar los modelos de redes neuronales, a partir de celdas: tipo enrejado, sistema tipo voladizo en masa y una estructura que combina las 2 celdas anteriores. Utilizando un enfoque directo para la predicción de propiedades del band-gap, implementan modelos que contienen capas fully connected y modelos con capas convolucionales. Los desempeños medidos en el entrenamiento de los modelos, se evaluaron utilizando el coeficiente de determinación  $R^2$ , mostrando que no existen diferencias muy significativas en el uso de ambos tipos de capas. En cambio, en lo que sí se registran diferencias importantes, es en la predicción de propiedades en las distintas estructuras, siendo más favorable la predicción de band-gaps en las estructuras que poseen resonadores internos en su diseño.

Un segundo enfoque fue utilizado para evaluar el desempeño de redes neuronales en la predicción de band-gaps. Este enfoque consistía en predecir las bandas de frecuencia del diagrama de bandas, para posteriormente calcular el ancho y frecuencia media del band-gap. Se registran mejores resultados para este enfoque, ya que se logra disminuir la brecha en la predicción de ambas propiedades, algo que da luces de una menor pérdida de la información al seguir esta metodología para el problema. Por último, se concluye el modelo de redes neuronales se ajusta mejor sobre la base de datos de la Estructura 2.

### 7.1. Trabajos Futuros

Este trabajo se centró en la predicción de band-gaps a partir de los parámetros de diseño de paneles de metamateriales, utilizando redes neuronales como algoritmo de aprendizaje automático. Siguiendo esa misma línea, se propone realizar exploraciones más profundas del

espacio de parámetros, como topologías más complejas o más de dos materiales constituyentes. Otra arista interesante para este problema de predicción, sería probar el caso contrario. Es decir, diseñar el panel dado un set de propiedades del band-gap fonónico, utilizando estos mismos algoritmos.



# Bibliografía

- [1] C. Valencia, J. Gomez, and N. Guarín-Zapata. A general-purpose element-based approach to compute dispersion relations in periodic materials with existing finite element codes. *Journal of Theoretical and Computational Acoustics*, 2019.
- [2] H. N. Wadley. Multifunctional periodic cellular metals. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 364(1838):31–68, 2006.
- [3] A. Khelif, A. Choujaa, S. Benchabane, B. Djafari-Rouhani, and V. Laude. Experimental study of guiding and filtering of acoustic waves in a two dimensional ultrasonic crystal. *Zeitschrift Für Kristallographie - Crystalline Materials*, 220:836–840, 2005.
- [4] A. Géron. *Hands on Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. O’Reilly Media, Inc, 2019.
- [5] Hong, K. Artificial neural network. <https://www.bogotobogo.com/python/scikit-learn/Artificial-Neural-Network-ANN-1-Introduction.php>, 2020. [Online; Accessed on April 19 2021].
- [6] V. Meruane. Redes neuronales, regularización, selección del modelo. In *ME7200 - Procesamiento de señales y aprendizaje de máquinas en mantenimiento predictivo*, pages 30–34. Departamento de Ingeniería Mecánica, Universidad de Chile, 2021.
- [7] B. Ramsundar and R. B. Zadeh. *TensorFlow for Deep Learning*. O’Reilly Media, Inc, 2018.
- [8] V. Fedotov. Metamaterials. In *Springer Handbook of Electronic and Photonic Materials*, pages 1351–1377. Springer International Publishing, 2017.
- [9] D. Del Vescovo and I. Giorgio. Dynamic problems for metamaterials: Review of existing models and ideas for further research. *International Journal of Engineering Science*, 80:153–172, 2014.
- [10] M.I. Hussein, M.J. Leamy, and M. Ruzzene. Dynamics of phononic materials and structures: Historical origins, recent progress, and future outlook. *Applied Mechanics Reviews*, 2014.
- [11] J.F. Davalos, P. Qiao, X.F. Xu, J. Robinson, and K.E. Barth. Modeling and cha-

- racterization of fiber-reinforced plastic honeycomb sandwich panels for highway bridge applications. *Composite Structures*, 52(3-4):441–452, 2001.
- [12] V. D’Alessandro, G. Petrone, F. Franco, and S. De Rosa. A review of the vibroacoustics of sandwich panels: Models and experiments. *Journal of Sandwich Structures & Materials*, 15(5):541–582, 2013.
- [13] P.A. Deymier. Acoustic metamaterials and phononic crystals. *Springer Science & Business Media*, 173:1–12, 2013.
- [14] O. Sigmund and J.S. Jensen. Systematic design of phononic band-gap materials and structures by topology optimization. *Philosophical Transactions of the Royal Society of*, 361(1806):1001–1019, 2003.
- [15] M. Sigalas, M. S. Kushwaha, E. N. Economou, M. Kafesaki, I. E. Psarobas, and W. Steurer. Classical vibrational modes in phononic lattices: theory and experiment. *Zeitschrift Für Kristallographie - Crystalline Materials*, 220:9–10, 2005.
- [16] L. Quinteros. *Optimizaci3n de Band-Gap en materiales tipo s3ndwich*. Universidad de Chile, 2020.
- [17] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning internal representations by error propagation. In *Parallel Distributed Processing: Explorations in the Microstructure of Cognition: Foundations*, pages 318–362. MIT Press, 1986.
- [18] F. Li, R. Krishna, and D. Xu. Convolutional neural networks. In *CS231n - Convolutional Neural Networks for Visual Recognition Course*. Computer Science Department, Stanford University, 2021.
- [19] X. Zhang. Neural networks. In *A Matrix Algebra Approach to Artificial Intelligence*, pages 441–615. Springer International Publishing, 2020.
- [20] X. Zhang. Machine learning. In *A Matrix Algebra Approach to Artificial Intelligence*, pages 223–440. Springer International Publishing, 2020.
- [21] Brownlee, J. How to use data scaling improve deep learning model stability and performance. <https://machinelearningmastery.com/how-to-improve-neural-network-stability-and-modeling-performance-with-data-scaling/>, 2020. [Online; Accessed on April 30 2021].
- [22] Scikit-Learn Developers. 3.3. metrics and scoring: quantifying the quality of predictions. In *Scikit-Learn 0.22. 1 Documentation*. 2019.
- [23] Stanton A. Glantz and B. K. Slinker. *Primer of Applied Regression and Analysis of Variance*. McGraw-Hill, 1990.
- [24] L. J. Saunders, R. A. Russell, and D. P. Crabb. The coefficient of determination: what determines a useful r2 statistic? *Investigative Ophthalmology & Visual Science*, 53(11):6830–6832, 2012.

- [25] Tripathi, A. What is the coefficient of determination. <https://ashutoshtripathi.com/2019/01/22/what-is-the-coefficient-of-determination-r-square>, 2019. [Online; Accessed on April 30 2021].
- [26] S. Astudillo. *Optimización topológica de un panel de metamaterial para la supresión de vibraciones a través de un metamodelo de Band-Gap en materiales tipo sándwich*. Universidad de Chile, 2020.
- [27] A. Olsson, G. Sandberg, and O. Dahlblom. On latin hypercube sampling for structural reliability analysis. *Structural Safety*, 25(1):47–68, 2003.

# Anexo A

## Códigos

Código A.1: Ejemplo de arquitectura de modelo de redes neuronales según Enfoque I.

```
1
2 # cargar archivo .npz que contiene los dataset
3 npz_spt = np.load('/content/datasets/splits2_nc4.npz')
4
5 # extraer sets de datos
6 X_train, Y_train = npz_spt['X_train'], npz_spt['Y_train']
7 X_val, Y_val = npz_spt['X_val'], npz_spt['Y_val']
8 X_test, Y_test = npz_spt['X_test'], npz_spt['Y_test']
9
10 # normalizar
11 mmx_scaler = MinMaxScaler()
12 X_train = mmx_scaler.fit_transform(X_train)
13 X_val = mmx_scaler.transform(X_val)
14 X_test = mmx_scaler.transform(X_test)
15
16 Y_train = mmx_scaler.fit_transform(Y_train)
17 Y_val = mmx_scaler.transform(Y_val)
18 Y_test = mmx_scaler.transform(Y_test)
19
20 # guardar en archivo .npz
21 np.savez('//content//E1FC.npz',
22         X_train=X_train, Y_train=Y_train,
23         X_val=X_val, Y_val=Y_val,
24         X_test=X_test, Y_test=Y_test)
25
26 #####construccion modelo NN#####
27
28 # cargar archivo .npz que contiene los sets de entrenamiento
29 npz_dst = np.load('//content//E1FC.npz')
30
31 # extraer sets de datos
32 X_train, Y_train = npz_dst['X_train'], npz_dst['Y_train']
```

```

33 X_val, Y_val = npz_dst['X_val'], npz_dst['Y_val']
34 X_test, Y_test = npz_dst['X_test'], npz_dst['Y_test']
35
36
37 # inicializar modelo keras.Sequential
38 model = Sequential()
39
40 # agregar capa Input
41 input_dim = (X_train.shape[1], )
42 model.add(GaussianNoise(0.01, input_shape= input_dim))
43 #model.add( Input( shape=input_dim ) )
44
45 # agregar las capas Dense.
46 model.add( Dense(units=64, activation='tanh') )
47 model.add( Dense(units=64, activation='tanh') )
48 model.add( Dense(units=64, activation='tanh') )
49 model.add( Dropout(rate=0.2) )
50 model.add( Dense(units=128, activation='tanh') )
51 model.add( Dense(units=128, activation='tanh') )
52 model.add( Dense(units=128, activation='tanh') )
53 model.add( Dropout(rate=0.2) )
54 model.add( Dense(units=256, activation='tanh') )
55 model.add( Dense(units=256, activation='tanh') )
56 model.add( Dense(units=256, activation='tanh') )
57
58 # por último configurar la capa de salida
59 labels_num = 2
60 model.add( Dense(units=labels_num, activation='linear') )
61
62 # -----
63 # compilar modelo
64 opt = Adam(learning_rate=1e-3)
65 model.compile(loss='mse', optimizer=opt, metrics = ['mae'])
66
67 #early stopping
68 es = EarlyStopping(monitor='val_loss', mode='min', verbose=1, patience=200)
69
70 # entrenar modelo
71 model_history = model.fit(X_train, Y_train, batch_size=256, epochs=600,
72                           validation_data = (X_val, Y_val), callbacks=[es])
73
74 # -----
75 # obtener predicciones de X_test
76 Y_pred = model.predict(X_test)
77
78 # calcular accuracy de la clasificación.
79 accuracy = mean_squared_error(Y_test, Y_pred, squared= False)
80 print('validation accuracy: {:.13f}\n'.format(accuracy))
81
82 # -----

```

```

83 # plot gráfico de función de pérdida
84 plot_loss_function(model_history, figsize=(10,4))
85
86 # evaluate the model
87 _, train_acc = model.evaluate(X_train, Y_train, verbose=0)
88 _, test_acc = model.evaluate(X_test, Y_test, verbose=0)
89 print('Train: %.3f, Test: %.3f' % (train_acc, test_acc))
90
91 Y_pred = mmx_scaler.inverse_transform(Y_pred)
92 Y_test = mmx_scaler.inverse_transform(Y_test)
93
94 dwt = []
95 wt = []
96
97 dwt_pred = []
98 wt_pred = []
99
100
101 for i in range(len(Y_test)-1):
102     dwt.append(Y_test[i][0])
103     wt.append(Y_test[i][1])
104
105 for i in range(len(Y_pred)-1):
106     dwt_pred.append(Y_pred[i][0])
107     wt_pred.append(Y_pred[i][1])
108
109 # inicializar un dataframe vacío para agregar los arreglos anteriores
110
111 columns = ['dwt', 'wt', 'dwt_pred', 'wt_pred']
112
113 df_2 = pd.DataFrame(columns=columns)
114 df_2 = df_2.fillna(0) # with 0s rather than NaNs
115
116 df_2['dwt'] = dwt
117 df_2['wt'] = wt
118 df_2['dwt_pred'] = dwt_pred
119 df_2['wt_pred'] = wt_pred
120
121
122 X = np.array(df_2['wt']).reshape(-1, 1)
123 y = df_2['wt_pred']
124 rf = LinearRegression()
125 rf.fit(X, y)
126 y_pred = rf.predict(X)
127
128 # graficar etiquetas reales vs predichas de la frecuencia media
129
130 plt.scatter(df_2['wt'], df_2['wt_pred'])
131 plt.plot(X, y_pred, color='red')
132 plt.xlabel('Real')

```

```

133 plt.ylabel('Predicho')
134 plt.title('Coeficiente de Determinación (R2) para wt')
135
136 # calculo r2 para la predicción de la frecuencia media
137 r2_score(wt, wt_pred)
138
139
140 # graficar etiquetas reales vs predichas del ancho de banda
141
142 X = np.array(df_2['dwt']).reshape(-1, 1)
143 y = df_2['dwt_pred']
144 rf2 = LinearRegression()
145 rf2.fit(X, y)
146 y_pred = rf2.predict(X)
147
148 plt.scatter(df_2['dwt'], df_2['dwt_pred'])
149 plt.plot(X, y_pred, color='red')
150 plt.xlabel('Real')
151 plt.ylabel('Predicho')
152 plt.title('Coeficiente de Determinación (R2) para dwt')
153
154 # calculo r2 para la predicción del ancho de banda
155 r2_score(dwt, dwt_pred)

```

El código anterior corresponde al modelo fully connected implementado para la predicción de band-gaps en la Estructura 1 según el Enfoque I. Para un modelo CNN, solo se tienen diferencias en las series de capas agregadas:

#### Código A.2: Arquitectura modelo CNN para Enfoque I.

```

1
2 # inicializar modelo keras.Sequential
3 model = Sequential()
4
5 # las capas Conv2D reciben tensores de la forma (height, width, channels)
6 input_dim = ( 2, 12, 1)
7 model.add(GaussianNoise(0.1, input_shape= input_dim))
8
9 # ---
10 # agregar las capas CNN
11
12
13 model.add( Conv2D(64, (3, 3), activation='tanh', padding='same' ) )
14 model.add( Conv2D(64, (3, 3), activation='tanh', padding='same' ) )
15 model.add( Conv2D(128, (5, 5), activation='tanh', padding='same' ) )
16 model.add( Conv2D(128, (5, 5), activation='tanh', padding='same' ) )
17
18
19 # agregar las capas Dense para procesar la #información hasta la capa de salida.
20

```

```
21 model.add( Flatten() )
22 model.add( Dropout(rate=0.1) )
23 model.add( Dense(units=96, activation='tanh') )
24 model.add( Dense(units=96, activation='tanh') )
25 model.add( Dropout(rate=0.1) )
26 model.add( Dense(units=96, activation='tanh') )
27
28 # por último configurar la capa de salida
29 labels_num = 2
30 model.add( Dense(units=labels_num, activation='linear') )
31
32 # print model.summary()
33 model.summary()
```