UNIVERSIDAD NACIONAL AUTÓNOMA DE MEXICO

PROGRAMA DE MAESTRÍA Y DOCTORADO EN CIENCIAS QUÍMICAS

DESCUBRIMIENTO DE COMPUESTOS CON ESTRUCTURA TIPO PEROVSKITA A TRAVÉS DE INTELIGENCIA ARTIFICIAL Y CÁLCULOS QUÍMICO CUÁNTICOS

TESIS

PARA OPTAR POR EL GRADO DE

DOCTOR EN CIENCIAS

PRESENTA

M. en C. JUAN IVÁN GÓMEZ PERALTA

DR. J. GUADALUPE PÉREZ RAMÍREZ
INSTITUTO DE FÍSICA, UNAM

CIUDAD UNIVERSITARIA, CD. MX., ENERO 2021

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

PROGRAMA DE MAESTRÍA Y DOCTORADO EN CIENCIAS QUÍMICAS

DESCUBRIMIENTO DE COMPUESTOS CON ESTRUCTURA TIPO PEROVSKITA A TRAVÉS DE INTELIGENCIA ARTIFICIAL Y CÁLCULOS QUÍMICO CUÁNTICOS

T E S I S
PARA OPTAR POR EL GRADO DE

DOCTOR EN CIENCIAS

P R E S E N T A

M. en C. JUAN IVÁN GÓMEZ PERALTA

DR. J. GUADALUPE PÉREZ RAMÍREZ
INSTITUTO DE FÍSICA, UNAM

Ciudad de México, ENERO 2021.

El presente trabajo de tesis se desarrolló en el Laboratorio de Inteligencia Artificial del Instituto de Física de la Universidad Nacional Autónoma de México, de febrero de 2017 a diciembre de 2020.

Parte de los resultados obtenidos en este trabajo doctoral se presentaron en el IX Congreso Nacional de Cristalografía, celebrado en la ciudad de Oaxaca, Oaxaca, México, del 20 – 25 de octubre de 2018.

Agradecimientos

Al Consejo Nacional de Ciencia y Tecnología (CONACyT) por la beca otorgada para poder realizar mis estudios de doctorado (No. de becario: 336003).

Al Laboratorio de Inteligencia Artificial del Instituto de Física de la Universidad Nacional Autónoma de México por las facilidades prestadas para realizar el cómputo que demandó el presente trabajo de tesis, así como también por el financiamiento durante el IX Congreso Nacional de Cristalografía.

A mi tutor principal, el Dr. J. Guadalupe Pérez Ramírez (Xim Bokhimi). Hay personas que transforman tu visión de las cosas al trabajar con ellas. Esto me ha pasado a mí después de estos cuatro años de estudios doctorales. Y el Dr. Bokhimi es responsable de este cambio.

A los miembros que integran mi sínodo, por su tiempo y comentarios hechos al presente manuscrito: al Dr. Luis Emilio Orgaz Baque, al Dr. Rodolfo Zanella Specia, al Dr. José Enrique Barquera Lozada, al Dr. Fernando Cortés Guzmán, y al Dr. Edilso Francisco Reguera Ruiz.

Al Dr. Carlos Amador Bedolla, quien junto con el Dr. Fernando Cortés, fue parte de mi Comité Tutor.

Al Dr. René Luna García, por sus observaciones hechas en cuanto a temas de aprendizaje de máquina e inteligencia artificial durante mis exámenes de candidatura y de Comité Tutor Ampliado.

A mis compañeros de laboratorio: Christian Rodríguez Martínez, Juan Daniel Soto Montes, Karen Daniela Cruz Hernández, Óscar Márquez Esquivel, y José Jasso Guzmán. De igual manera, también agradezco al Técnico Antonio Morales Espino, al Mtro. Alejandro Herrera González y a la Dra. Nancy Vargas Becerril.

A mi compañera de vida, la Dra. Nidia García Peña, por su enorme paciencia, cariño y su confianza en mí.

ÍNDICE

ÍNDICE DE FIGURAS Y TABLAS

En este trabajo, la convención para ordenar y ubicar las tablas y figuras es la siguiente:

1. El primer número hace referencia al capítulo de la tesis.
2. El segundo número se refiere a la sección del capítulo.
3. El último número enumera la figura o tabla dentro de la sección de ese capítulo.

Ejemplo:

Figura 6.3.5: Se encuentra en el capítulo 6, en la sección 3 y es la Figura 5 dentro de la misma sección.

A. Índice de figuras

B. Índice de tablas

A Rosa Peralta† y a Nidia García.

A Tulio, a Tollo, a Mona y a Hitomi†.

PRESENTACIÓN

Esta tesis está organizada de la siguiente manera:

En la Introducción (Capítulo 1) se revisan temas de Cristalografía, Perovskitas e Inteligencia Artificial. El propósito de revisar conceptos de Cristalografía es presentar que, gracias a la existencia de grupos de simetría, un sistema cristalino, con una cantidad de partículas del orden del número de Avogrado, se describe con un número reducido de átomos que se encuentran dentro de la celda unitaria y que, además, corresponden a diferentes entornos químicos caracterizados por los sitios de Wyckoff. Posteriormente, se presenta a la estructura perovskita como aquella definida por un marco de octaedros de vértice compartido y su conexión con la descripción en términos de sitios de Wyckoff ocupados. La estructura perovskita es paradigmática para explicar fenómenos como la piezoelectricidad, la superconductividad y, recientemente, ha estado relacionada con las celdas fotovoltaicas. Adicionalmente, se presenta a los grupos espaciales que corresponden a una estructura tipo perovskita distorsionada. Esta desviación de la estructura ideal modifica la cantidad de entornos químicos diferentes con los que se describe el sistema cristalino e influye en la caracterización requerida para el desarrollo del modelo de Inteligencia Artificial. Sobre este último tema de la parte introductoria, más que presentar una revisión de las diferentes definiciones sobre lo que este término implica, se tiene por objetivo explicar en qué consiste el aprendizaje de máquina (traducción comúnmente empleada del término *Machine Learning*, en inglés) y cómo funciona el algoritmo de redes neuronales artificiales, que fue el que se utilizó en esta tesis. Para terminar la Introducción, se hace una revisión de algunos trabajos relacionados a esta tesis que utilizaron algún algoritmo de aprendizaje de máquina. Lo anterior tiene la intención de presentar algunas aproximaciones que se han implementado para caracterizar a las muestras de una colección: si bien es cierto que las redes neuronales artificiales dotan de autonomía a una computadora para que ésta *aprenda* a resolver una tarea, su desempeño está condicionado a una adecuada construcción del vector de datos de entrada (caracterización) de las muestras. Considero que la caracterización implementada

en este trabajo de tesis representa la parte fina del modelo desarrollado. Además, la caracterización de las muestras fue quizás lo que más tardó en concebirse ya que, desde el comienzo, se buscó que ésta no dependiera de un grupo espacial y que, de alguna manera, fuera general a cualquier tipo de estructura cristalina.

Una vez revisados los elementos para comprender mejor esta tesis, se plantea el problema (Capítulo 2), su hipótesis (Capítulo 3) y los objetivos de este trabajo (Capítulo 4). El problema que se plantea en esta tesis se inspira en algunas iniciativas que tienen como objetivo acelerar el proceso de descubrimiento – comercialización de un material. Para esto, es necesario el uso de técnicas computacionales que tengan incluso un menor costo computacional que los comúnmente utilizados (cálculos químico cuánticos). Es ahí donde las redes neuronales artificiales entran en juego como posible alternativa a los cálculos tradicionales. No obstante, se requiere de una metodología adecuada para caracterizar a los compuestos cristalinos y que sea aplicable a cualquier tipo de estructura. Las premisas de esta metodología se señalan puntualmente en la Hipótesis, que servirán para inferir nuevos compuestos con la estructura perovskita.

La metodología implementada en este trabajo se describe en el Capítulo 5 y tiene que ver con la obtención y curación de la base de datos, la caracterización de las muestras utilizadas para desarrollar las redes neuronales artificiales y la validación de los compuestos que, de acuerdo a las redes neuronales entrenadas, son candidatos potenciales a cristalizar en estructura tipo perovskita. El código desarrollado, *patolli.py*, para entrenar a las redes neuronales se puede consultar en GitHub (https://github.com/gomezperalta/) y en esta parte de la tesis se explica paso a paso cómo funciona.

Los resultados obtenidos y su discusión se presentan en el Capítulo 6. Este capítulo inicia presentando la naturaleza de los compuestos utilizados en el desarrollo de las redes neuronales en términos de número de sitios de Wyckoff o entornos químicos diferentes, grupos espaciales y composición química. Posteriormente, los desempeños de las mejores redes neuronales desarrolladas se muestran en términos de las métricas de precisión, exhaustividad y valor F1. Los

resultados obtenidos fueron prometedores y, en parte, justifican la caracterización implementada en esta tesis. La parte de resultados concluye la presentación de algunos compuestos de tipo halogenuro que la red neuronal encontró como candidatos y que fueron validados con cálculos de energía de optimización de la geometría con condiciones periódicas en la frontera. En efecto, se encontró que algunos de estos compuestos son candidatos a ser usados en dispositivos fotovoltaicos.

Finalmente, se presentarán las conclusiones y las perspectivas de este trabajo en los capítulos 7 y 8. Además de los compuestos inferidos como candidatos, considero que se tiene una metodología que se puede aplicar a otras estructuras cristalinas, que sirve como base para estudiar las propiedades de los materiales, y que es necesaria su incorporación en la currícula de cualquier científico interesado en el descubrimiento de nuevos materiales.

1. ANTECEDENTES

1.1 CRISTALOGRAFÍA

La siguiente exposición sobre los conceptos de cristalografía, con los que se desarrolló esta tesis, pueden consultarse en las referencias 1 – 4.

1.1.1 CELDA UNITARIA

Los compuestos cristalinos se caracterizan por poseer un arreglo de largo alcance en sus partículas (átomos, iones o moléculas). Derivado de este arreglo de largo alcance es posible identificar una unidad mínima, conocida como *celda unitaria*, la cual se repite de manera periódica en el espacio para describir a todo el cristal. Al introducir el concepto de celda unitaria también se presenta a los compuestos cristalinos como sistemas que tienen simetría traslacional. La existencia de simetría traslacional permite simplificar la descripción de un sistema de 10^{23} partículas al de unas cuantas, que corresponden a las que se encuentran dentro de la celda unitaria (Figura 1.1.1).

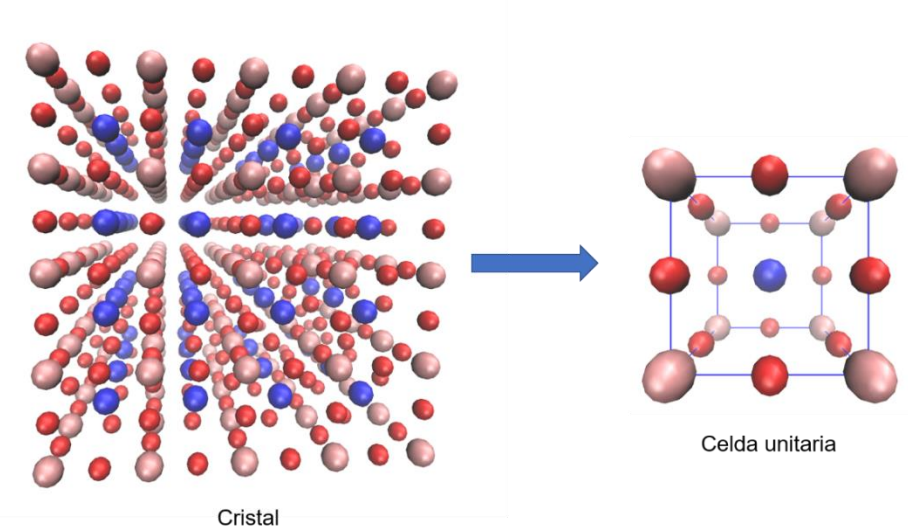


Figura 1.1.1: A la izquierda se ilustra un cristal de $BaTiO_3$. Los átomos de bario, titanio y oxígeno están representados con los colores azul, rosa y rojo, respectivamente. En el cristal de la izquierda se puede identificar una unidad mínima, que se representa a la derecha, y se conoce como celda unitaria. Mediante la aplicación de la simetría traslacional en los tres ejes espaciales se recupera el cristal descrito en la izquierda. De esta manera, la descripción de un cristal se reduce al contenido de la celda unitaria.

Otra forma de definir a un cristal es a través de la regla de composición que se ilustra en la Figura 1.1.2:

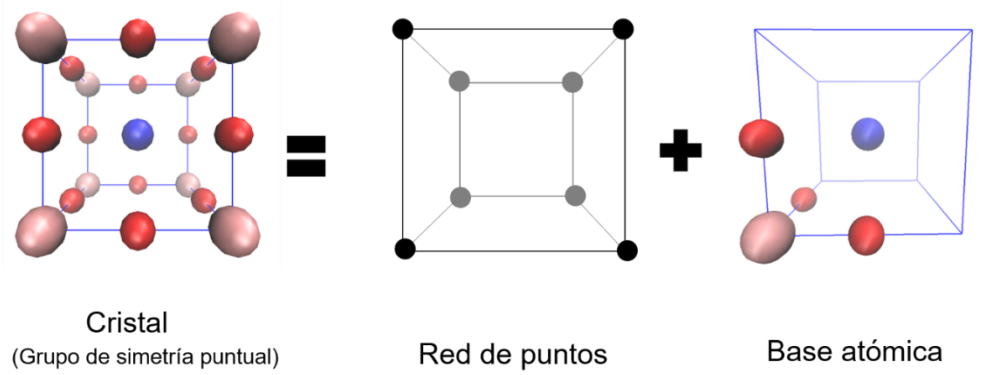


Figura 1.1.2: Definición de cristal como la composición de una red de puntos y una base atómica.

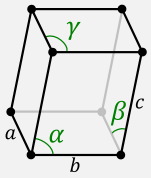
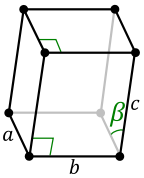
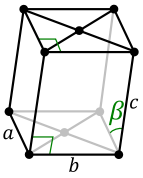
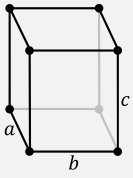
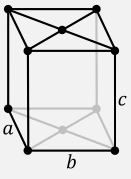
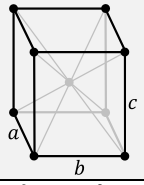
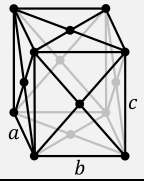
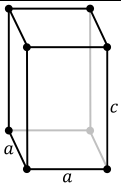
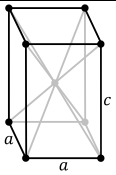
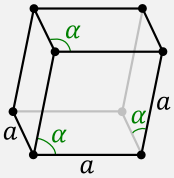
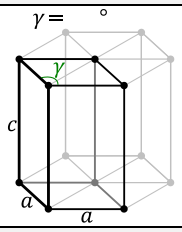
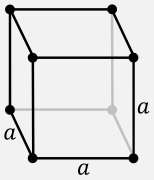
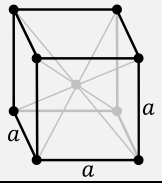
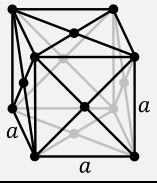
La red de puntos es una descripción matemática de puntos que no corresponde a las partículas del cristal. En un espacio de tres dimensiones, existen únicamente 14 diferentes redes de puntos con los que se puede llenar por completo mediante el uso de operaciones de simetría de traslación. A estas redes de puntos se les conoce como redes de Bravais. En la Tabla 1.1.1 se presentan las celdas unitarias de las catorce redes de Bravais.

Por otra parte, a cada punto de la red de Bravais es posible asociar un grupo de partículas del cristal. Este grupo de partículas asociado a cada punto de la red es lo que se conoce como base atómica. En el caso más simple, a cada punto de la red de Bravais le corresponde un átomo. No obstante, no hay un límite en la cantidad de partículas asociadas a un punto de la red.

1.1.2 GRUPOS DE SIMETRÍA

Los compuestos cristalinos se pueden considerar como sistemas *muy simétricos*. Esto significa que la distribución de los átomos en la celda unitaria puede describirse a través de grupos de simetría. Un grupo es un conjunto de elementos los cuales, a través de una regla de composición, satisfacen las propiedades de 1) Cierre, 2) Asociatividad, 3) Existencia del elemento neutro, 4) Existencia del inverso para cada elemento del grupo.

Tabla 1.1.1: Celdas unitarias de las 14 redes de Bravais. Las celdas unitarias están organizadas tomando en cuenta los siete sistemas cristalinos y el centrado de éstas: P para una celda primitiva; B, para un centrado en las bases; I, para un centrado en el cuerpo; y F, para el centrado en las caras.

	P	B	I	F
Triclínico				
Monoclínico				
Ortorrómbico				
Tetragonal				
Trigonal				
Hexagonal				
Cúbico				

Las colecciones de elementos de simetría que satisfacen las propiedades anteriormente enlistadas son, por lo tanto, grupos de simetría. Entre más elementos de simetría tenga un grupo se considera que el nivel de simetría del mismo es mayor

y, por lo tanto, también el sistema es más simétrico. El número de elementos de simetría de un grupo se le conoce como *orden del grupo*.

Además de la existencia de la simetría traslacional en los sistemas cristalinos, pueden existir en éstos otros elementos tales como:

- Ejes de rotación de orden *n*, donde *n* es el valor del denominador en el cociente $\frac{360^{\circ}}{n}$. Dicho cociente es el valor del ángulo con el cual hay que rotar al sistema alrededor del eje de rotación para que éste permanezca invariante. En tres dimensiones las únicas rotaciones posibles son aquellas donde *n* = 1,2,3,4 o 6.
- Planos de reflexiones.
- Centros de inversión.
- Centros de rotoinversión de orden $\bar{n}$, que son acoplamientos entre una rotación de orden *n* con un centro de inversión.

Las colecciones de los elementos de simetría anteriormente enlistados, sin considerar a la traslación, que satisfacen las propiedades de grupo se conocen como grupos de simetría puntual. Se llaman así ya que los elementos que conforman a estos grupos dejan al menos un punto inmóvil en el espacio. En tres dimensiones, existen 32 grupos puntuales, que se enlistan en la Tabla 1.1.2.

Para formar los 32 grupos puntuales no se tomó en cuenta la existencia de la simetría traslacional. Cuando este tipo de simetría se considera, surgen otros elementos de simetría del acoplamiento de la traslación con una rotación o con una reflexión. A dichos acoplamientos se les conoce como rotaciones helicoidales y reflexiones deslizadas, respectivamente. Así, al considerar la simetría traslacional, se tiene que en tres dimensiones existen 230 grupos espaciales. La lista de los 230 grupos espaciales se puede consultar en el volumen A de las Tablas Internacionales de Cristalografía.

La nomenclatura más usada para representar a los grupos espaciales es la que se conoce como Notación Internacional. Esta notación se compone de las siguientes partes:

- Al comienzo, existe una letra que indica el centrado de la celda unitaria la cual puede ser primitiva **P**, centrada en las caras **F**, centrada en el cuerpo **I**, o centrada en las bases, **A**, **B** o **C**.
- El grupo de simetría espacial del cristal. Éste está representado por tres símbolos adyacentes. Cada uno de estos símbolos corresponde al elemento de simetría característico de las tres direcciones principales del sistema cristalino.

Los 230 grupos espaciales se clasifican en siete sistemas cristalinos. Estos sistemas cristalinos son presentados comúnmente en libros introductorios de Estado Sólido a través de la relación que guardan sus parámetros de red (vectores unitarios y ángulos); sin embargo, los sistemas cristalinos dependen de la existencia de un elemento de simetría característico en su celda unitaria:

- Triclínico: Es el sistema cristalino de más baja simetría y se caracteriza por la ausencia tanto de ejes de rotación diferentes a la unidad como de planos de reflexión.
- Monoclínico: Existe sólo un eje de rotación doble paralelo a uno de los vectores unitarios.
- Ortorrómbico: Existen tres ejes de rotación doble, que son ortogonales entre ellos.
- Tetragonal: Existe sólo un eje de rotación cuádruple paralelo a uno de los vectores unitarios.
- Trigonal: Existe sólo un eje de rotación triple paralelo a uno de los vectores unitarios.
- Hexagonal: Existe sólo un eje de rotación séxtuple paralelo a uno de los vectores unitarios.
- Cúbico: Existe al menos un eje de rotación triple a lo largo de la diagonal de la celda unitaria

Tabla 1.1.2: Los 32 grupos puntuales en los sistemas cristalinos

No.	Notación Internacional	Notación Schönflies	Orden del grupo	Elementos de simetría en los ejes característicos		
		Triclínico		*a*	*b*	*c*
1	1	C_1	1	-	-	-
2	$\bar{1}$	C_i	2	-	-	-
		Monoclínico		*a*	*b*	*c*
3	2	C_2	2	-	2	-
4	m	C_s	2	-	m	-
5	$\frac{2}{m}$	C_{2h}	4	-	$2/m$	-
		Ortorrómbico		*a*	*b*	*c*
6	222	D_2	4	2	2	2
7	$mm2$	C_{2v}	4	m	m	2
8	mmm $(\frac{2}{m}\frac{2}{m}\frac{2}{m})$	D_{2h}	8	$2/m$	$2/m$	$2/m$
		Tetragonal		*c*	*a*	*<110>*
9	4	C_4	4	4	-	-
10	$\bar{4}$	S_4	4	$\bar{4}$	-	-
11	$\frac{4}{m}$	C_{4h}	8	$4/m$	-	-
12	422	D_4	8	4	2	2
13	$4mm$	C_{4v}	8	4	2	m
14	$\bar{4}2m$	D_{2d}	8	$\bar{4}$	2	m
15	$4/mmm$ $(\frac{4}{m}\frac{2}{m}\frac{2}{m})$	D_{4h}	16	$4/m$	$2/m$	$2/m$
		Trigonal		c	a	-
16	3	C_3	3	3	-	-
17	$\bar{3}$	S_6	6	$\bar{3}$	-	-
18	32	D_3	6	3	2	-
19	$3m$	C_{3v}	6	3	m	-
20	$\bar{3}m$ $(\bar{3}2/m)$	D_{3d}	12	$\bar{3}$	$2/m$	-
		Hexagonal		c	a	<210>
21	6	C_6	6	6	-	-
22	$\bar{6}$	C_{3h}	6	$\bar{6}$	-	-
23	$\frac{6}{m}$	C_{6h}	12	$6/m$	-	-
24	622	D_6	12	6	2	2
25	$6mm$	C_{6v}	12	6	m	m
26	$\bar{6}m2$	D_{3h}	12	$\bar{6}$	m	2
27	$6/mmm$ $(\frac{6}{m}\frac{2}{m}\frac{2}{m})$	D_{6h}	24	$6/m$	$2/m$	$2/m$
		Cúbico		a	<111>	<110>
28	23	T	12	2	3	-
29	$m\bar{3}$ $(2/m\bar{3})$	T_h	24	$2/m$	$\bar{3}$	-
30	432	O	24	4	3	2
31	$\bar{4}3m$	T_d	24	$\bar{4}$	3	m
32	$m\bar{3}m$ $(\frac{4}{m}\bar{3}\frac{2}{m})$	O_h	48	$4/m$	$\bar{3}$	$2/m$

1.1.3 SITIOS DE WYCKOFF

Además de que la celda unitaria se caracteriza con un grupo de simetría, dentro de la misma hay posiciones que se identifican con subgrupos invariantes del grupo de simetría puntual de la celda unitaria. Dichas posiciones se conocen como sitios de Wyckoff.

Los sitios de Wyckoff se denotan a través de un coeficiente y un símbolo. El coeficiente se conoce como multiplicidad y designa cuántas posiciones dentro la celda unitaria comparten ese mismo subgrupo de simetría puntual. El símbolo es una letra del alfabeto que representa a un subgrupo de simetría puntual. Dicho subgrupo puntual representado por cada letra varía según el grupo espacial. No obstante, el orden del subgrupo de simetría disminuye en orden alfabético. En el volumen A de las Tablas Internacionales de Cristalografía se puede consultar la localización de estos sitios de Wyckoff según el grupo espacial [5]. A manera de ejemplo, en la Tabla 1.1.3 se recoge la información sobre los sitios de Wyckoff que existen en el grupo espacial $Pnma$, que corresponde al sistema cristalino ortorrómbico.

Para este grupo espacial, existen cuatro sitios de Wyckoff denotados con los símbolos *a, b, c* y *d*. La multiplicidad asociada a cada símbolo de Wyckoff corresponde a las coordenadas, relativas a los parámetros de red, de las posiciones indicadas en la Tabla 1.1.3. De esta manera, los sitios de Wyckoff determinan dónde están localizados los átomos en una celda unitaria que comparten un subgrupo de simetría puntual.

Las posiciones asociadas a un subgrupo de simetría puntual que es diferente al de la identidad se les conoce como posiciones especiales, mientras que aquellas asociadas al subgrupo de simetría puntual identitario se les llama posiciones generales.

En los grupos espaciales donde el centrado de la celda unitaria es primitivo, la multiplicidad de cada sitio de Wyckoff corresponde al cociente entre el orden del grupo de simetría puntual y el del subgrupo asociado al sitio. Como se indicó

anteriormente, este cociente coincide con el número de posiciones o el número de átomos que comparten ese subgrupo de simetría. Cuando el centrado de la celda unitaria es diferente al mencionado, ese cociente se multiplica por el número de puntos que hay dentro de la celda unitaria: dos para cuando el centrado es en el cuerpo (**I**) o en las bases (**A**, **B** o **C**) y cuatro para cuando es en las caras (**F**).

Tabla 1.1.3: Información sobre los sitios de Wyckoff del grupo espacial $Pnma$ (No. 62), que corresponde al sistema ortorrómbico.

Multiplicidad	Símbolo de Wyckoff	Grupo de simetría del sitio	Coordenadas de las posiciones asociadas al sitio de Wyckoff
8	*d*	1	(1) x, y, z (2) $\bar{x}+\frac{1}{2}, \bar{y}, z+\frac{1}{2}$ (3) $\bar{x}, y+\frac{1}{2}, \bar{z}$ (4) $x+\frac{1}{2}, \bar{y}+\frac{1}{2}, \bar{z}+\frac{1}{2}$ (5) $\bar{x}, \bar{y}, \bar{z}$ (6) $x+\frac{1}{2}, y, \bar{z}+\frac{1}{2}$ (7) $x, \bar{y}+\frac{1}{2}, z$ (8) $\bar{x}+\frac{1}{2}, y+\frac{1}{2}, z+\frac{1}{2}$
4	*c*	$.m.$	(1) $x, \frac{1}{4}, z$ (2) $\bar{x}+\frac{1}{2}, \frac{3}{4}, z+\frac{1}{2}$; (3) $\bar{x}, \frac{3}{4}, z$ (4) $x+\frac{1}{2}, \frac{1}{4}, \bar{z}+\frac{1}{2}$
4	*b*	$\bar{1}$	(1) $0,0, \frac{1}{2}$ (2) $\frac{1}{2}, 0, 0$ (3) $0, \frac{1}{2}, \frac{1}{2}$ (4) $\frac{1}{2}, \frac{1}{2}, 0$
4	*a*	$\bar{1}$	(1) $0,0,0$ (2) $\frac{1}{2}, 0, \frac{1}{2}$ (3) $0, \frac{1}{2}, 0$ (4) $\frac{1}{2}, \frac{1}{2}, \frac{1}{2}$

Los átomos que comparten el mismo sitio de Wyckoff están inmersos en un mismo entorno; es decir, tienen el mismo ambiente químico [6]. De aquí se sigue que la descripción de las partículas dentro de la celda unitaria se reduce al número de ambientes diferentes dentro de la misma, es decir, al número de sitios de Wyckoff distintos. Esta idea es muy importante para el desarrollo de esta tesis y se explicará

con mayor detalle en la parte de Metodología. Sin embargo, el lector que desee adelantarse a dicha explicación puede consultar la subsección 5.3.3.

1.2 PEROVSKITAS

1.2.1 ESTRUCTURA DEL ARISTOTIPO

Los compuestos con estructura tipo perovskita han construido un paradigma dentro de los campos de la física, química y la ciencia de materiales [7]. Este tipo de estructuras cristalinas han estado involucradas en materiales con propiedades piezoeléctricas [8], superconductoras [9 – 10] y, desde hace un poco más de una década, han ganado mayor relevancia por su aplicación en dispositivos de conversión fotovoltaica [11 – 13].

La forma de mayor simetría de cualquier tipo de estructura cristalina se conoce como aristotipo. El aristotipo de la estructura perovskita presenta una simetría cúbica, correspondiente al grupo espacial $Pm\bar{3}m$ (No. 221). En la Figura 1.2.1 se representa a un cristal formado del aristotipo de la perovskita. Dicho cristal contiene ocho celdas, dos por cada eje cartesiano. Adicionalmente, la fórmula química más sencilla de un compuesto con este tipo de estructura se representa como ABX_3. En la fórmula, generalmente X se trata de un elemento no metálico mientras que A y B tienen carácter metálico. Además, los compuestos con estructura tipo perovskita se suelen considerar como de tipo iónico, por lo que X se trata, entonces, de un anión y A y B son cationes.

De los elementos de la fórmula ABX_3, se suele considerar que el radio del catión del elemento B es menor que el del elemento A. Así, el elemento B se aloja en sitios donde los primeros vecinos, que corresponden a los aniones X, definen una geometría octaédrica. De hecho, la estructura de tipo perovskita se caracteriza por estar hecha de un marco de octaedros, los cuales comparten sus vértices. Los huecos que se forman de este marco tienen la geometría cuboctaédrica, que es donde se aloja el metal A.

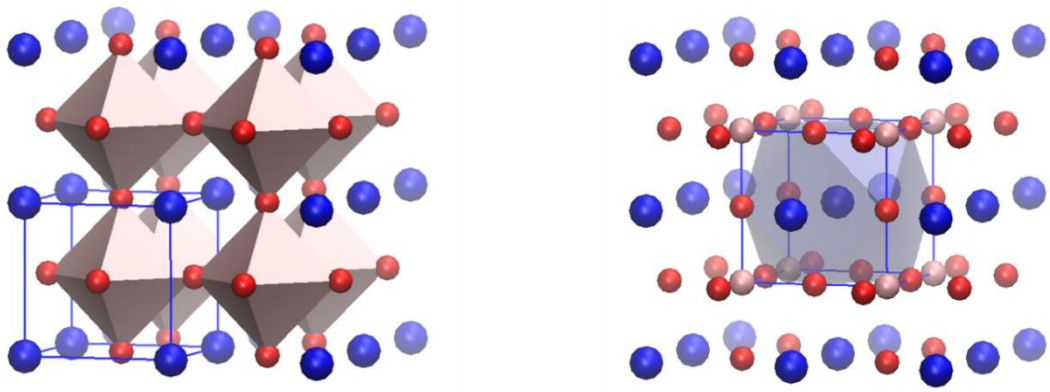


Figura 1.2.1: Dos representaciones de un mismo cristal formado de la estructura aristotípica de la perovskita. El cristal contiene dos veces la celda unitaria en cada dimensión espacial. Dicha celda unitaria se señala con un cubo de aristas azules. En la imagen de la izquierda se aprecia el marco de octaedros de esquina compartida que caracteriza a la estructura perovskita. En la imagen de la derecha se aprecia la geometría de los huecos formados por el marco de octaedros. Dicha geometría corresponde a la de un cuboctaedro.

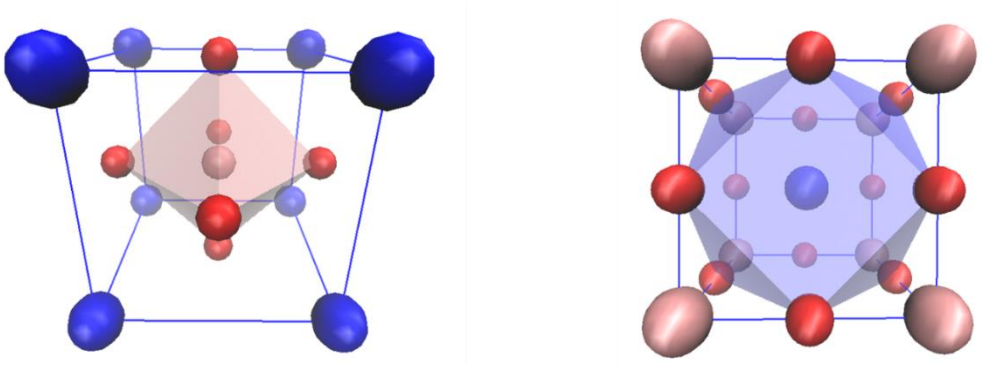


Figura 1.2.2: Dos representaciones de la celda unitaria de la perovskita aristotípica.

En términos cristalográficos, el aristotipo de la estructura perovskita se define por la ocupación de los sitios de Wyckoff [5, 14] *a*, *b* y *c* o de los sitios *a, b* y *d* (Figura 1.2.2). En la primera representación de la celda unitaria, los átomos con geometría octaédrica se localizan en los sitios de Wyckoff con el símbolo *b*, mientras que los átomos con geometría cuboctaédrica se localizan en los sitios de Wyckoff con el símbolo *a* (Figura izquierda 1.2.2). En la segunda representación, los símbolos de Wyckoff están invertidos (Figura derecha 1.2.2). En cualquiera de esas dos representaciones, los átomos en los sitios de Wyckoff con el símbolo *c* o *d* son los que están en los vértices de los octaedros.

Más allá de que los átomos se localicen en sitios con cierto símbolo de Wyckoff, lo importante es que éstos se localicen en sitios con un tipo de simetría puntual. Lo que define a la estructura perovskita, en su forma más simétrica, es la ocupación de los dos sitios con simetría puntual $m\bar{3}m$ y otro con simetría puntual $4/mmm$. En la Tabla 1.2.1 se recoge la información de las Tablas Internacionales de Cristalografía sobre los sitios de Wyckoff para el grupo espacial $Pm\bar{3}m$. En esa misma Tabla sólo se plasma la información de los sitios de Wyckoff que describen al aristotipo de una perovskita.

Tabla 1.2.1: Información sobre los sitios de Wyckoff del grupo espacial $Pm\bar{3}m$ (No. 221), que corresponde al sistema cúbico.

Multiplicidad	Símbolo de Wyckoff	Grupo de simetría del sitio	Coordenadas de las posiciones asociadas al sitio de Wyckoff
3	*d*	$4/mm.m$	(1) $\frac{1}{2},0,0$ (2) $0,\frac{1}{2},0$ (3) $0,0,\frac{1}{2}$
3	*c*	$4/mm.m$	(1) $0,\frac{1}{2},\frac{1}{2}$ (2) $\frac{1}{2},0,\frac{1}{2}$ (3) $\frac{1}{2},\frac{1}{2},0$
1	*b*	$m\bar{3}m$	(1) $\frac{1}{2},\frac{1}{2},\frac{1}{2}$
1	*a*	$m\bar{3}m$	(1) 0,0,0

Para finalizar esta sección, el lector puede comprobar de que la relación de las multiplicidades de los sitios de Wyckoff ocupados en la perovskita aristotípica corresponde a la estequiometría ABX_3.

1.2.1 DESVIACIONES DE LA ESTRUCTURA PEROVSKITA IDEAL

Antes de presentar a los grupos espaciales en los que, además del $Pm\bar{3}m$, se presenta la estructura perovskita es conveniente revisar las razones por las que el marco de octaedros de vértice compartido se deforma.

Las reglas de Pauling sobre la estabilidad de compuestos iónicos [15] establecen que alrededor de cada catión se forma un poliedro de coordinación de

aniones. Dicho poliedro de coordinación está asociado a su cociente de radios iónicos del par catión-anión. En la Tabla 1.2.2 se muestra la relación entre el poliedro de coordinación más estable con el cociente de radios iónicos del par catión-anión. Como se observa en la Tabla 1.2.2, el poliedro de coordinación no tiene asociado un cociente fijo sino un intervalo de valores. Para el caso de un catión con geometría octaédrica, que es característico de la estructura perovskita, el umbral del cociente debe ser 0.414. Dicho umbral se obtiene de considerar un empaquetamiento compacto entre los iones. Naturalmente, el radio del catión es menor que el radio del anión.

Tabla 1.2.2: Relación entre la geometría definida por los primeros vecinos entorno a un catión y su cociente de radios iónicos del par catión-anión.

Geometría del átomo central (catión)	Intervalo de estabilidad
Lineal	$r^+/r^- < 0.155$
Triangular	$.\ 55 \leq r^+/r^- < 0.255$
Tetraédrico	$.\ 55 \leq r^+/r^- < 0.414$
Octaédrico	$.4\ \ 4 \leq r^+/r^- < 0.732$
Cúbico	$.73\ \ \leq r^+/r^-$

Otro de los factores, y quizás el más conocido, para describir la estabilidad de un compuesto con estructura tipo perovskita es el factor de tolerancia de Goldschmidt [16]. El factor de tolerancia se establece considerando que los iones de la estructura adoptan el aristotipo, además de estar en contacto. Con base en la celda unitaria del aristotipo de la perovskita, que en la derecha de la Figura 1.2.2 pone en el centro al átomo con geometría cuboctaédrica, la diagonal de una de las caras, *d*, es el doble de la suma de los radios (iónicos) del catión cuboctaédrico, $r_{cubocta}$, con el del anión, $r_{vértice}$. Por otro lado, una arista de la misma celda unitaria, *a*, es igual al doble de la suma de los radios del catión octaédrico, r_{octa}, con el del anión, $r_{vértice}$. Por cuestiones geométricas, se tiene que la diagonal de una cara de la celda es igual a la arista por $\sqrt{2}$. Así, se cumple lo siguiente:

$$d = \sqrt{2}a$$

$$2(r_{cubocta} + r_{vértice}) = 2\sqrt{2}(r_{octa} + r_{vértice})$$

$$t = \frac{r_{cubocta} + r_{vértice}}{\sqrt{2}\,(r_{octa} + r_{vértice})}$$

donde *t* es el factor de tolerancia. En el caso de que el empaquetamiento de los iones sea el ideal, *t* = 1.

El factor de tolerancia es una medida de la eficiencia en el empaquetamiento de las partículas en la estructura cristalina de la perovskita: si la combinación de los radios iónicos es la adecuada para llenar el espacio, el factor de tolerancia será cercano a la unidad. Por otro lado, un factor de tolerancia menor a la unidad indica que el ion localizado en los huecos cuboctaédricos no es suficientemente grande para estabilizar el marco de octaedros de vértice compartido. Como consecuencia, dicho marco empieza a distorsionarse a fin de estabilizar a la estructura perovskita. La manera más común en que esa distorsión ocurre es mediante la rotación entre los octaedros. Esta rotación da lugar a una reducción en la simetría del sistema, además de un colapso en la geometría de los huecos originalmente cuboctaédricos.

El grado de la distorsión por rotación entre los octaedros se puede describir a través de una notación introducida por A. M. Glazer [17] (1972). Con base en el aristotipo de la perovskita, la rotación existente entre los octaedros de dicha estructura se denota como (Figura 1.2.3):

$$a^0a^0a^0$$

Donde *a* representa el giro en cada uno de los tres ejes ortogonales de rotación cuádruple de un octaedro y el superíndice representa la dirección del giro entre capas adyacentes de octaedros. Aquí, el superíndice *0* denota que no existe rotación entre los octaedros, naturalmente. Cabe mencionar que los ejes de rotación cuádruple de un octaedro son paralelos a los vectores unitarios de la celda cúbica.

En las Figuras 1.2.4 y 1.2.5 se representan dos estructuras perovskitas distorsionadas. Con base en la notación de Glazer, la distorsión en estas perovskitas se representa como $a^0a^0c^+$ y $a^0a^0c^-$, respectivamente, y , a diferencia del aristotipo, se señala que sólo hubo rotación en uno de los ejes cuádruples del

octaedro, que en las Figuras se muestra que es paralelo al eje *z*. Dentro de una misma capa de octaedros, la distorsión ocurre en forma de engrane: considerando al octaedro representado en el centro de la Figura 1.2.4, sus vecinos adyacentes, aquellos con los que tiene conexión, giraron en sentido contrario. Este mecanismo de engrane ocurre en todas las capas de octaedros de las perovskitas distorsionadas por rotación. La diferencia entre las estructuras de la Figuras 1.2.4 y 1.2.5, que se señala en la notación a través de los superíndices + y –, tiene que ver con que el mecanismo de engrane se extienda a los octaedros vecinos ubicados arriba y abajo (capas superior e inferior). En la Figura 1.2.4, dicho mecanismo no se aplicó a los octaedros vecinos ubicados arriba y abajo, por lo que las capas que se apilan a lo largo del eje *z* aparecen eclipsadas. Dicho mecanismo sí se ocurrió en la Figura 1.2.5 y, por esa razón, los octaedros giraron en sentidos contrarios a lo largo del eje *z*.

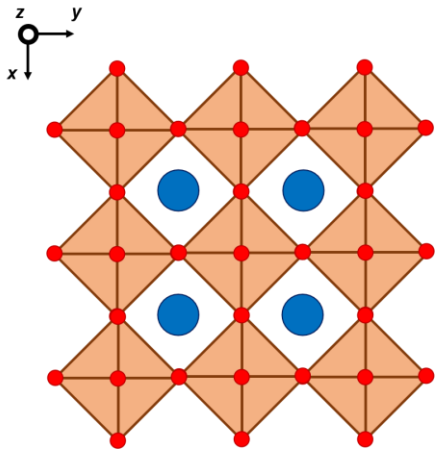


Figura 1.2.3: Vista desde el eje z de un cristal del aristotipo de la perovskita. En esta representación sobre el plano *xy* se muestran cuatro celdas unitarias. En el centro de cada celda unitaria se encuentra el catión cuboctaédrico, con color azul. Los aniones, ubicados en los vértices, se representan en rojo. En el aristotipo no hay distorsión en el marco de octaedros.

En la Tabla 1.2.3 se relacionan las distorsiones en el marco octaédrico por rotación con el grupo espacial al que la estructura perovskita deviene [14, 17 – 19]. En el caso de las perovskitas simples, aquellas cuya fórmula mínima es ABX_3, existen quince combinaciones de rotación que dan origen a diferentes grupos espaciales. Las distorsiones $a^0a^0c^+$ y a^0a^0c, anteriormente explicadas, dan lugar a los grupos

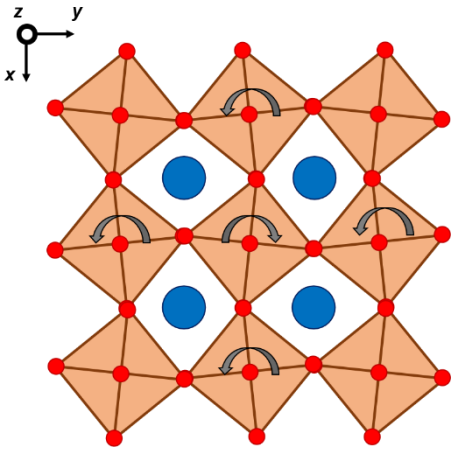


Figura 1.2.4: Se representa un cristal con distorsión en el marco octaédrico por rotación de tipo $a^0a^0c^+$. Esta representación se hace desde el eje *z* que, con base en la notación de esta distorsión, indica que la rotación de los octaedros sólo ocurrió en los ejes de rotación cuádruple paralelos al eje *z*. La rotación de los octaedros dentro de una misma capa (plano *xy*) sigue un mecanismo de engrane: el octaedro en el centro de la Figura giró hacia la derecha, mientras que sus vecinos adyacentes lo hicieron a la izquierda. El superíndice + en la notación $a^0a^0c^+$ señala que el sentido de la rotación en los octaedros ubicados en las capas inferior y superior fue el mismo y, por esa razón, las demás capas de octaedros aparecen eclipsadas.

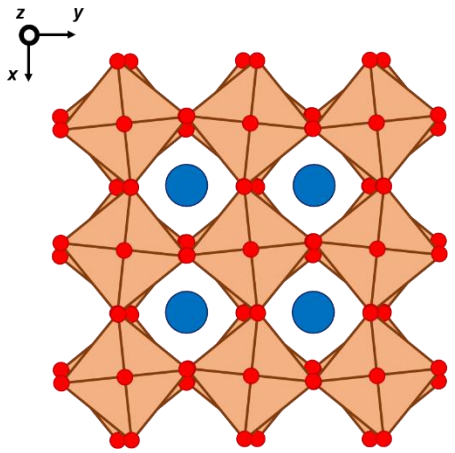


Figura 1.2.5: Se representa un cristal con distorsión en el marco octaédrico por rotación de tipo $a^0a^0c^-$. Esta representación se hace desde el eje *z* que, con base en la notación de esta distorsión, se indica que la rotación de los octaedros sólo ocurrió en los ejes de rotación cuádruple paralelos al eje *z*. La rotación de los octaedros dentro de una misma capa (plano *xy*) sigue el mecanismo de engrane explicado en la Figura 1.2.4. La diferencia con la Figura 1.2.4 está en que el superíndice - en la notación $a^0a^0c^-$ señala que el sentido de la rotación en los octaedros vecinos en las capas inferior y superior fueron en sentidos contrarios.

Tabla 1.2.3: Se muestra el grupo espacial, tanto para perovskitas simples y dobles, que se genera después de una distorsión por rotación en el marco octaédrico. Las perovskitas dobles se caracterizan por tener dos cationes octaédricos diferentes. Debido a que los cationes octaédricos ocupan posiciones definidas en el cristal, los vectores unitarios del aristotipo de las perovskitas dobles es dos la de una perovskita sencilla.

Distorsión	Grupo espacial	
	Perovskitas simples ABX_3	Perovskitas dobles $A_2B'B''X_6$
ARISTOTIPO		
$a^0a^0a^0$	$Pm\bar{3}m$ (221)	$Fm\bar{3}m$ (225)
ROTACIÓN EN 1 EJE		
$a^0a^0c^-$	$I4/mcm$ (140)	$I4/m$ (87)
$a^0a^0c^+$	$P4/mbm$ (127)	$P4/mnc$ (128)
ROTACIÓN EN 2 EJES		
$a^0b^-b^-$	$Imma$ (74)	$I2/m$ (123)
$a^0b^-c^-$	$C2/m$ (12)	$P\bar{1}$ (2)
$a^0b^+c^-$	$Cmcm$ (63)	$C2/c$ (15)
$a^0b^+b^+$	$I4/mmm$ (139)	$P4_2/nnm$ (134)
ROTACIÓN EN LOS 3 EJES		
$a^-a^-a^-$	$R\bar{3}c$ (167)	$R\bar{3}$ (148)
$a^-b^-b^-$	$C2/c$ (15)	$P\bar{1}$ (2)
$a^-b^-c^-$	$P\bar{1}$ (2)	$P\bar{1}$ (2)
$a^+b^-b^-$	$Pnma$ (62)	$P2_1/n$ (14)
$a^+b^-c^-$	$P2_1/m$ (11)	$P\bar{1}$ (2)
$a^+a^+c^-$	$P4_2/nmc$ (137)	$P4_2/n$ (86)
$a^+a^+a^+$	$Im\bar{3}$ (204)	$Pn\bar{3}$ (201)
$a^+b^+c^+$	$Immm$ (74)	$Pnnn$ (48)

espaciales tetragonales $P4/mbm$ y $I4/mcm$, respectivamente. Las perovskitas dobles se caracterizan por tener dos cationes octaédricos distintos que ocupan posiciones definidas en la celda unitaria. Se les llama perovskitas dobles ya que los vectores unitarios del aristotipo (grupo espacial $Fm\bar{3}m$, No. 225) equivalen al doble de los de una perovskita sencilla. La fórmula mínima de este tipo de perovskita es $A_2B'B''X_6$, donde B' y B'' designan a los elementos que ocupan los sitios octaédricos.

En esta tesis, se consideraron todas la perovskitas que se derivan de los aristotipo de las perovskitas simples y dobles. Las llamadas perovskitas hexagonales, que se caracterizan por compartir caras en lugar de vértices, así como las perovskitas mixtas quedaron excluidas del tratamiento para la elaboración del modelo de redes neuronales artificiales.

1.3 INTELIGENCIA ARTIFICIAL

Se le atribuye a John McCarthy (1956) la acuñación [20] del término *Inteligencia Artificial* (IA) para referirse a la ciencia e ingeniería que construye máquinas que manifiestan una inteligencia similar a la humana [21]. También se considera que el campo de la IA se fundó, como disciplina académica, durante la Conferencia de Darmouth de 1956. Otras definiciones, como la de M. Minsky [22] (9 68), establecen que la IA se trata de "la ciencia de hacer máquinas capaces de hacer las tareas que requieren inteligencia humana".

Años antes de que McCarthy introdujera el término de *Inteligencia Artificial*, Alan Turing [23] (1950) propuso una prueba con la que se podría saber si una computadora piensa igual que una persona. Dicho test, conocida como la prueba de Turing, ha servido de paradigma sobre lo que significa que una máquina tenga Inteligencia Artificial. La prueba de Turing consiste en que una persona *A* entabla una conversación con dos interlocutores desconocidos: uno de ellos se trata de una persona *B* mientras que el otro interlocutor sería la computadora. La computadora pasa la prueba si la persona A no puede diferenciar que uno de sus interlocutores se trata de una computadora. Esta prueba, con fuerte carácter conductual, ha sido considerada por algunos como superada por Google Duplex [24] en 2018.

Las ideas anteriormente expuestas sobre lo que es la IA se basan, sobre todo, en emular a la Inteligencia humana. Por otro lado, E. Rich (1983) [25-26], la IA es "el estudio de cómo hacer que las computadoras hagan cosas que, por el momento, las personas son mejores". Esta definición sobre lo que es la IA se complementa con lo que se conoce como *el efecto de la AI*, por L. Tesler, y que

establece que la Inteligencia es “aquello que las máquinas todavía no han hecho” [27].

1.3.1 APRENDIZAJE DE MÁQUINA

En los últimos años el término de *IA* se ha confundido con el de un subcampo de esta misma disciplina, que se conoce como el aprendizaje de máquina (traducción que comúnmente se le ha dado al término en inglés de *Machine Learning*). Jordan y Mitchell [28] afirman que el aprendizaje de máquina aborda el problema de cómo hacer computadoras que mejoren automáticamente con su experiencia. Además, estos autores justifican el éxito reciente del aprendizaje de máquina se debe al desarrollo de nuevos algoritmos, la disponibilidad creciente de datos en línea y el abaratamiento del cómputo.

El aprendizaje de máquina se puede definir como un subcampo [29] que abarca a todos los métodos que permiten que las computadoras aprendan de datos sin ser explíctamente programadas. En adición a esta definición anterior, Tom Mitchell [30] define que un algoritmo de aprendizaje de máquina se trata de

“Un programa de computadora que aprende de la experiencia E con cierto tipo T de tarea y una métrica de desempeño P si su desempeño en las tareas de tipo T, medidas con P, mejora con la experiencia E”.

En el aprendizaje de máquina se diferencian tres paradigmas de aprendizaje, los cuales son el aprendizaje supervisado, el aprendizaje no supervisado y el aprendizaje por refuerzos. Los primeros dos tipos de aprendizaje se implementan a partir de una colección de datos o muestras. Dichas muestras se encuentran caracterizadas por un vector de datos de entrada o rasgos. En el aprendizaje supervisado se dispone de un vector de salida con los que se quiere relacionar al vector de datos de entrada. El aprendizaje supervisado es útil para resolver problemas de clasificación o de regresión. En un problema de clasificación, las categorías del vector de salida están caracterizados de manera discreta. Como ejemplo de este tipo de problema se puede mencionar la colección de flores de iris [31]. En dicha colección hay 150 muestras. Cada muestra se encuentra caracterizada

con cuatro rasgos: el tamaño del pétalo, el ancho del pétalo, el tamaño del sépalo y el ancho del sépalo. Lo que se busca es mapear los rasgos anteriores con alguno de los tipos de flor de iris: *Virgina*, *Versicolor* o *Setosa*. En un problema de regresión, los valores del vector de salida son números reales continuos. El método de mínimos cuadrados, que se introduce en los primeros semestres de las carreras de física y química, se puede considerar como un algoritmo de aprendizaje de máquina.

En el aprendizaje de máquina no supervisado no se dispone de un vector de valores de salida. Lo que se busca con este paradigma del aprendizaje de máquina es analizar de grupos para formar cúmulos de datos (*clustering*), la reducción de dimensionalidad en la caracterización de los datos de una colección o la estimación de la densidad dentro del espacio del vector de datos de entrada.

De acuerdo con C. M. Bishop [32], el aprendizaje de máquina por refuerzos "se avoca a resolver el problema de encontrar acciones adecuadas que ejecutar a fin de maximizar una recompensa en una determinada situación. A diferencia del aprendizaje supervisado, no existe una colección de muestras, sino que el algoritmo *aprende* mediante un proceso de prueba y error. El aprendizaje de dicho algoritmo depende de la interacción con su ambiente. Un ejemplo de la aplicación de estos algoritmos se encuentra en los juegos como el backgammon o el ajedrez. En estas aplicaciones, las recompensas deben asignarse a todos los movimientos del juego hechos por el algoritmo, aun cuando algunos movimientos sean mejores que otros y cuando la recompensa mayor se la victoria en el juego. Este tipo de problemas del aprendizaje por refuerzos se dice que es de asignación de crédito".

Con base en lo anteriormente descrito sobre los problemas que se abordan con los diferentes paradigmas del aprendizaje de máquina, el trabajo que se desarrolló en esta tesis corresponde a un problema de clasificación binaria, en donde el algoritmo de aprendizaje de máquina que se utilizó, una red neuronal artificial, debe clasificar a un conjunto de compuestos cristalinos como perovskita (muestras verdaderas) o no perovskitas (muestras falsas).

1.3.2 REDES NEURONALES ARTIFICIALES

Las redes neuronales artificiales son un algoritmo del aprendizaje de máquina inspirado en el funcionamiento del sistema nervioso, el cual se basa en la transmisión de impulsos eléctricos (McCulloch & Pitts [33], 1943). En la Figura 1.3.1 se representa a una neurona. Las partes de una neurona son el cuerpo o soma, que es donde se asume que ocurre el procesamiento de los estímulos recibidos; las dendritas, con las que se reciben los estímulos de otras neuronas; y el axón, que es la parte que envía el impulso eléctrico a otras neuronas. El proceso mediante el cual se transmite el impulso nervioso se conoce como sinapsis. Se dice que la magnitud del impulso enviado por la célula presináptica debe superar un umbral a fin de activar a la célula postsináptica.

En la Figura 1.3.2 se representa a una red neuronal artificial de tipo *Feed – Forward* y *Full-Connected* (conectada completamente). El concepto de que una red neuronal sea de tipo *feed-forward* se refiere a que los nodos, que representan a las neuronas biológicas, sólo alimentan de información a los nodos ubicados la capa inmediata. Existen otro tipo de redes neuronales en donde los nodos, además de alimentar, reciben una especie de retroalimentación de los nodos ubicados en capas

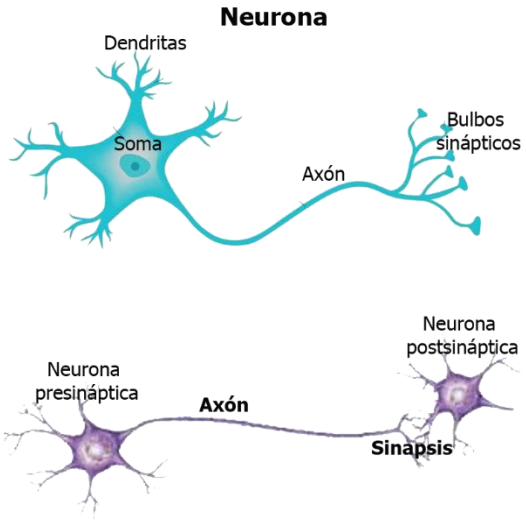


Figura 1.3.1: Arriba se ilustra las partes de una neurona biológica, que es la célula básica del sistema nervioso. Se asume que el sistema nervioso de cualquier organismo se basa en la transmisión de estímulos eléctricos. En una neurona, los estímulos son recibidos en las dendritas, los cuales son procesados en el cuerpo o *soma* de la neurona. Los estímulos procesados son enviados a otras neuronas a través de las terminales del axón. El mecanismo por el cual los estímulos eléctricos son trasmitidos se conoce como sinapsis (abajo)

posteriores. Este tipo de redes neuronales con retroalimentación se les conoce como recurrentes. Otra manera de referirse a las redes neuronales de tipo *feed-forward* es como Perceptrones Multicapa, que son una extensión del Perceptrón de F. Rosenblatt [34], 1957. Este tipo de red neuronal es la que se utilizó en este trabajo.

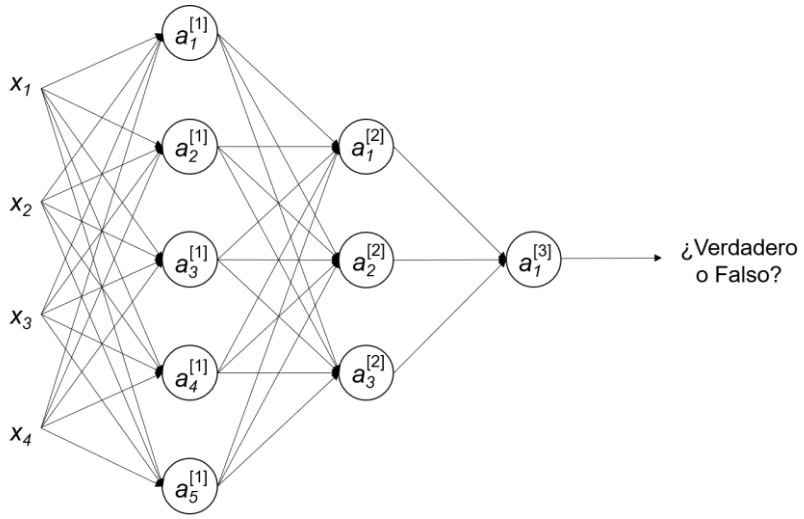


Figura 1.3.2: Representación de una red neuronal artificial de tipo *Feed-Forward* y totalmente conectada. Este tipo de red neuronal también se le conoce como Perceptrón multicapa. La red neuronal representada tiene una profundidad de tres capas de nodos ($L = 3$), los cuales simulan a las neuronas biológicas. El vector de datos de entrada de una muestra se señala a la izquierda como $x_1, \ldots, x_4$. A este vector de entrada se le puede encontrar en la literatura que conforma otra capa, que es la capa de entrada. La última capa de la red neuronal se le conoce como capa de salida. La información que se obtiene del único nodo de esa capa sirve para hacer la inferencia del tipo de muestra que se alimentó a la red neuronal. Las capas intermedias entre el vector de entrada y la capa de salida se les conoce como capas ocultas. Toda la información tanto del vector de entrada como de los nodos de las capas ocultas está conectada a través de una serie de pesos. Dichos pesos simulan la sinapsis que ocurre en las neuronas biológicas.

Los nodos de una red neuronal están organizados en capas. La red neuronal representada en la Figura 1.3.2 tiene una profundidad de tres capas ($L = 3$). A la izquierda, se representa el vector de datos de entrada con el que se caracteriza a una muestra. El vector de entrada representado en la Figura 1.3.2 consta de cuatro componentes (x_1, x_2, x_3, x_4). Posteriormente se tienen tres capas de nodos, los cuales simulan a las neuronas biológicas (la capa del vector de entrada se excluye para designar la profundidad de una red neuronal). La última capa de nodos recibe el nombre de capa de salida. Esta capa de salida consta de un solo nodo en la Figura 1.3.2. El valor calculado de ese nodo sirve para realizar la inferencia sobre

la clase de la muestra alimentada a la red neuronal. Entre el vector de entrada y la capa de salida se tiene dos capas de nodos, que reciben el nombre de capas ocultas. La cantidad de nodos que tienen las capas ocultas de la Figura 1.3.2 son cinco y tres, respectivamente.

Cada componente del vector de entrada está conectados a todos los nodos de la primera capa de la red neuronal mediante una serie de pesos. Estos pesos son el análogo a la sinapsis que ocurre en las neuronas biológicas. Esta conexión se repite entre los nodos de la primera capa de la red neuronal con la de la segunda, etcétera. La Figura 1.3.2 ha servido de ejemplo para explicar la arquitectura de una red neuronal para clasificación binaria, en donde hay un solo nodo en la capa de salida. No hay un límite establecido entre la cantidad de capas que una red neuronal debe tener. De hecho, el campo emergente de Aprendizaje Profundo [35] (*Deep Learning*, en inglés) se refiere, en parte, a los redes neuronales que tienen más de una capa ($L > 1$). Tampoco hay un límite máximo en la cantidad de nodos que debe tener una capa. En lo que concierne a la Figura 1.3.2 se puede afirmar que, entre el vector de datos de entrada y la primera capa, hay un mapeo de un espacio de dimensión cuatro al de uno de dimensión cinco; entre la primera capa y la segunda, el mapeo es de un espacio de dimensión cinco al de una dimensión de tres componentes; y entre la segunda capa y la última el mapeo es de un espacio tridimensional al de un espacio unidimensional.

A continuación, se procederá a describir el mecanismo [36] con el que una red neuronal, como la representada en la Figura 1.3.2, computa los valores de salida. Antes de comenzar, cabe mencionar que cada capa de la red neuronal es un vector columna con dimensión igual al número de nodos en ella. En la Figura 1.3.2, la primera capa es un vector de dimensión cinco; la segunda, de dimensión tres; y la última, un vector unidimensional. Cada componente del vector de entrada se indizará con la letra h; mientras que las componentes de los vectores en las capas de la red neuronal se indizarán, sucesivamente, con las letras i, j, y k.

En cada nodo de la primera capa de la red neuronal de la Figura 1.3.2 se efectúa, primeramente, una combinación lineal del vector de entrada de la muestra:

$$z_1^{[1]} = w_{11}^{[1]}x_1 + w_{12}^{[1]}x_2 + w_{13}^{[1]}x_3 + w_{14}^{[1]}x_4 + b_1^{[1]} \quad \text{(ec. 1)}$$

$$z_2^{[1]} = w_{21}^{[1]}x_1 + w_{22}^{[1]}x_2 + w_{23}^{[1]}x_3 + w_{24}^{[1]}x_4 + b_2^{[1]} \quad \text{(ec. 2)}$$

$$z_3^{[1]} = w_{31}^{[1]}x_1 + w_{32}^{[1]}x_2 + w_{33}^{[1]}x_3 + w_{34}^{[1]}x_4 + b_3^{[1]} \quad \text{(ec. 3)}$$

$$z_4^{[1]} = w_{41}^{[1]}x_1 + w_{42}^{[1]}x_2 + w_{43}^{[1]}x_3 + w_{44}^{[1]}x_4 + b_4^{[1]} \quad \text{(ec.4)}$$

$$z_5^{[1]} = w_{51}^{[1]}x_1 + w_{52}^{[1]}x_2 + w_{53}^{[1]}x_3 + w_{54}^{[1]}x_4 + b_5^{[1]} \quad \text{(ec.5)}$$

donde los parámetros $w_{ih}^{[1]}$ y $b_i^{[1]}$ reciben el nombre de pesos y términos de sesgo, respectivamente. En las ecuaciones anteriores el superíndice se refiere a la capa al que pertenecen los pesos y términos de sesgo. En efecto, las ecuaciones anteriormente plasmadas se pueden representar también de manera matricial como:

$$\vec{z}^{[1]} = \mathbf{W}^{[1]}\vec{x} + \vec{b}^{[1]} \quad \text{(ec. 6)}$$

donde $\vec{z}^{[1]}$, $\vec{x}$ y $\vec{b}^{[1]}$ se tratan de vectores columna:

$$\vec{z}^{[1]} = \begin{bmatrix} z_1^{[1]} \\ z_2^{[1]} \\ z_3^{[1]} \\ z_4^{[1]} \\ z_5^{[1]} \end{bmatrix} \qquad \vec{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} \qquad \vec{b}^{[1]} = \begin{bmatrix} b_1^{[1]} \\ b_2^{[1]} \\ b_3^{[1]} \\ b_4^{[1]} \\ b_5^{[1]} \end{bmatrix}$$

Y los pesos quedan representados como elementos de matriz:

$$\mathbf{W}^{[1]} = \begin{bmatrix} w_{11}^{[1]} & w_{12}^{[1]} & w_{13}^{[1]} & w_{14}^{[1]} \\ w_{21}^{[1]} & w_{22}^{[1]} & w_{23}^{[1]} & w_{24}^{[1]} \\ w_{31}^{[1]} & w_{32}^{[1]} & w_{33}^{[1]} & w_{34}^{[1]} \\ w_{41}^{[1]} & w_{42}^{[1]} & w_{43}^{[1]} & w_{44}^{[1]} \\ w_{51}^{[1]} & w_{52}^{[1]} & w_{53}^{[1]} & w_{54}^{[1]} \end{bmatrix}$$

El valor resultante de cada combinación lineal puede, posteriormente, transformarse mediante una función de activación $g(z_i^{[1]})$ que modele la no linealidad (Tabla 1.3.1).

$$a_1^{[1]} = g\left(z_1^{[1]}\right)$$

$$a_2^{[1]} = g(z_2^{[1]})$$

$$a_3^{[1]} = g(z_3^{[1]})$$

$$a_4^{[1]} = g\left(z_4^{[1]}\right)$$

$$a_5^{[1]} = g(z_5^{[1]})$$

Las funciones de activación que comúnmente se utilizan para modelar la no linealidad son la función sigmoide logística, la tangente hiperbólica y la unidad lineal rectificadora, que también se conoce como ReLU, por su acrónimo en inglés. En caso de que la función que se utilice sea la función lineal, $a_j^{[l]} = z_j^{[l]}$. Considerando el símil con las neuronas biológicas, el resultado de la función de activación es el estímulo que se envía a los nodos de la siguiente capa.

Los valores obtenidos después de la activación se pueden representar de manera compacta como vector columna:

$$\vec{a}^{[2]} = g\left(\vec{z}^{[2]}\right) \qquad \text{(ec. 7)}$$

$$\vec{a}^{[1]} = \begin{bmatrix} a_1^{[1]} \\ a_2^{[1]} \\ a_3^{[1]} \\ a_4^{[1]} \\ a_5^{[1]} \end{bmatrix}$$

El procedimiento anteriormente descrito, de combinar linealmente y activar la suma de dicha combinación, se repite en los nodos de las capas sucesivas. En la segunda capa oculta, por ejemplo, esto queda representado como:

$$\vec{z}^{[2]} = \mathbf{W}^{[2]}\vec{a}^{[1]} + \vec{b}^{[2]} \qquad \text{(ec. 8)}$$

$$\vec{a}^{[2]} = g\left(\vec{z}^{[2]}\right) \qquad \text{(ec. 9)}$$

Con:

$$\vec{z}^{[2]} = \begin{bmatrix} z_1^{[2]} \\ z_2^{[2]} \\ z_3^{[2]} \end{bmatrix} \qquad \vec{a}^{[2]} = \begin{bmatrix} a_1^{[2]} \\ a_2^{[2]} \\ a_3^{[2]} \end{bmatrix} \qquad \vec{b}^{[2]} = \begin{bmatrix} b_1^{[2]} \\ b_2^{[2]} \\ b_3^{[2]} \end{bmatrix}$$

Y

$$\mathbf{W}^{[2]} = \begin{bmatrix} w_{11}^{[2]} & w_{12}^{[2]} & w_{13}^{[2]} & w_{14}^{[2]} & w_{15}^{[2]} \\ w_{21}^{[2]} & w_{22}^{[2]} & w_{23}^{[2]} & w_{24}^{[2]} & w_{25}^{[2]} \\ w_{31}^{[2]} & w_{32}^{[2]} & w_{33}^{[2]} & w_{34}^{[2]} & w_{35}^{[2]} \end{bmatrix}$$

Y en la tercera capa (capa de salida):

$$\vec{z}^{[3]} = \mathbf{W}^{[3]}\vec{a}^{[2]} + \vec{b}^{[3]} \qquad \text{(ec. 10)}$$

$$\vec{a}^{[3]} = g(\vec{z}^{[3]}) \qquad \text{(ec. 11)}$$

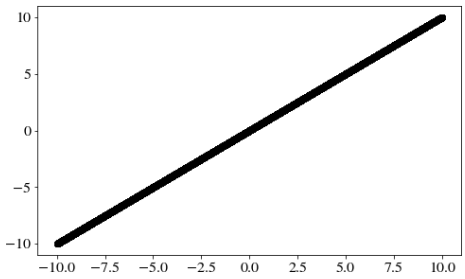
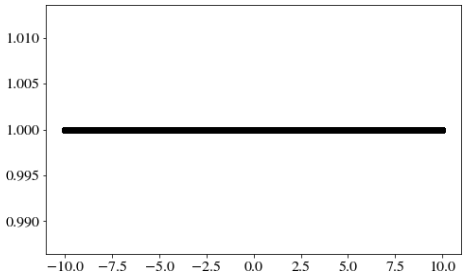
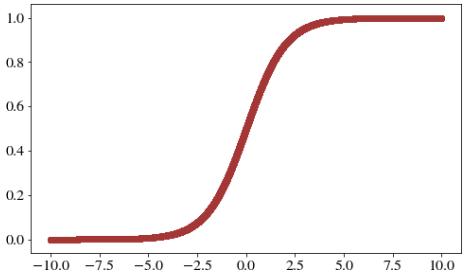
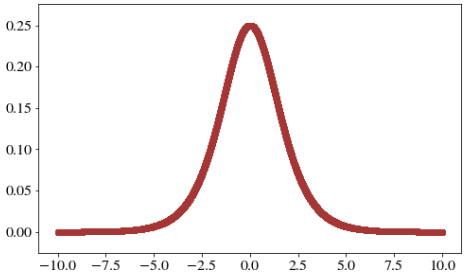
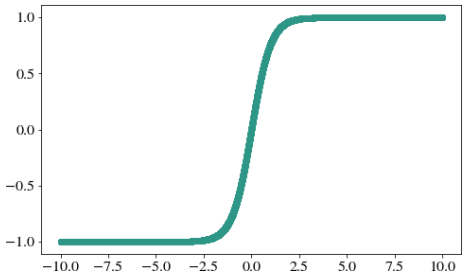
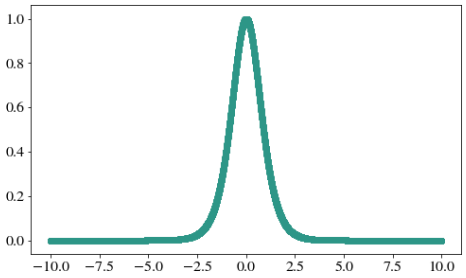
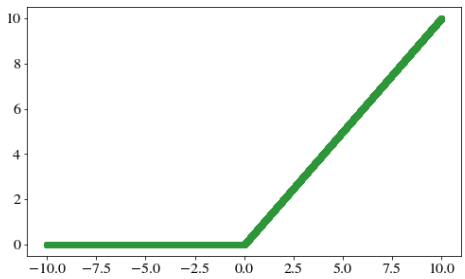
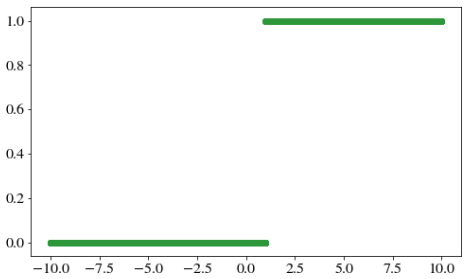
Con:

$$\vec{z}^{[3]} = \begin{bmatrix} z_1^{[3]} \end{bmatrix} \qquad \vec{a}^{[3]} = \begin{bmatrix} a_1^{[3]} \end{bmatrix} \qquad \vec{b}^{[3]} = \begin{bmatrix} b_1^{[3]} \end{bmatrix}$$

Y

$$\mathbf{W}^{[3]} = \begin{bmatrix} w_{11}^{[3]} & w_{12}^{[3]} & w_{13}^{[3]} \end{bmatrix}$$

Tabla 1.3.1: Lista de las funciones de activación de uso común en Aprendizaje Profundo. Las funciones de activación son utilizadas después de efectuar la combinación lineal en un nodo. A excepción de la función lineal, que es el equivalente a dejar sin cambios el resultado de la combinación, las funciones de activación se utilizan para modelar la no linealidad. En la Tabla se ilustra el perfil de estas funciones de activación sus derivadas.

Función de activación	Gráfica	Derivada
Lineal	 $g(x) = x$	 $g'(x) = 1$
Sigmoide logística	 $g(x) = \sigma(x)$ $\sigma(x) = \frac{1}{1+e^{-x}}$	 $g'(x) = \sigma'(x)$ $g'(x) = \sigma(x)(1-\sigma(x))$
Tangente hiperbólica	 $g(x) = \tanh(x)$	 $g'(x) = 1 - \tan^2(x)$
Unidad Lineal Rectificadora (ReLU)	 $g(x) = \begin{cases} x \text{ si } x > 0 \\ 0 \; x \leq 0 \end{cases}$	 $g(x) = \begin{cases} 1 \text{ si } x > 0 \\ 0 \; x \leq 0 \end{cases}$

El vector de salida $\vec{a}^{[3]}$ tiene una sola componente (un nodo en la capa oculta). El valor de dicha componente deberá ser cercano al valor real de la muestra que se alimentó a la red neuronal artificial una vez que ésta haya aprendido correctamente: Si la muestra que se alimentó a la red neuronal representa a una muestra de la clase verdadero, el valor de salida será cercano a uno; de lo contrario, el valor será cercano a cero. Cabe mencionar que, en un problema de clasificación binaria, la función de activación utilizada en el vector de salida es una sigmoide logística. Las razones para utilizar la función sigmoide en la última capa se comprenden al revisar la ecuación de la función que sirve para optimizar los parámetros de la red neuronal.

El mecanismo anteriormente descrito de cómo una red neuronal calcula el vector de salida es el mismo que se utiliza para entrenarla. Los pesos son inicializados de manera aleatoria, mientras que los términos de sesgo típicamente se inicializan como cero. Mediante un proceso iterativo, los parámetros son optimizados. Una vez concluida la optimización, que en el lenguaje de aprendizaje de máquina se conoce como entrenamiento, se dice que la red neuronal aprendió.

En problemas de clasificación binaria se utiliza la función de entropía cruzada como función de costo (o función objetivo) en la optimización de los parámetros de la red neuronal. Para una muestra, la función de entropía cruzada es la siguiente

$$\mathrm{J} = -\left[\hat{\mathrm{y}}\log_2 \mathrm{a}_{\mathrm{k}}^{[\mathrm{L}]} + (1-\hat{\mathrm{y}})\log_2(1-\mathrm{a}_{\mathrm{k}}^{[\mathrm{L}]})\right] \qquad \text{(ec.12)}$$

Donde $\hat{y}$ es el valor de salida real de la muestra (0 o 1, en caso de que pertenezca a la clase falso o verdadero), $a_k^{[L]}$ es el valor de salida calculado (del nodo *k*) por la red neuronal. En el caso del ejemplo presentado en la Figura 1.3.2, *L* = 3. Esta función de costo tiene como característica ser siempre positiva y cercana a cero cuando el valor de salida de la red neuronal es cercano al valor real. Nótese que, según la información sobre la muestra alimentada a la red neuronal, sólo uno de los términos contribuye en la función de entropía: si la muestra es de tipo verdadero, el término que es diferente de cero es $-\hat{y}\log_2 a_k^{[L]}$, de lo contrario, es

$-(1-\hat{y})\log_2(1-a_k^{[L]})$. Esta función es, además, convexa, por lo que permite la optimización de los parámetros de la red neuronal.

Uno de los métodos más utilizados en la optimización de parámetros de una red neuronal es el del gradiente descendente [37 – 38]. De hecho, las librerías de Python que permiten implementar rápidamente una red neuronal artificial basan la optimización de sus parámetros en alguna variante del descenso del gradiente.

Con el descenso del gradiente, los parámetros de la red neuronal, como la de la Figura 1.3.2, se optimizan de la siguiente manera:

$$\mathrm{w}_{\mathrm{mn}}^{[\mathrm{l}]} := \mathrm{w}_{\mathrm{mn}}^{[\mathrm{l}]} - \alpha\frac{\partial \mathrm{J}}{\partial \mathrm{w}_{\mathrm{mn}}^{[\mathrm{l}]}} \qquad \text{(ec. 13)}$$

$$\mathrm{b}_{\mathrm{m}}^{[\mathrm{l}]} := \mathrm{b}_{\mathrm{m}}^{[\mathrm{l}]} - \alpha\frac{\partial \mathrm{J}}{\partial \mathrm{b}_{\mathrm{m}}^{[\mathrm{l}]}} \qquad \text{(ec.14)}$$

donde $\mathrm{w}_{\mathrm{mn}}^{[\mathrm{l}]}$ es el peso que conecta al nodo *m* de una capa *l* con el nodo *n* de la capa anterior *l-1*, y que se representa como elemento de matriz *mn*; y $\mathrm{b}_{\mathrm{m}}^{[\mathrm{l}]}$ es el término de sesgo *m* de la capa *l* de una red neuronal. Además, α es un parámetro ajeno a la red neuronal (hiperparámetro) que se conoce como velocidad de aprendizaje. Esta velocidad de aprendizaje influye en el número de iteraciones necesarias para alcanzar el aprendizaje de una red neuronal. El valor óptimo en la velocidad de aprendizaje se encuentra de manera heurística. Un valor pequeño en la velocidad de aprendizaje necesitará de varias iteraciones para terminar el entrenamiento de la red neuronal; mientras que valores altos de dicha velocidad pueden representar problemas para alcanzar el mínimo en la función de costo.

Cuando la red neuronal haya sido optimizada (llegue a un mínimo), los gradientes $\frac{\partial \mathrm{J}}{\partial \mathrm{w}_{\mathrm{mn}}^{[\mathrm{l}]}}$ y $\frac{\partial \mathrm{J}}{\partial \mathrm{b}_{\mathrm{m}}^{[\mathrm{l}]}}$ serán cercanos a cero.

Una manera eficiente de calcular los gradientes de cada peso (elemento de matriz) y término de sesgo es a través de la retropropagación [39 – 40] (*backpropagation*, en inglés). El nombre de retropropagación se debe a que el cálculo de los gradientes de los parámetros de la red neuronal se hace desde la

última capa a la primera, además de que el error de la capa de salida se propaga hacia las primeras capas de la red neuronal. Aunque el método de retropropagación se propuso originalmente considerando el error cuadrático como función de costo, éste se utiliza también para problemas de clasificación. El error cuadrático es una función de costo adecuada para un problema de regresión. El error cuadrático de una muestra se define como:

$$J = \frac{1}{2}\left(\hat{y} - a_k^{[L]}\right)^2 \qquad \text{(ec. 15)}$$

La primera deriva de la función anterior respecto al valor de salida de la red neuronal en el nodo *k*, $a_k^{[L]}$, es:

$$\frac{\partial J}{\partial a_k^{[L]}} = \hat{y} - a_k^{[L]} \qquad \text{(ec. 16)}$$

$$\delta_k^{[L]} = \hat{y} - a_k^{[L]} \qquad \text{(ec. 17)}$$

La diferencia entre el valor de salida real y el valor calculado por la red neuronal se ha representado como $\delta_k^{[L]}$ y es el error. El punto de la retropropagación es relacionar ese error con los gradientes de los pesos y términos de sesgo de todas las capas *l* anteriores (ocultas) de la red neuronal. Esto se consigue a través de la regla de la cadena. Por ejemplo, para los pesos y términos de sesgo de los parámetros de la última capa, capa de salida, de la red neuronal de la Figura 1.3.2 se tiene que:

$$\frac{\partial J}{\partial w_{kj}^{[3]}} = \frac{\partial J}{\partial a_k^{[3]}} \cdot \frac{\partial a_k^{[3]}}{\partial z_k^{[3]}} \cdot \frac{\partial z_k^{[3]}}{\partial w_{kj}^{[3]}} \qquad \text{(ec.18)}$$

$$\frac{\partial J}{\partial b_k^{[3]}} = \frac{\partial J}{\partial a_k^{[3]}} \cdot \frac{\partial a_k^{[3]}}{\partial z_k^{[3]}} \cdot \frac{\partial z_k^{[3]}}{\partial b_k^{[3]}} \qquad \text{(ec. 19)}$$

La derivada $\frac{\partial a_k^{[3]}}{\partial z_k^{[3]}}$ es, esencialmente, la derivada de la función de activación (Tabla 1.3.1). En el caso de una función sigmoide logística, esta derivada es:

$$\sigma'\left(z_k^{[3]}\right) = \sigma\left(z_k^{[3]}\right) \cdot \left(1 - \sigma\left(z_k^{[3]}\right)\right) \qquad \text{(ec. 20)}$$

Mientras que de la ecuación 10 se obtienen:

$$\frac{\partial z_k^{[3]}}{\partial w_{kj}^{[3]}} = a_j^{[2]} \qquad \text{(ec. 21)}$$

$$\frac{\partial z_k^{[3]}}{\partial b_k^{[3]}} = 1 \qquad \text{(ec. 22)}$$

Por lo tanto, los gradientes de los pesos y términos de sesgo de la capa de salida quedan como:

$$\frac{\partial J}{\partial w_{kj}^{[3]}} = \delta_k^{[L]} \cdot \sigma'\left(z_k^{[3]}\right) \cdot a_j^{[2]} \qquad \text{(ec. 23)}$$

$$\frac{\partial J}{\partial b_k^{[3]}} = \delta_k^{[L]} \cdot \sigma'\left(z_k^{[3]}\right) \qquad \text{(ec. 24)}$$

El cálculo de los gradientes de los pesos y términos de sesgo de las capas ocultas es similar a las ecuaciones 18 y 19:

$$\frac{\partial J}{\partial w_{mn}^{[l]}} = \frac{\partial J}{\partial a_m^{[l]}} \cdot \frac{\partial a_m^{[l]}}{\partial z_m^{[l]}} \cdot \frac{\partial z_m^{l}}{\partial w_{mn}^{[l]}} \qquad \text{(ec. 25)}$$

$$\frac{\partial J}{\partial b_m^{[l]}} = \frac{\partial J}{\partial a_m^{[l]}} \cdot \frac{\partial a_m^{[l]}}{\partial z_m^{[l]}} \cdot \frac{\partial z_m^{[l]}}{\partial b_m^{[l]}} \qquad \text{(ec. 26)}$$

Sin embargo, queda pendiente cómo calcular el error del *n* – ésimo nodo de la *l* – ésima capa:

$$\frac{\partial J}{\partial a_n^{[l]}} = \delta_n^{[l]} \qquad \text{(ec. 27)}$$

A diferencia de los nodos de la última capa, el error en los nodos de las capas ocultas no es tan directo como la diferencia entre el valor real y el calculado. Por regla de la cadena se tiene, por ejemplo, para los nodos de la penúltima de la red neuronal de la Figura 1.3.2 que:

$$\frac{\partial J}{\partial a_j^{[2]}} = \frac{\partial J}{\partial a_k^{[3]}} \cdot \frac{\partial a_k^{[3]}}{\partial z_k^{[3]}} \cdot \frac{\partial z_k^{[3]}}{\partial a_j^{[2]}} \qquad \text{(ec. 28)}$$

Del lado derecha de la ecuación anterior, se tiene que el primer factor es el error en el nodo *k* de la última capa, el segundo factor es la derivada de la función

de activación del nodo *k* de la última capa y el tercer factor es el elemento de matriz que relaciona a los nodos k y j de la última y penúltima capa, respectivamente.

$$\frac{\partial z_k^{[3]}}{\partial a_j^{[2]}} = w_{kj}{}^{[3]} \qquad \text{(ec. 29)}$$

Así que, el error en el nodo *j* de la segunda capa de la red neuronal de la Figura 1.3.2 es:

$$\frac{\partial J}{\partial a_j^{[2]}} = \delta_j^{[2]} = \delta_k^{[3]} \cdot \frac{\partial a_k^{[3]}}{\partial z_k^{[3]}} \cdot w_{kj}{}^{[3]} \qquad \text{(ec. 30)}$$

En general, para los nodos de cualquier capa se tiene que:

$$\delta_n^{[l]} = \delta_m^{[l+1]} \cdot \frac{\partial a_m^{[l+1]}}{\partial z_m^{[l+1]}} \cdot w_{mn}{}^{[l+1]} \qquad \text{(ec. 31)}$$

De esta manera, en la Tabla 1.3.2 se recogen las cuatro ecuaciones de la retropropagación que permiten calcular los gradientes necesarios para implementar la optimización por descenso del gradiente. En la última columna de la Tabla 1.3.2 se representan las mismas ecuaciones de forma vectorizada. Dicha representación ayuda a que el código sea más eficiente en términos de cómputo y tiempo.

Tabla 1.3.2: Ecuaciones de la retropropagación (*backpropagation*, en inglés). El objetivo de la retropropagación es calcular los gradientes de los parámetros de la red neuronal (pesos y términos de sesgo) a fin de implementar la optimización por descenso del gradiente. En la Tabla, la primera columna recoge las ecuaciones que se obtuvieron al considerar a los pesos y términos de sesgo por separado. No obstante, el código es más eficiente al implementar una versión vectorizada de la retropropagación, que se muestra en la tercera columna.

Ecuación	Descripción	Forma vectorizada
$\delta_m^{[L]} = y_m - a_m^{[L]}$	Error de salida en el valor real y el computado por la red neuronal.	$\vec{\delta}^{[L]} = \vec{y} \quad - \vec{a}^{[L]}$
$\delta_n^{[l]} = \delta_m^{[l+1]} \cdot g'(z_m^{[l+1]}) \cdot w_{mn}{}^{[l+1]}$	Error en el *m* – ésimo nodo de la *l* – ésima capa. Esta ecuación de utiliza en las capas anteriores a la de salida.	$\vec{\delta}^{[l]} = \mathbf{W}^{[l+1]\,T} \cdot (\vec{\delta}^{[l+1]} \odot \vec{g}'(z^{[l+1]}))$
$\frac{\partial J}{\partial w_{mn}^{[l]}} = \delta_m^{[l]} \cdot g'(z_m^{[l]}) \cdot a_n^{[l-1]}$	Gradiente del *mn* – ésimo elemento de matriz (peso) de la *l* – ésima capa. Válido para cualquier capa.	$\frac{\partial J}{\partial \mathbf{W}^{[l]}} = \left(\vec{\delta}^{[l]} \odot \vec{g}'\left(z^{[l]}\right)\right) \cdot \vec{a}^{[l-1]\,T}$
$\frac{\partial J}{\partial b_m^{[l]}} = \delta_m^{[l]} \cdot g'(z_m^{[l]})$	Gradiente del *m* – ésimo término de sesgo de la *l* – ésima capa. Válido para cualquier capa.	$\frac{\partial J}{\partial \vec{b}^{[l]}} = \vec{\delta}^{[l]} \odot \vec{g}'(z^{[l]})$

En las ecuaciones 12 y 15 se sugiere que la optimización de los parámetros de la red neuronal se hace cada vez que se alimenta una muestra al algoritmo. Dicha forma de optimizar se conoce como descenso estocástico del gradiente. Esta manera de implementar el descenso del gradiente resulta conveniente cuando el tamaño de muestras en la colección es grande (centenas de miles) y se busca una optimización rápida. Otra forma de implementar el descenso del gradiente es por bloques pequeños de muestras, que en inglés se le conoce como *mini-batch gradient descent* [38]. Considérese que el conjunto de muestras para entrenar contiene 3200. Esas 3200 muestras se reparten en bloques de veinte muestras, por ejemplo, de manera que se tendrán 160 bloques pequeños. Al implementar una iteración en la red neuronal se barre el conjunto de 3200 muestras con los 160 bloques que se crearon. En el lenguaje de aprendizaje de máquina se suele llamar a las iteraciones *épocas*. Por lo tanto, la optimización de los parámetros en una época habrá ocurrido 160 veces. La función de error que se utiliza es el promedio de la entropía binaria en un bloque pequeño.

El caso completamente opuesto al descenso estocástico del gradiente es cuando la optimización de los parámetros de la red neuronal se hace utilizando todas las muestras disponibles para entrenar. Dicha implementación se conoce como descenso del gradiente por bloque y es recomendado cuando el conjunto de muestras para entrenar es pequeño (hasta 1000 muestras, aproximadamente).

1.3.4 TRABAJOS DE *IA* RELACIONADOS CON ESTA TESIS

Después de la introducción anterior sobre IA, aprendizaje de máquinas y redes neuronales, merece la pena revisar algunos problemas que se han abordado en el campo de la química, la física y la ciencia de materiales. La aplicación del aprendizaje de máquina en el área científica ha sido, sobre todo, en problemas de aprendizaje supervisado. Parte de la justificación de revisar los problemas abordados en ciencia tiene que ver con que una caracterización adecuada de los datos de entrada de las muestras influirá en el éxito de cualquier algoritmo de aprendizaje de máquina, como es el caso de las redes neuronales. Las redes neuronales son una herramienta que dota de autonomía a un programa de computadora en la búsqueda de correlación entre datos.

Lorenz, Behler y Artrith [41 – 43] han utilizado las energías obtenidas de cálculos con teoría de funcionales de la densidad a fin de crear una red neuronal que reconstruya la superficie de energía potencial (*PES*, por su acrónimo en inglés). En los sistemas en donde se ha probado este enfoque con redes neuronales es en la disociación de moléculas diatómicas sobre superficies metálicas y en las fases anatasa, brookita y rutilo del óxido de titanio (IV). La caracterización del sistema de partículas se basa en la construcción de funciones de simetría radial G_i^1 y angular G_i^2 de cada átomo.

$$G_i^1 = \sum_{j \neq i} e^{-\eta\left(R_{ij} - R_s\right)^2} f_c(R_{ij})$$

$$G_i^2 = 2^{1-\zeta} \sum_{j,k \neq i} \left(1 + \lambda \cos\theta_{ijk}\right)^{\zeta} e^{\eta\left(R_{ij}^2 + R_{ik}^2 + R_{jk}^2\right)} f_c(R_{ij}) f_c(R_{ik}) f_c(R_{jk})$$

En las ecuaciones anteriores, η, R_s, ζ son parámetros que se ajustan y que deben ser encontrados para cada sistema. Por otro lado, f_c define una función de corte con radio $R_c = 6$ Å:

$$f_c(R_{ij}) = \begin{cases} \frac{1}{2}\cos\left(\frac{\pi R_{ij}}{R_c} + 1\right) para\ R_{ij} \leq R_c \\ 0\ si\ R_{ij} > R_c \end{cases}$$

Los datos de entrada son alimentados a una red neuronal para cada átomo. Los nodos de salida de cada red neuronal por átomo se conectan a una última red neuronal que es la que calcula la energía del sistema de partículas. El error cuadrático promedio en la energía que se obtiene con este tipo de redes neuronales es del orden de 5 – 6 meV (0.12 – 0.14 kcal/mol). En adición, una de las ventajas que se tienen al usar redes neuronales es que el tiempo de cómputo depende de la cantidad de átomos.

Hansen *et. al.* [44] han utilizado diversos algoritmos de aprendizaje de máquina, entre ellos los Perceptrones multicapa, para determinar las energías de atomización de moléculas con hasta 23 átomos. La colección de muestras utilizada por Hansen *et. al.* constó de 7165 moléculas de átomos de los dos primeros periodos de la Tabla periódica. La caracterización de las moléculas se consiguió a través de la matriz de Coulomb. La matriz de Coulomb [45] (Rupp, 2012) de un compuesto se define como:

$$C_{ij} = \begin{cases} 0.5\ Z_i^{0.4}\ \forall i = j \\ \frac{Z_i Z_j}{|R_i - R_j|}\ \forall i\ \neq j \end{cases}$$

Con la matriz de Coulomb se obtiene una representación en dos dimensiones de una estructura tridimensional, como lo es una molécula. Z_i y R_i designan la carga nuclear y la coordenada atómica del átomo *i*. La caracterización de una molécula a través de la matriz de Coulomb tiene dos fuertes desventajas: moléculas con diferente cantidad de átomos dan lugar a diferentes tamaños de matriz, y que no existe una manera concreta de ordenar los átomos a lo largo de las filas y columnas de una matriz. El primer problema se resuelve homogenizando el tamaño de la

matriz mediante la adición de columnas y filas con valores nulos (*padding)*. Para el segundo problema, Hansen *et. al.* utilizaron tres enfoques diferentes: 1) la obtención de los valores propios de la matriz; 2) El ordenamiento en forma decreciente de los valores C_{ij} a lo largo de las filas y columnas; y 3) La construcción de un tensor en el que se apilan diferentes matrices de Coloumb del mismo sistema. La diferencia entre las matrices de un mismo tensor está en la permutación del orden entre columnas y filas. El mejor error cuadrático promedio obtenido por Hansen *et. al.,* utilizando Perceptrones multicapa, fue de 5.96 kcal mol^{-1}. Este resultado se obtuvo mediante la caracterización por tensores de mil matrices de Coulomb apiladas.

La caracterización de las muestras de una colección que se deriva de la matriz de Coulomb ha sido también utilizada por Faber *et. al.* [46] para inferir las energías de formación de compuestos sólidos mediante regresión Ridge utilizando Kernel [47]. La colección de muestras utilizada por Faber *et. al*. constó de 3898 sistemas sólidos que obtuvieron del sitio web *Materials Project* [48]. La construcción de los datos de entrada de las muestras de la colección se consiguió mediante tres modificaciones de la matriz de Coulomb: 1) La matriz de suma de Ewald, 2) La matriz extendida de Coulomb y 3) La matriz de seno. Los elementos de la matriz de suma de Ewald corresponden a un par de átomos en la celda primitiva. Los elementos de dicha matriz se calculan mediante la siguiente fórmula:

$$x_{ij} = x_{ij}^{sr} + x_{ij}^{lr} + x_{ij}^{0}$$

Los términos x_{ij}^{sr} , x_{ij}^{lr} corresponden a la interacción de un par de átomos *i* y *j* de corto y largo alcance, respectivamente; mientras que x_{ij}^{0} sirve como un término de corrección que descibe la interacción de los *cores* átomicos. Los términos que describen la interacción en corto y largo alcance entre dos átomos $i \neq j$ se definen como:

$$x_{ij}^{sr} = Z_i Z_j \sum_{L} \frac{erfc\left(a\left\|r_i - r_j + L\right\|_2\right)}{\left\|r_i - r_j + L\right\|_2}$$

$$x_{ij}^{lr} = \frac{Z_i Z_j}{\pi V} \sum_{G} \frac{1}{\|G\|_2^2} \frac{e^{-\|G\|_2^2}}{(2a)^2} \cos(G \cdot (r_i - r_j))$$

L y G corresponden a vectores en los espacios real y recíprocos y V corresponde al volumen de la celda primitiva. Es importante la definición de un radio de corte L_{max} y G_{max}. Faber *et. al.* reportan que el parámetro a afecta en la velocidad de convergencia de las sumas de las interacciones de corto y largo alcance. Dicho parámetro se define como:

$$a = \sqrt{\pi} \left(\frac{0.01M}{V}\right)^{\frac{1}{6}}$$

Donde M es el número de átomos en la celda primitiva. El término correctivo se define como:

$$X_{ij}^{0} = \begin{cases} -\left(Z_i^2 + Z_j^2\right)\left(\frac{a}{\sqrt{\pi}} + \frac{\pi}{2Va^2}\right) \; si \; i \neq j \\ -\left(Z_i^2\right)\left(\frac{a}{\sqrt{\pi}} + \frac{\pi}{2Va^2}\right) \; si \; i = j \end{cases}$$

La matriz extendida de Coulomb es similar a la matriz de Coulomb original salvo que: 1) sus dimensiones son de *M* por *MN*, donde *M* corresponden a los átomos en la celda primitiva y *N* corresponde al número de celdas vecinas y; 2) El potencial utilizado tiene la forma $Z_i Z_j e^{\|r_i - r_j\|}$.

La matriz de seno difiere de la matriz de Coulomb original en que los elementos de matriz que no pertenecen a la diagonal se calculan como:

$$\frac{Z_i Z_j}{\left\|\mathbf{B} \cdot \sum_{k=\{x,y,z\}} \widehat{e_k} \sin^2[\pi \hat{e}_k \mathbf{B}^{-1} \cdot (r_i - r_j)\right\|_2}$$

Donde $\mathbf{B}$ es la matriz formada por los vectores base de la celda. Los errores absolutos promedio obtenidos en por Faber *et. al.* fueron 0.49 eV atom^{-1} con la matriz de suma de Ewald, 0.64 eV atom^{-1} con la matriz extendida de Coulomb y 0.37 eV atom^{-1} con la matriz de seno. Los autores sugieren que, aun cuando con la matriz de seno se hayan obtenido mejores resultados, el desempeño obtenido es

prácticamente similar. Sin embargo, el cálculo de la matriz de seno es más directo que las otras dos representaciones utilizadas por Faber *et. al.*

Schmidt y coautores [49] utilizaron diferentes algoritmos de aprendizaje de máquina, entre ellos Perceptrones multicapa, para determina la estabilidad termodinámica de sólidos tipo perovskita. La caracterización de las muestras que Schmidt *et. al.* implementaron se basó en un vector de entrada con información para cada elemento de la fórmula como el número de electrones de valencia, la electronegatividad, sus estados de oxidación más comunes, la masa atómica, el punto de fusión, entre otros. Con una colección de 249692 compuestos calculados vía DFT, y que se encuentra disponible en *Materials Project*, ellos obtienen un error promedio absoluto de 121 meV $atom^{-1}$ en la distancia al envolvente convexo [50]. Todas las muestras utilizadas poseían la estructura perovskita cúbica.

Fedorov y Shamanaev [51] han utilizado las redes neuronales de tipo Perceptron multicapa para determinar la capacidad molar calorífica, la entropía molar estándar y la energía de red de compuestos cristalinos inorgánicos. La colección de muestras utilizadas para entrenar a las redes neuronales fue tomada de la base de datos de estructuras inorgánicas cristalinas (*ICSD*, por su acrónimo en inglés) y de la base de datos libre *COD (Crystallography Open Database*). El número de compuestos máximo utilizado en su desarrollo de redes neuronales fue de 168. La caracterización de las muestras de la colección se consigue por la caracterización de los centros topológicos [52] del compuesto cristalino (Thimm, 2009), para los cuales hay una red neuronal independiente. Cada centro topológico se caracteriza mediante un vector de entrada con información sobre la electronegatividad, el estado de oxidación, la masa molar, el radio covalente y la distancia entre un centro topológico y sus vecinos. Las salidas de las redes neuronales de cada centro topológico, que es un vector de una componente, son utilizadas como vector de entrada de una red neuronal que computa los valores termodinámicos mencionados. El error porcentual absoluto promedio que se consigue mediante redes neuronales y la caracterización implementada por Fedorov y Shamanaev fue menor al 8%. El trabajo desarrollado en la presente tesis

es similar al publicado por Fedorov y Shamanaev, sólo que aquí se utilizaron los sitios de simetría de Wyckoff de los grupos espaciales cristalinos para definir los rasgos que caracterizan a cada compuesto.

Ye *et. al* [53]*.* publicaron en 2018 sobre la determinación de energías de formación de compuestos tipo granate y perovskita mediante redes neuronales de tipo Perceptrón multicapa. Para el desarrollo de su algoritmo de aprendizaje de máquina utilizaron 542 compuestos tipo perovskita ABO_3 y 1407 compuestos tipo granate $C_3A_2D_3O_{12}$, que consistieron en compuestos puros y soluciones sólidas. La caracterización de los compuestos cristalinos implementada por Ye y coautores se basó en la caracterización de los sitios de Wyckoff de sus cationes. Los sitios de Wyckoff de los cationes fueron caracterizados mediante los radios iónicos de Shannon y las electronegatividades de Pauling promedio. A diferencia de las redes neuronales de Fedorov y Behler, en donde existe una red neuronal para cada centro topológico o átomo, todos los rasgos de los sitios de Wyckoff se concatenaron en un vector de entrada único a una red neuronal. En concreto, se obtiene un vector de entrada de seis y ocho componentes para el caso de las muestras tipo perovskita y granate, respectivamente. Ye y coautores obtuvieron un error absoluto promedio de 7 – 10 meV $atom^{-1}$ y 20 – 31 meV $atom^{-1}$ para los compuestos tipo granate y perovskita.

Javed [54] y Majid [55 – 56] han utilizado diversos algoritmos de aprendizaje de máquina, entre ellos las redes neuronales de tipo perceptrón multicapa, para determinar los parámetros de red de compuestos cristalinos con estructura tipo perovskita. En su colección han utilizado compuestos tipo perovskita con simetría ortorrómbica, así como con simetría monoclínica y cúbica. El error porcentual absoluto que Javed y Majid obtienen es menor al 1%. La caracterización que los autores mencionados implementaron es mediante la descripción de los radios iónicos y electronegatividades de los cationes, así como el estado de oxidación del catión más voluminoso.

Pilania y coautores [57-58] han utilizado máquinas de vectores soportados [59-60] y árboles de decisión con potenciación del gradiente [61] para inferir nuevos

compuestos con estructura perovskita. Pilania y coautores utilizaron 185 compuestos experimentales de tipo ABX_3 [62] y 354 compuestos de tipo ABO_3. La caracterización de los compuestos de las colecciones usadas se consiguió a través de los radios iónicos de los elementos, el factor de tolerancia de Goldschmidt y el factor octaédrico (sección 1.2.1 de esta tesis), los radios de valencia del enlace, cocientes de suma de los radios de los orbitales *s* y *p* y diferencias de electronegatividades. Las precisiones en la clasificación global (*accuracy*, en inglés) fueron de 92.1 y 95.1 %.

Isayev y coautores [63] han caracterizado los materiales del repositorio AFLOW [64] mediante lo que ellos mismos han denominado como Fragmentos de materiales etiquetados con propiedades (*Property-Labelled Materials Fragments, PLMF*). La teselación de Voronoi [65] es la esencia de esta caracterización, la cual se efectúa en una partición de la estructura cristalina. Con la teselación de Voronoi se establece qué átomos están conectados. Para esto, es necesario que los átomos compartan una cara de Voronoi y que la distancia interatómica sea menor a la suma de radios covalentes de Cordero [66]. Mediante la teselación, se obtiene un grafo y una matriz de adyacencia, $\mathbf{A}$. Los rasgos se obtienen mediante las fórmulas:

$$T^E = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} |q_i - q_j| M_{ij}$$

$$T^E_{bond} = \sum_{\{i,j\} \in enlaces} |q_i - q_j| M_{ij}$$

Donde $\mathbf{M}$ es la matriz Galvez, de dimensión *nxn*, donde *n* es el número de átomos en la celda unitaria. La matriz Galvez se obtiene del producto entre elementos de matriz $\mathbf{A} \cdot \mathbf{D}$, donde $\mathbf{D}$ es la matriz recíproca de cuadrados de distancia ($\mathrm{D_{ij}} = 1/r_{i,j}^2$). Las variables q_iy q_j son propiedades de un átomo como el número de electrones de valencia, la masa atómica, la electronegatividad, la capacidad calorífica, la dureza química, entre otras. Con esta aproximación de Fragmentos de materiales etiquetados con propiedades se obtiene un vector de datos de entrada

de 2494 rasgos, después de filtrar los rasgos con una correlación $r^2 > 0.95$ y con una varianza menor al 0.001.

Tanto Isayev y coautores como Xie y Grossman [67] han utilizado la caracterización por fragmentos de materiales etiquetados con propiedades para clasificar entre metales y aislantes, determinar energías absolutas, brechas enérgeticas, módulos de compresibilidad y corte, razones de Poisson, temperatura de Debye y capacidades caloríficas. Isayev y coautores han implementado árboles de decisión con potenciación del gradiente, mientras que Xie y Grossman utilizaron redes neuronales del tipo Convolucional. Ambos grupos utilizaron conjuntos de muestras superiores a 46,000 compuestos. Los resultados obtenidos con esta caracterización son prometedores y pueden consultarse en las referencias 63 y 67.

2. PLANTEAMIENTO DEL PROBLEMA

En los últimos años han aparecido iniciativas como *Materials Genome Project* [68], *Mission Innovation* [69], el proyecto de energía limpia de la Universidad de Harvard [70] y el plan alemán de innovación en materiales para la industria y la sociedad [71], WING, han destacado el uso de herramientas computacionales en la búsqueda de nuevos compuestos con el fin de acortar los tiempos entre descubrimiento y comercialización de un material. En adición, la aceleración de dicho proceso viene acompañado de un cambio de paradigma en la forma de descubrir materiales, lo que implica pasar de un proceso experimental de prueba y error a lo que se ha llamado “diseño guiado por computadoras”. En este último enfoque, la viabilidad de obtener un material es evaluada computacionalmente previo al experimento, lo que supone un ahorro en costos.

Una de las principales desventajas de los cálculos mecanocuánticos es que el tiempo y el costo computacional se incrementa en sistemas tales como cristales o macromoléculas, debido a que estos métodos dependen principalmente de la cantidad de electrones. Una forma de resolver este inconveniente es mediante el uso de la información acumulada en bases de datos con algoritmos de Inteligencia Artificial. En particular, las redes neuronales artificiales son un algoritmo versátil que se puede adaptar en cualquier tipo de problema de aprendizaje supervisado o no supervisado.

El uso de las técnicas de Inteligencia Artificial en el campo científico es relativamente nuevo, razón por la que todavía no existe una metodología general que permita relacionar un tipo de estructura cristalina con su información cristalográfica. En adición, las metodologías propuestas no permiten caracterizar a un tipo de estructura cristalina en sus diferentes grupos espaciales. Dicha caracterización sería deseable que no dependiera de la ejecución de algún cálculo mecanocuántico para su uso con redes neuronales.

3. HIPÓTESIS

Premisas:

1. Cualquier tipo de estructura cristalina, como es el caso de la perovskita, se caracteriza porque sus átomos ocupan, dentro de la celda unitaria, sitios descritos a través de un grupo de simetría puntual. Estos sitios caracterizados con un grupo de simetría puntual se conocen como los sitios de Wyckoff.
2. Los sitios de Wyckoff agrupan un conjunto de átomos que poseen el mismo ambiente químico.
3. Los compuestos cristalinos se pueden caracterizar mediante rasgos construidos a partir del número de sitios de Wyckoff que el compuesto posee.
4. Los rasgos construidos a partir de sitios de simetría cuando describan a la estructura cristalina mediante factores de empaquetamiento y de geometría, así como de ambiente químico contribuirán a que las redes neuronales clasifiquen eficientemente a los compuestos cristalinos mediante un modelo de clasificación binaria.

Por lo tanto, se sigue que:

- Las redes neuronales artificiales con alta precisión en la clasificación serán de ayuda para inferir nuevos compuestos de tipo perovskita. Dichas inferencias podrán validarse mediante cálculos cuánticos.

4. OBJETIVOS

Objetivo general:

Inferir nuevos compuestos candidatos a poseer la estructura tipo perovskita.

Objetivos particulares:

1. Desarrollar una metodología para caracterizar compuestos cristalinos con base en sus sitios de simetría puntual (sitios de Wyckoff).
2. Entrenar, utilizando la metodología desarrollada, diferentes redes neuronales que clasifiquen a los compuestos cristalinos en tipo perovskita y no perovskita.
3. Utilizar las redes neuronales desarrolladas para inferir nuevos compuestos con la estructura tipo perovskita.
4. Validar los compuestos inferidos por la red neuronal como potenciales candidatos de adoptar la estructura perovskita con cálculos químico cuánticos.

5. METODOLOGÍA

5.1 OBTENCIÓN Y PREPARACIÓN DE LA BASE DE DATOS.

Para desarrollar las redes neuronales, fue necesario tener acceso a una base de datos de archivos CIF, que son los archivos que contienen toda la información cristalográfica de un compuesto. Esto se consiguió descargando los archivos contenidos en la base de datos libre *Crystallography Open Database* (COD) [72], que se puede consultar a través de http://www.crystallography.net. Las instrucciones para descargar los archivos CIF se pueden consultar en http://wiki.crystallography.net/howtoobtaincod/.

Para descargar la base de datos desde una terminal de Linux fue necesario tener instalada la paquetería *subversion*. Ésto se consigue ingresando en la terminal:

```
$ sudo apt-get install subversion
```

Una vez instalada *subversion*, se ingresa en la terminal la siguiente instrucción para descargar la base de datos:

```
$ svn co svn://www.crystallography.net/cod/cif
```

Los archivos CIF se descargan dentro de una carpeta homónima, la cual se ubica en el directorio donde se haya ejecutado la instrucción anterior en la terminal de Linux. El proceso puede tardar algunas horas dependiendo de la conexión de Internet.

De esta base de datos se detectaron que había algunos archivos CIF donde los símbolos de los elementos de la tabla periódica estaban escritos únicamente en mayúscula. Este hecho puede ocasionar problemas posteriores. Con el programa *cif_fixer.sh*, cuyo código se encuentra en el apéndice de esta tesis, se corrigió la sintaxis de esos símbolos. El programa *cif_fixer.sh*, que se ejecuta desde la terminal de Linux, requiere de un archivo de texto llamado *control.txt*, que contiene el nombre de todos los archivos CIFs de la base de datos.

Posterior a la corrección de los archivos CIF con el programa *cif_fixer.sh*, se procedió a la creación de un DataFrame, el cual es un objeto de la librería Pandas de Python y que se puede guardar como un archivo separado por comas (*.csv). Dicho DataFrame contiene la siguiente información (columnas) de cada uno de los compuestos de *COD* con base en:

- El número de archivo CIF
- Fórmula condensada
- Número de grupo espacial
- Número de elementos diferentes en la fórmula
- Número de átomos en total dentro de la celda unitaria
- Ocupación de cada uno de los sitios de Wyckoff del compuesto cristalino.

De todas las informaciones mencionadas, la ocupación de los sitios de Wyckoff es la que se utiliza para construir los rasgos de los compuestos cristalinos para el modelo de clasificación.

El DataFrame se crea con el programa *"create_dataframe.py"* y se guarda como "cod_dataframe.csv" y como "cod_dataframe.pkl". Este último tipo de archivo está en binario y corresponde al objeto de la librería de Pandas llamado "DataFrame". Al ejecutar *"create_dataframe.py"* también se depuran aquellos compuestos repetidos. Para considerar que un compuesto está repetido, se consideró que dos compuestos no tuvieran la misma fórmula y el mismo grupo espacial. El programa *"create_dataframe.py"* necesita de otro código contenido en el archivo *"Wyckoff_finder.py"* que es con el que se genera la información sobre la ocupación de cada sitio de Wyckoff del compuesto cristalino. El código de *"Wyckoff_finder.py"* se encuentra en el apéndice y es una colección de funciones que permite conocer información sobre los sitios de Wyckoff en el cristal. Para su funcionamiento, el código *Wyckoff_finder.py* utilizó la librería de Python Pymatgen [73].

La creación del DataFrame *“cod_dataframe.pkl”* tiene como finalidad facilitar el acceso a la información de *COD* necesaria para el desarrollo de un modelo de clasificación con redes neuronales.

5.2 CREACIÓN DE LA COLECCIÓN DE COMPUESTOS

5.2.1 DEFINICIÓN DE LAS ESTRUCTURAS TIPO PEROVSKITA DE VÉRTICE COMPARTIDO

Una de las premisas en las que se basa este trabajo es que una estructura cristalina se define por la ocupación de ciertos sitios de Wyckoff en un grupo espacial [6]. La lista de los grupos espaciales donde se pueden encontrar estructuras de tipo perovskita se encuentra en el Apéndice B.1. Este apéndice corresponde a un archivo de texto que se basa en lo publicado por Woodward. Dicho archivo de texto es necesario ya que posteriormente será leído por el programa *“patolli.py*”. Por ello, es necesario que el archivo de texto tenga la estructura que se presenta en el apéndice B.1. A continuación, se presenta en la Figura 5.2.1 una captura de pantalla del archivo *structure_dictionary.txt*, que contiene la información del apéndice B.1:

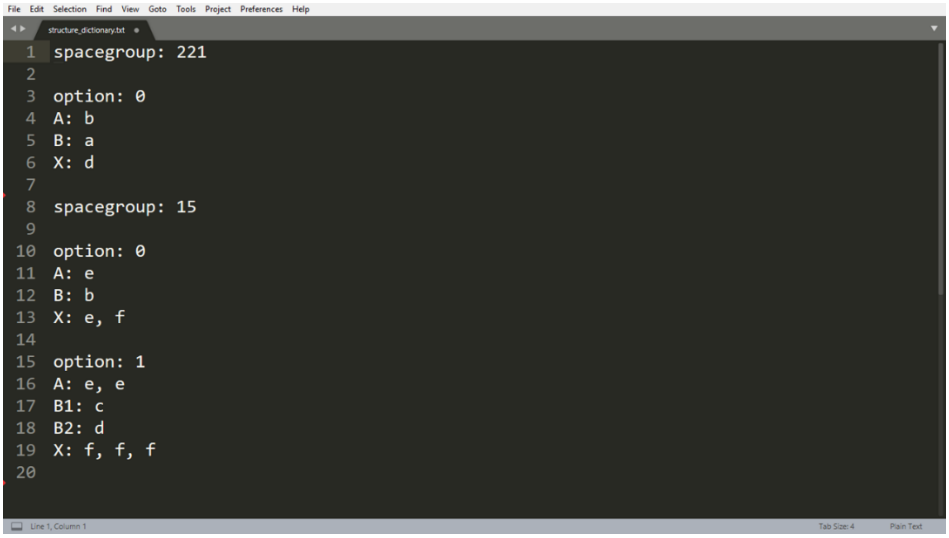


Figura 5.2.1: Captura de pantalla del archivo structrue_dictionary.txt. En esta Figura se detalla la ocupación de los sitios de Wyckoff para las estructuras perovskitas en los grupos espaciales 221 (el aristotipo) y 15 (perovskita monoclínica). Para el grupo espacial 221, sólo hay un tipo de perovskita (sencilla) que se indica con la palabra clave *option*. Para el grupo espacial 15 hay dos tipos: en la *option 0* se indica la perovskita sencilla, mientras que con *option 1* se indica a la perovskita doble. Las letras A, B y X señalan a los átomos en los cuboctaedros, octaedros y vértices, respectivamente. En minúsculas se indica únicamente los símbolos de los sitios de Wyckoff que ocupan los átomos A, B y X. Para las perovskitas dobles, existen dos átomos octaédricos B1 y B2.

En el archivo de texto, cada grupo espacial viene especificado en un renglón con la instrucción *spacegroup*. Debido a que pueden existir estructuras de tipo perovskita sencilla y doble en un mismo grupo espacial, la diferenciación entre estos tipos viene indicado en el renglón con la instrucción *option*. En *option* se deben enlistar todos los tipos de perovskita desde el índice cero (*option* 0). Luego, se especifica qué sitios de Wyckoff son ocupados por cada tipo de átomo: *A* para el sitio cuboctaédrico, *B* para el sitio octaédrico y *X* para el sitio donde se localizan los vértices. La nomenclatura sobre cada tipo de átomo es libre; es decir, se podrían utilizar otras letras mientras se enlisten los símbolos de Wyckoff correctamente.

5.2.2 SELECCIÓN DE LAS MUESTRAS

El programa con el que se desarrollan las redes neuronales se llama *patolli.py* y ahí se encuentra un conjunto de funciones de Python necesarias para entrenar, desde cero, cualquier número de redes neuronales diferentes. Esto se hace mediante la definición de dos archivos de texto, uno que es *structure_dictionary.txt*, que ya hemos mencionado, y otro es *model_control_file.txt*, que define las características de las redes neuronales a entrenar y que se explicará posteriormente.

Al inicio, *patolli.py* convierte la información del archivo *structure_dictionary.txt* a un diccionario de Python con la función *create_dictionary*. En este diccionario de Python, las llaves *(keys)* corresponden a cada uno de los grupos espaciales donde se pueden encontrar estructuras tipo perovskita y sus valores *(values)* son diccionarios que relacionan cada tipo de perovskita (sencilla o doble) con los grupos de simetría puntual en notación Internacional. Es importante mencionar este último hecho, que los sitios ocupados no están descritos con los símbolos de Wyckoff sino con el grupo de simetría puntual.

Posteriormente, *patolli.py* crea la colección de compuestos con la función *create_collection* de la siguiente manera:

1. El usuario puede restringir qué compuestos se considerarán como perovskitas con base en:
    a. Un máximo de número de sitios de Wyckoff.
    b. Un mínimo de elementos en la fórmula. En el caso de los compuestos de tipo perovskita, se restringió a que éstos tuvieran al menos tres elementos diferentes en su fórmula.

Además, se puede imponer una restricción en los compuestos que no son de tipo perovskita con base en un número máximo de átomos dentro de la celda unitaria.

2. La función *create_collection* busca primero a los compuestos con estructura de tipo perovskita dentro de *cod_dataframe.pkl*. Para ello, separa a los compuestos de *cod_dataframe.pkl* que poseen un grupo espacial donde cristalizan las perovskitas. Luego, compara que la ocupación de los sitios de estos compuestos separados corresponda a la de un compuesto tipo perovskita con base en lo especificado en el archivo de texto *structure_dictionary.txt*. De esta comparación, se etiquetan los compuestos como verdaderos (*True*) o falsos (*False*).

Los compuestos etiquetados como *False* se reincorporan a la base de datos original mientras que aquellos como etiquetados como *True* se mantienen aparte.

3. Al final, se cuenta la cantidad de compuestos con la etiqueta *True* para seleccionar de manera aleatoria una misma cantidad de compuestos cristalinos que no representen a una estructura de tipo perovskita. De esta manera, se crea una colección donde las etiquetas estén en proporción 1:1.

Cuando la función *create_collection* termina de ejecutarse, se genera un archivo separado por comas con el nombre de *compounds_collection.csv*. Este archivo contiene, además de las columnas mencionadas para *cod_dataframe.pkl*, la información en la columna *target*, que corresponde al tipo de muestra según nuestro tipo de estructura cristalina.

5.3 CONSTRUCCIÓN DE LOS RASGOS

5.3.1 PREPARACIÓN DE LOS DATOS: CARACTERIZACIÓN DE LOS SITIOS

Previo a la construcción de los rasgos, fue necesario calcular el radio atómico y la electronegatividad promedio de cada sitio de Wyckoff para cada compuesto de la colección creada. El radio atómico, *r*, y la electronegatividad, *χ*, promedios del *i* – ésimo sitio de Wyckoff en el compuesto cristalino es la suma de los productos de los radios atómicos (o electronegatividades) de cada elemento con su fracción de ocupación, *f*, en ese sitio:

$$r_i = \sum_{n=1} f_n r_n$$

$$\chi_i = \sum_{n=1} f_n \chi_n$$

Lo anterior permite modelar la ocupación promedio de un conjunto de elementos en un sitio de Wyckoff sin importar de que el compuesto se trate de una solución solida o de que en su estructura cristalina haya vacancias.

El cálculo de los radios atómicos y electronegatividades promedio se hizo a través de la función *raw_features_extractor*, que se encuentra dentro de *patolli.py*. Esta función necesita del archivo separado por comas *datosrahm.csv* (Apéndice B.2). El archivo *datosrahm.csv* contiene información sobre el radio atómico en Å reportado por Rahm-Ashcroft-Hoffmann [74] y la electronegatividad de Pauling [75] de cada uno de los primeros 96 elementos de la tabla periódica. La función *raw_features_extractor* transforma a datosrahm.csv en un diccionario de Python donde se relaciona a los símbolos de los elementos con su radio atómico y su electronegatividad de Pauling.

Al ejecutar *raw_features_extractor* se homogenizan todos los compuestos de la colección a tener el mismo número de sitios cristalográficos. Esto se hace añadiendo un sitio extra de ocupación nula en aquellos compuestos descritos con menos sitios de Wyckoff. Después de esta adición, se consigue que todos los compuestos de la colección tengan la misma cantidad de sitios. En lo sucesivo, se

hará hincapié en esta diferencia: se llamará como sitios al resultado de la homogeneización y no como sitios de Wyckoff, que es la información original que proviene de la base de datos.

Después de ejecutar la función *raw_features_extractor*, se generan los siguientes objetos de la librería Numpy, los cuales además se guardan:

- **raw_features.npy**, que es un tensor de tres dimensiones: la primera dimensión corresponde a cada compuesto de la colección *compounds_collection.csv*; la segunda, a la caracterización con un máximo número de sitios; y la última, al radio atómico y la electronegatividad de cada sitio.
- **multiplicities.npy**, que es un tensor de tres dimensiones, similar a raw_features.npy. La diferencia está en la última dimensión, pues se refiere a la multiplicidad de cada sitio. Esta información es, esencialmente, la multiplicidad de cada sitio de Wyckoff. Aquellos sitios añadidos tienen una multiplicidad de cero, pues su ocupación es cero.
- **occupation_fractions.npy**, que es un tensor de tres dimensiones similar a raw_features.npy. La diferencia está en la última dimensión, que refiere a la suma de las fracciones de ocupación en ese sitio. En los sitios añadidos el resultado de esta suma era cero.
- **output_values.npy**, que es una matriz, donde la primera dimensión corresponde al número de muestras en la colección y la segunda al valor de salida: 0 para un compuesto que no es perovskita, 1 para los que sí son.

En cada compuesto la información se ordenó en orden creciente de la multiplicidad del sitio.

Al ser objetos de la librería Numpy, todos los archivos generados se guardan en binario.

5.3.2 CONSTRUCCIÓN DE LOS RASGOS 1: FACTORES GEOMÉTRICOS Y DE EMPAQUETAMIENTO.

Los rasgos que comprenden factores geométricos y de empaquetamiento corresponden tanto a cocientes de los radios atómicos promedio en los sitios de Wyckoff como a cocientes de sumas de pares de radios atómicos promedio de los sitios de Wyckoff. La cantidad de rasgos que corresponden a los factores geométricos y a los de empaquetamiento dependen del número de sitios máximo con el que se caracterizan a los compuestos cristalinos.

A continuación, se ilustrará este cálculo de rasgos suponiendo que se tiene una colección de compuestos caracterizados con cuatro sitios. El número de combinaciones ($C(n, r)$) en parejas ($r = 2$) de una colección de cuatro elementos ($n = 4$):

$$C(n,r) = \frac{n!}{r!\,(n-r)!}$$

$$C(4,2) = \frac{4!}{2!\,(4-2)!} = 6$$

Esto implica que, para cuatro radios atómicos promedio r_1, r_2, r_3 y r_4 (que son los elementos en la fórmula de número de combinaciones), hay seis cocientes que se forman por combinaciones en pareja, que se presentan en la parte superior de la Tabla 5.3.1. Estos cocientes están determinados por la librería de Python *itertools*. De igual manera, hay seis posibles sumas en pareja de radios atómicos: $r_1 + r_2$, $r_1 + r_3$, $r_1 + r_4$, $r_2 + r_3$, $r_2 + r_4$ y $r_3 + r_4$. Con esta lista de 6 sumas diferentes, se calcula que el número de combinaciones en pareja es 15. En otras palabras, hay 15 posibles cocientes que se pueden formar con esta lista de sumas y que se encuentran en la parte central de la Tabla 5.3.1.

Los cocientes de radios atómicos están relacionados con la geometría de un átomo central definida por los primeros vecinos. Por otro lado, los cocientes de sumas de pares de radios atómicos están relacionados con la eficiencia en el empaquetamiento del cristal, como es el caso del factor de tolerancia de Goldschmidt. Lo anterior está relacionado con lo expuesto en la sección 1.2.2. Por ejemplo, para el caso de un compuesto de aristotipo de perovskita el rasgo 21 correspondería al factor de tolerancia de Goldschmidt. El aristotipo, que

corresponde al grupo espacial 221, describe a la estructura perovskita con tres sitios de Wyckoff; sin embargo, debido a que se añadieron sitios vacíos, $r_1 = 0$ y por eso el rasgo 21 corresponde al factor de Goldschmidt.

Tabla 5.3.1: Rasgos construidos para una colección caracterizada con cuatro sitios

Cocientes de radios atómicos

$x_1: \frac{r_1}{r_2}$	$x_3: \frac{r_1}{r_4}$	$x_5: \frac{r_2}{r_4}$
$x_2: \frac{r_1}{r_3}$	$x_4: \frac{r_2}{r_3}$	$x_6: \frac{r_3}{r_4}$

Cocientes de sumas de pares de radios atómicos

$x_7: \frac{r_1 + r_2}{r_1 + r_3}$	$x_{10}: \frac{r_1 + r_2}{r_2 + r_4}$	$x_{13}: \frac{r_1 + r_3}{r_2 + r_3}$	$x_{16}: \frac{r_1 + r_4}{r_2 + r_3}$	$x_{19}: \frac{r_2 + r_3}{r_2 + r_4}$
$x_8: \frac{r_1 + r_2}{r_1 + r_4}$	$x_{11}: \frac{r_1 + r_2}{r_3 + r_4}$	$x_{14}: \frac{r_1 + r_3}{r_2 + r_4}$	$x_{17}: \frac{r_1 + r_4}{r_2 + r_4}$	$x_{20}: \frac{r_2 + r_3}{r_3 + r_4}$
$x_9: \frac{r_1 + r_2}{r_2 + r_3}$	$x_{12}: \frac{r_1 + r_3}{r_1 + r_4}$	$x_{15}: \frac{r_1 + r_3}{r_3 + r_4}$	$x_{18}: \frac{r_1 + r_4}{r_3 + r_4}$	$x_{21}: \frac{r_2 + r_4}{r_3 + r_4}$

Funciones de ambiente local

$x_{22}: f_{12}$	$x_{25}: f_{21}$	$x_{28}: f_{31}$	$x_{31}: f_{41}$
$x_{23}: f_{13}$	$x_{26}: f_{23}$	$x_{29}: f_{32}$	$x_{32}: f_{42}$
$x_{24}: f_{14}$	$x_{27}: f_{24}$	$x_{30}: f_{34}$	$x_{33}: f_{43}$

Los rasgos geométricos son calculados con la función *compute_quotients* de *patolli.py*. Esta función necesita del tensor raw_features, calculado en la subsección 5.3.1. Cuando se termina de ejecutar *compute_quotients*, se obtienen los rasgos geométricos. Estos rasgos son guardados como objeto de Numpy, con el nombre de **X.npy**. Este objeto es un tensor de tres dimensiones, donde la primera dimensión corresponde al número de compuestos en la colección, mientras que las dos últimas es una matriz de 1 por la cantidad de rasgos geométricos calculados bajo las consideraciones arriba expuestas. En el caso de que se considerara compuestos con hasta cuatro sitios, esa matriz sería de 1x21.

5.3.3 CONSTRUCCIÓN DE LOS RASGOS 2: ADICIÓN DE LAS FUNCIONES DE LOCALIDAD A LOS RASGOS

Las funciones de localidad, f_{ij}, modelan el entorno químico generado por los todos los átomos vecinos en determinado sitio *j* sobre un átomo central en un sitio

i. Los vecinos que contribuyen en la función de localidad son aquellos que están dentro de una esfera con radio, R_c. Dicho radio representa un radio de corte.

La función de localidad f_{ij} está definida mediante la siguiente ecuación:

$$f_{ij} = (\chi_i - \chi_j) \sum_{n=1}^{d_{ij[n]} \leq R_c} \left[\frac{1}{2}\left(\cos\frac{\pi d_{ij_{[n]}}}{R_c} + 1\right)\right] exp\left[-\left(\frac{d_{ij_{[n]}}}{r_i^{norm} + r_j^{norm}}\right)^2\right] \quad \text{(eq. 1)}$$

Donde $d_{ij[n]}$ es la distancia entre el par de átomos involucrados; χ_i y χ_j refieren a las electronegatividades promedio en esos sitios; r^{norm} refiere a los radios atómicos de cada sitio, donde la suma fue normalizada con la fracción de ocupación; y R_c es una distancia de corte que en este trabajo se consideró de 25 Å.

La ecuación anterior demanda que los sitios *i* y *j* tengan una composición diferente a fin de que $f_{ij} \neq 0$. Por esta razón, los valores de la función cuando $i = j$ no son calculados. Por lo tanto, si los compuestos en una colección se caracterizan con una cantidad *k* de sitios, el número de rasgos que corresponden a las funciones de localidad será $k(k-1)$. En la Tabla 5.3.1 se enlistan las funciones de localidad considerando que el número máximo de sitios en los compuestos es cuatro.

Las funciones de localidad se calculan mediante la generación de un cristal suficientemente grande con el que se pueda generar una esfera con radio igual al de corte (25 Å). En el centro de dicha esfera se encuentra el átomo central en el sitio *i*. Posteriormente se calculan las funciones de localidad considerando a los átomos vecinos en cada sitio *j* distinto. En la Figura 5.3.1 se ilustra este proceso de cálculo de las funciones de localidad.

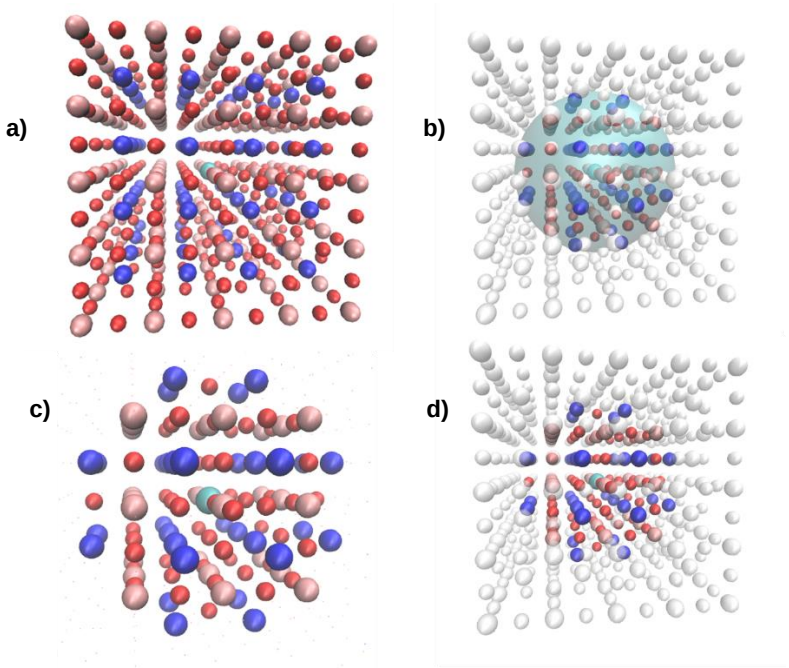


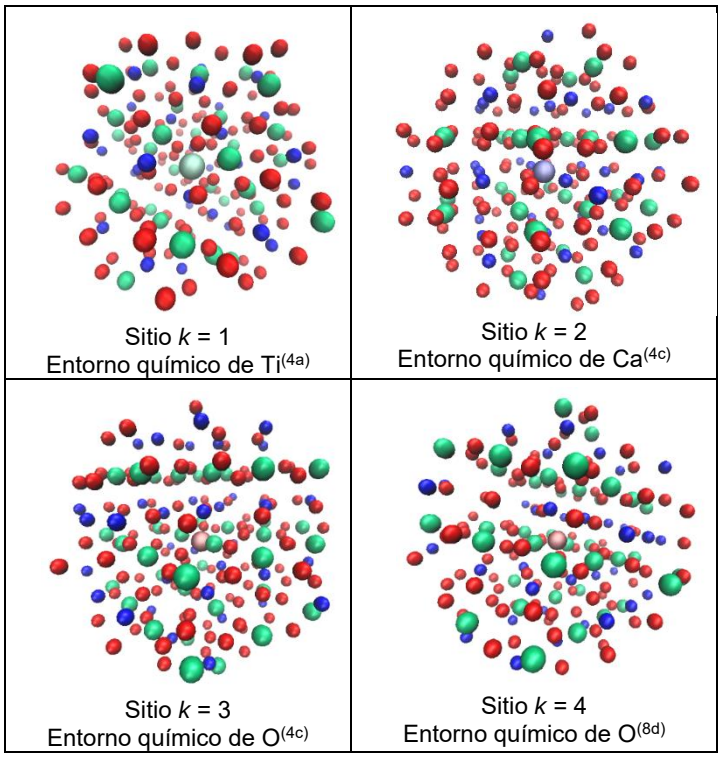
Figura 5.3.1: Cálculo de las funciones de localidad. En a) se representa a un cristal con estructura de perovskita aristotípica. El átomo central, cuyas funciones de localidad se calcularán, se ha señalado en color turquesa. En b) se representa a la esfera de corte con radio R_c. También en b) se diferencian a los átomos que no contribuyen en las funciones de localidad con color blanco. En c) se oculta la esfera de corte y d) se ocultan a los átomos que no contribuyen en las funciones de localidad.

A fin de explicar un poco la información modelada en las funciones de localidad, se presentan las Tablas 5.3.2 y 5.3.3. Para ello, se consideró el compuesto $CaTiO_3$, que cristaliza en el grupo espacial $Pnma$, correspondiente al sistema cristalino ortorrómbico. En ese grupo espacial, dicho compuesto se describe con cuatro sitios de Wyckoff. Por esta razón, es innecesaria la adición de sitios extra con ocupación cero cuando dicho compuesto pertenece a una colección de compuestos caracterizados con cuatro sitios. Los entornos químicos de los átomos localizados en cada sitio de este compuesto son los representados en la Tabla 5.3.2. En esta Tabla también se indica el sitio de Wyckoff asociado. Los átomos se ordenan con base en una multiplicidad ascendente.

Cada entorno químico representado en la Tabla 5.3.2 es posteriormente diferenciado según los vecinos en cierto sitio *j*. Esto está representado en la tabla 5.3.3. Adicionalmente, cada entorno creado por los átomos vecinos *j* sobre el átomo

i está asociado a un rasgo de la Tabla 5.3.1. Dicha asociación se hace explícita en la Tabla 5.3.3.

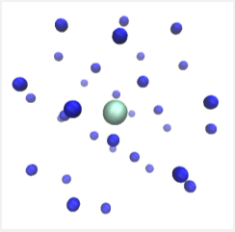
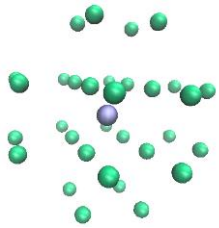
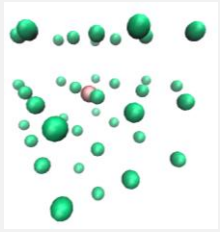
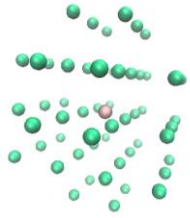
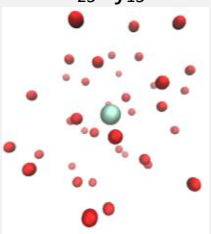
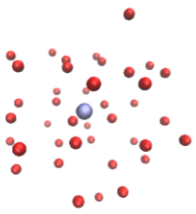
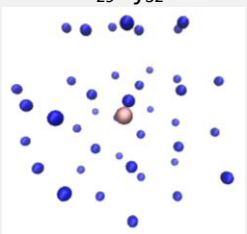
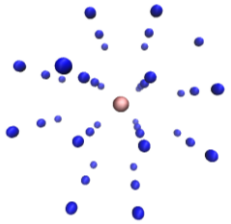
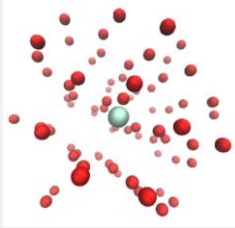
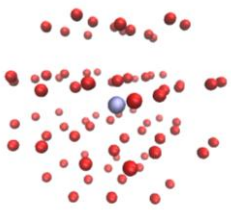
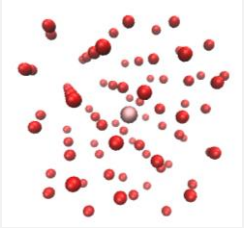
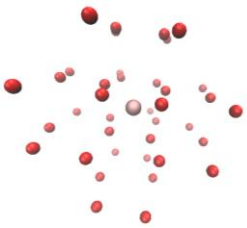
Tabla 5.3.2: Entornos químicos sobre cada átomo central según su sitio para el compuesto $CaTiO_3$ con estructura de perovskita ortorrómbica (grupo espacial $Pnma$). El sitio 1 está ocupado por átomos de titanio (verde); el sitio 2, por átomos de calcio (azul) mientras que los sitios 3 y 4 los ocupan los átomos de oxígeno (rojo). Adicionalmente, los sitios de Wyckoff asociados a cada sitio son el: 4a, para el sitio 1; 4c, para los sitios 2 y 3; y 8d para el sitio 4. Los átomos centrales están señalados con un color más desvanecido que el mencionado.



Debido a que el cómputo de las funciones de ambiente local puede demorar, éstas fueron precalculadas para todos los compuestos de la base de datos y sus valores se encuentran guardados en un archivo en binario (fij_2.0_25_diccio.npy). Lo que se hace con *patolli.py* es utilizar la función *append_local_functions* para anexar a los rasgos contenidos en el objeto X.npy dichas funciones.

Al anexar las funciones de localidad al objeto X.npy, se concluye la construcción de los datos de entrada de las muestras que se alimentarán a las redes neuronales.

Tabla 5.3.3: Entornos químicos generados por los átomos vecinos en un sitio *j* sobre un átomo central en un sitio *i* para el compuesto $CaTiO_3$. Los átomos de titanio, calcio y oxígeno están coloreados en verde, azul y rojo. En la parte superior de cada recuadro de la Tabla se relaciona la función de localidad con el rasgo de la Tabla 5.3.1. En la parte inferior de cada recuadro, se indica los átomos involucrados en el par *ij*.

$x_{22} : f_{12}$  $Ti^{(4a)} \leftarrow Ca^{(4c)}$	$x_{25} : f_{21}$  $Ca^{(4c)} \leftarrow Ti^{(4a)}$	$x_{28} : f_{31}$  $O^{(4c)} \leftarrow Ti^{(4a)}$	$x_{31} : f_{41}$  $O^{(8d)} \leftarrow Ti^{(4a)}$
$x_{23} : f_{13}$  $Ti^{(4a)} \leftarrow O^{(4c)}$	$x_{26} : f_{23}$  $Ca^{(4c)} \leftarrow O^{(4c)}$	$x_{29} : f_{32}$  $O^{(4c)} \leftarrow Ca^{(4c)}$	$x_{32} : f_{42}$  $O^{(8d)} \leftarrow Ca^{(4c)}$
$x_{24} : f_{14}$  $Ti^{(4a)} \leftarrow O^{(8d)}$	$x_{27} : f_{24}$  $Ca^{(4c)} \leftarrow O^{(8d)}$	$x_{30} : f_{34}$  $O^{(4c)} \leftarrow O^{(8d)}$	$X_{33} : f_{43}$  $O^{(8d)} \leftarrow O^{(4c)}$

5.4 DIVISIÓN DE LA COLECCIÓN TOTAL EN LOS CONJUNTOS DE ENTRENAMIENTO-VALIDACIÓN Y DE PRUEBA

Antes de entrenar las redes neuronales es conveniente dividir la colección total en dos conjuntos. El conjunto mayoritario (85 %) será destinado para entrenar y validar los modelos mientras que el conjunto en menor proporción (15%) será utilizado para probar el modelo con muestras nunca usadas antes. Esta división se consigue con la función *split_collection* que pertenece a *patolli.py*. Al terminar de ejecutarse esta función, se generan los siguientes archivos:

- Xtraval.npy y Xtest.npy, que son los rasgos de las muestras en los conjuntos de entrenamiento-validación (del inglés *TRAining* y *cross-VALidation*) y de prueba (*test)*.
- dbtraval.csv y dbtest.csv, que son archivos separados por coma que corresponden a las muestras de los conjuntos mencionados anteriormente.

5.5 ESCALAMIENTO DE LOS DATOS DE ENTRADA

Las redes neuronales fueron entrenadas con las muestras del conjunto Xtraval.npy. Antes de alimentar los datos de entrada de estas muestras a las redes neuronales, fue necesario escalarlos. Esto se hace con dos propósitos:

- Acelerar la convergencia en el proceso de optimización.
- Conseguir que los datos de entrada tengan una distribución de tipo normal.

La forma es que se escalan los datos de entrada fue a través de una estandarización de los mismos. Para ello, hay que calcular el promedio, μ, y la desviación estándar, σ, de cada *i* – ésimo rasgo, $x^{<i>}$, para posteriormente escalarlos con la siguiente fórmula:

$$x_{sc}^{<i>} = \frac{x^{<i>} - \sigma^{<i>}}{\mu^{<i>}}$$

donde $x_{sc}^{<i>}$ corresponde al i – ésimo rasgo escalado.

El escalamiento por estandarización se efectuó con la función *feature_standarisation*. Al ejecutarse esta función, se escalan los datos de entrada de las muestras y se genera un diccionario de Python que contiene los promedios y las desviaciones estándar de cada rasgo. Este diccionario se guarda tanto como un objeto de Numpy, con extensión *.npy*, como un archivo de texto con el nombre de *feature_standarisation*.

Una vez que las redes neuronales hayan sido desarrolladas y se quiera utilizarlas con alguna muestra nueva, será necesario escalar los datos de entrada de dicha muestra con los valores guardados en el diccionario.

5.6 DISEÑO DE LAS REDES NEURONALES ARTIFICIALES

El diseño de la arquitectura de las redes neuronales, así como los hiperparámetros involucrados en su entrenamiento, se especifica en el archivo llamado "*model_control_file.txt*". En la Figura 5.6.1 se presenta una captura del mismo archivo:

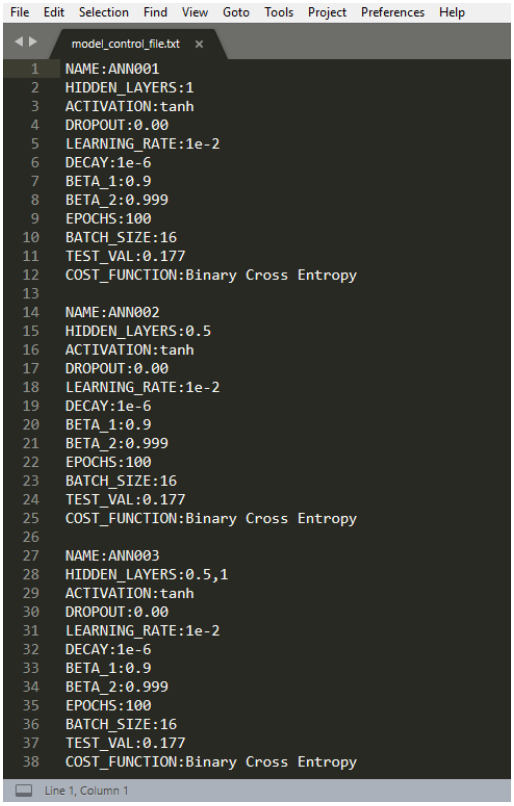


Figure 5.6.1: Captura de pantalla del archivo *model_control_file.txt*, que contiene los hiperparámetros de las redes neuronales a entrenar

Las características de cada red neuronal se presentan en bloques que comienzan con un reglón con la clave *NAME* y terminan con el renglón de la clave *COST_FUNCTION*. Entre cada bloque de renglones se debe dejar un espacio en blanco. Con la función *ctrl_dictionary* de *patolli.py* se convierte cada bloque a un diccionario de Python, que contiene otro diccionario con las claves y valores

indicadas en cada renglón. Todas las redes neuronales que se deseen entrenar deben estar señaladas de esta forma.

A continuación, se explica la función de cada renglón:

- NAME: Es el nombre que acompaña a todos los archivos generados del entrenamiento de una red neuronal.
- HIDDEN_LAYERS: La cantidad de capas ocultas se indica como una lista donde sus elementos se separan con comas. Por ejemplo, el número de capas ocultas en las redes ANN001 y ANN002 (Figura 5.6.1) es 1. El número de capas ocultas en la red ANN03 son dos. Si se quiere entrenar una red con tres capas ocultas, la lista sería de tres elementos separados con dos comas. Cada elemento de la lista indicada con la clave HIDDEN_LAYERS indica la fracción con la que hay que multiplicar a los datos de entrada para definir el número de nodos de cada capa oculta. Por ejemplo, el único elemento de HIDDEN_LAYERS de la red neuronal con el nombre ANN001 indica que hay una capa oculta con 33 rasgos de entrada. El número de nodos en la única capa oculta de ANN002 sería 16.5, pero en este caso el número se redondea al menor entero (*floor*). El número de nodos en las capas ocultas de la R.N.A. ANN003 es de 16 y 33, sucesivamente. Las capas ocultas se organizan de forma progresiva.
- ACTIVATION: Aquí se indica la función de activación que se utiliza en los nodos de las capas ocultas. Los valores permitidos son:
    - *tanh*: función tangente hiperbólica
    - *sigmoid*: función sigmoide
    - *relu*: función ReLU
    - *linear*: función linear
- DROPOUT: Lo que se busca al entrenar una red neuronal es que el aprendizaje de ésta sea aplicable a cualquier tipo de muestra diferente a las utilizadas en el conjunto de entrenamiento y validación. Para esto, es importante que los parámetros de la red neuronal no se sobreajusten. Una de las técnicas que impiden el sobreajuste de los parámetros (regularización)

se conoce como *Dropout* [76] y consiste en ocultar, de manera aleatoria, una fracción de los nodos de las capas ocultas al utilizar un bloque pequeño de muestras. Los nodos ocultos con un bloque pequeño de muestras no son optimizados momentáneamente. Este proceso, de ocultar nodos aleatoriamente, se repite durante todo el entrenamiento. Con la palabra clave DROPOUT se indica la fracción del total de los nodos que se ocultan durante una época en el entrenamiento de la red neuronal.

- LEARNING_RATE: Con esta clave se indica el valor de la velocidad de aprendizaje, α. Este es un número por el que el gradiente se multiplica durante la optimización por disminución del gradiente.
- DECAY: Es el valor del decaimiento, ε, de la velocidad de aprendizaje. La velocidad de aprendizaje en la *n* – ésima iteración se ve modificada de la siguiente manera:

$$\alpha_n = \frac{\alpha_0}{1 + n\varepsilon}$$

- BETA_1: Hiperparámetro de la optimización *Adam* [77] que afecta a la velocidad de aprendizaje y, por lo tanto, a la optimización de los parámetros de la red neuronal. Se recomienda no modificar este valor que por defecto es igual a 0.90.
- BETA_2: Hiperparámetro de la optimización *Adam* [77] que afecta a la velocidad de aprendizaje y, por lo tanto, a la optimización de los parámetros de la red neuronal. Se recomienda no modificar este valor que por defecto es igual a 0.99.
- EPOCHS: El número de épocas, o iteraciones, con las que se entrenarán la red neuronal.
- BATCH_SIZE: La optimización de los parámetros de las redes neuronales se hace por disminución del gradiente por bloques pequeños (*mini-batch gradient descent*). Con este método, las muestras del conjunto de entrenamiento se reparten en bloques que contienen la cantidad ingresada en la clave BATCH_SIZE. Con estos bloques, se optimizan los parámetros

de la redes neuronales y se barre con todo el conjunto de entrenamiento. Una vez que se concluye este barrido, se dice que ha transcurrido una época.

- TEST_VAL: La fracción del conjunto de entrenamiento y validación destinada al conjunto de validación.
- COST_FUNCTION: El nombre de la función de error o de costo.

5.7 DESARROLLO DE LAS REDES NEURONALES ARTIFICIALES

La función más importante del programa *patolli.py* se llama *create_patolli*. Esta función sólo necesita que el usuario proporcione los archivos de texto *"structure_dictionary.txt*" y "*model_control_file.txt*" para poder ejecutarse. Al ejecutarse, también se corren de manera secuencial las funciones involucradas en la creación de la colección de compuestos (sección 5.2), la construcción de los rasgos (sección 5.3), la división del conjunto completo (sección 5.4) y el escalamiento de los rasgos (sección 5.5). La arquitectura de cada red neuronal, que se indica en el archivo "*model_control_file.txt*", se interpreta con la función *model* de *patolli.py*, cuyo código se basa en la librería Keras. El entrenamiento de cada red neuronal se hace a través de la función *training* de *patolli.py*, también basada en la misma librería. Todas las redes neuronales que se entrenan con este programa son del tipo perceptrones multicapa (*feed-forward fully connected Artificial Neural Network*).

Para ejecutar *patolly.py*, es necesario ingresar en la terminal

$python patolly.py

Al ejecutarse la instrucción anterior, el programa solicitará al usuario la siguiente información:

- El nombre del archivo de texto, sin extensión, donde se define la estructura del cristal (*structure_dictionary* es el nombre por defecto).
- El nombre del archivo de texto, sin extensión, donde se definen las características de las redes neuronales a entrenar (*model_control_file* es el nombre por defecto).

- El nombre del archivo pickle, que es objeto DataFrame de Pandas, que contiene la información sobre la ocupación de los sitios de Wyckoff en el cristal. Este es el archivo *cod_dataframe.pkl*, por defecto.
- El número de sitios máximo con el que se restringirá la búsqueda de muestras verdaderas. Se puede dejar en blanco está instrucción para no restringir la búsqueda.
- El número de elementos mínimo con el que se restringirá la búsqueda de muestras verdaderas. Se puede dejar en blanco está instrucción para no restringir la búsqueda.
- El número máximo de átomos por celda unitaria con el que deben escogerse las muestras falsas. Se puede dejar en blanco está instrucción para no restringir la selección.
- La fracción de muestras del conjunto de entrenamiento y validación que serán destinadas a la creación del conjunto de validación. Por defecto, es 0.10
- La frecuencia con la que el programa debe imprimir mensajes en la terminal. Por defecto, el valor es 1. Los otros valores que admite son 0 (no imprime mensajes) y 2 (imprime con mayor frecuencia mensajes).
- Si, al terminar el entrenamiento de las redes neuronales, hay que probar con todas las muestras falsas que quedan disponibles de *cod_dataframe.pkl*. Por defecto, no lo hace.

Todos los archivos que se generen se guardan en una carpeta cuyo nombre es la fecha y hora en la que se ejecutó el programa. Se aconseja que cualquier cambio en el nombre de esa carpeta se realice una vez que el programa *patolli.py* haya terminado de ejecutarse.

Cada red neuronal entrenada genera los siguientes archivos:

- El modelo entrenado, con sus parámetros optimizados y las funciones de activación que se utilizan, se guardan en un archivo con extensión .h5, por ejemplo, ANN001.h5. Este archivo es el que se debe cargar para futuras evaluaciones con otras muestras.

- Un archivo separado por comas, por ejemplo ANN001.csv. Este archivo contiene cuatro columnas:
    - La primera, el valor de la función de costo al terminar una época con las muestras conjunto de entrenamiento.
    - La segunda, la precisión del modelo con las muestras del conjunto de entrenamiento al concluir cada época.
    - La tercera es similar a la primera, sólo que con las muestras del conjunto de validación.
    - La cuarta es similar que la segunda, pero utilizando las muestras del conjunto de validación.
- Las imágenes, por ejemplo, Accuracy_ANN001.png y Cost_function_ANN001.png, que son las gráficas que se construyen con el archivo separado por comas generado.
- Un archivo con terminación *_cnfmat.npy, por ejemplo ANN001_cnfmat.npy. Es un array de Numpy que consiste en la matriz de confusión. Esta matriz de confusión se obtuvo utilizando todas las muestras de los conjuntos de entrenamiento y de validación.
- El archivo, por ejemplo, cnfmat_ANN001.png, que es una gráfica de la matriz de confusión del archivo ANN001_cnfmat.npy.

Al terminar el entrenamiento de la primera red neuronal definida en el archivo *model_control_file.txt*, se crea en la carpeta donde se encuentra *patolli.py* el archivo PRFS_model_control_file.txt, donde se registran las métricas de Precisión, Exhaustividad y Valor-F1. Una vez concluido el entrenamiento de todas las redes neuronales, este archivo se mueve a la carpeta creada al ejecutarse *patolli.py*. Además, también se copian a esa carpeta generada los archivos model_control_file.txt y structure_dictionary.txt.

Las métricas anteriores también se calculan para generar el archivo *test_results.txt*, que utiliza las muestras del conjunto de prueba con cada una de las redes neuronales entrenadas. Este archivo se encuentra en la carpeta creada al ejecutar *patolli.py*.

Por último, si el usuario solicitó que los modelos entrenados se probaran con todas las muestras falsas todavía disponibles de *cod_dataframe.pkl*, se generará un archivo con el nombre de *test_with_all_false.txt*. Este archivo desglosa la recuperación de un tipo de muestra según el número de sitios de Wyckoff que ésta tiene. El archivo *test_with_all_false.txt* también se guarda en la carpeta creada al ejecutar *patolli.py*.

5.8 MODELOS DESARROLLADOS

En este trabajo de tesis, se utilizaron tres colecciones diferentes de muestras (compuestos) para desarrollar los modelos. Todas las colecciones tuvieron compuestos con al menos tres sitios de Wyckoff, ya sea del tipo perovskita o no perovskita:

1. Una colección tuvo muestras con hasta cuatro sitios de Wyckoff, con 2862 compuestos.
2. Otra colección tuvo muestras con hasta seis sitios de Wyckoff, con 3246 compuestos.
3. La última colección tuvo muestras con hasta ocho sitios de Wyckoff, con 3258 compuestos.

En todas las colecciones, la proporción entre compuestos tipo perovskita y no perovskita fue de 1:1.

Dependiendo de la máxima cantidad de sitios de Wyckoff que se puede encontrar en los compuestos de una colección, se homogenizaron éstos a tener un máximo de sitios y se construyeron los datos de entrada de cada compuesto de la forma indicada en la sección 5.3.2. Los datos de entrada de los compuestos de una colección caracterizada con cuatro sitios se presentaron en la Tabla 5.3.1. Los datos de entrada de las colecciones caracterizadas con seis y ocho sitios se presentan en el Apéndice D.

5.9 INFERENCIA DE NUEVOS COMPUESTOS

Las redes neuronales entrenadas estiman la probabilidad de que un compuesto cristalice con una estructura de tipo perovskita. Para hacer esta estimación, es necesario que cualquier compuesto propuesto sea caracterizado con los rasgos de la redes neuronales. A fin de poder calcular los rasgos relacionados con los factores geométricos y de empaquetamiento, se necesita de una propuesta de la distribución de los átomos en los sitios de Wyckoff propios de una estructura tipo perovskita. Además, para calcular las funciones de ambiente local es necesaria una propuesta del espacio donde estos átomos se encuentra dentro de la celda unitaria, es decir, una propuesta de parámetros de red. En la Figura 5.9.1 se presenta el esquema que se debe seguir para utilizar la mejor red neuronal.

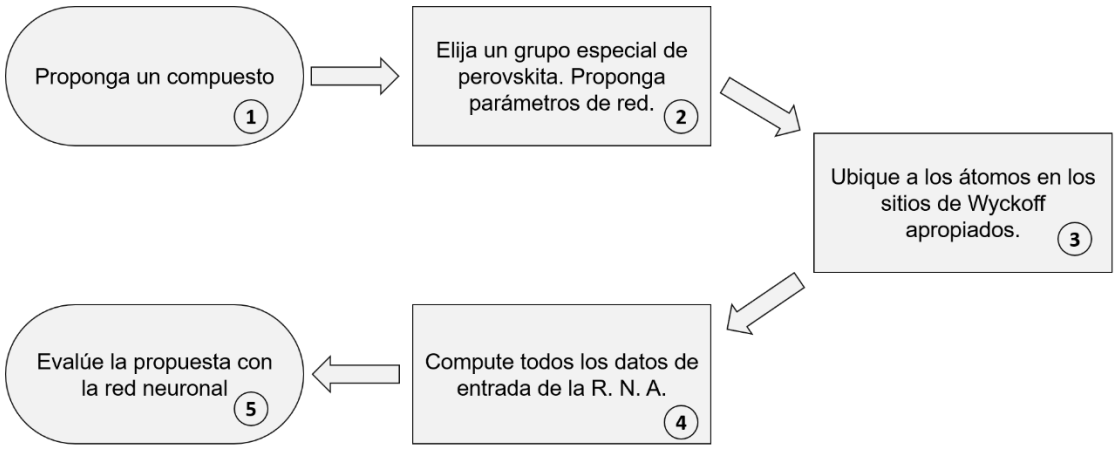


Figura 5.9.1: Esquema de trabajo para evaluar nuevas propuestas de compuestos tipo perovskita.

Si bien es cierto que en principio el modelo desarrollado permite tratar cualquier compuesto con estructura perovskita en cualquier sistema cristalino, la forma en la que se modelaron nuevos compuestos se hizo considerando que éstos cristalizaban en su forma más simétrica. Esta aproximación se sustenta en el hecho de que los compuestos con estructura tipo perovskita adoptan esta fase comúnmente a alta temperatura[10].

La longitud de la celda unitaria en una perovskita cúbica se considera igual al doble de la suma de los radios iónicos del catión en los sitios octaédricos y el anión [16]. En el presente trabajo, debido a que se utilizaron radios atómicos en la construcción de los rasgos, se estudió la validez de esta relación al utilizar los radios atómicos con los que se construyó el modelo. Para ello, se utilizaron 290 y 225

compuestos con estructura tipo perovskita de los grupos espaciales 221 y 225, respectivamente. La longitud de la celda unitaria en los compuestos con el grupo espacial 225 es el doble que de la celda unitaria simple de tipo perovskita. Todas las muestras utilizadas correspondieron a compuestos donde los sitios de Wyckoff estuvieran completamente ocupados por un solo elemento. Como primera aproximación, se propone que el parámetro de red sea la suma de radios atómicos de los átomos en los sitios octaédricos y en los vértices.

$$a_{pred} = n(r_{octaedro} + r_{vertice})$$

En la fórmula, *n* vale 1 o 2 según se trate de una celda unitaria sencilla (grupo espacial 221) o doble (grupo espacial 225)

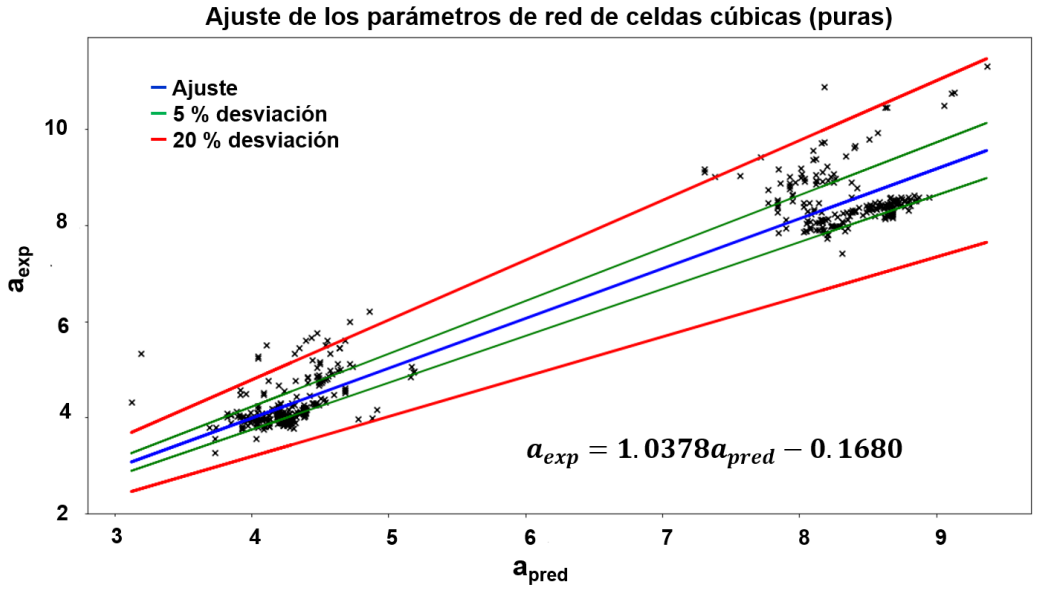
Al hacer un ajuste por mínimos cuadrados, la relación que se encontró entre el parámetro de red experimental y el propuesto fue:

$$a_{exp} = 1.0378 a_{pred} - 0.1680$$

Con un valor de R^2 = 0.9401. Como se pueden observar, la relación entre el parámetro de red experimental y el propuesto aquí es casi la unidad. Por otro lado, la raíz cuadrada del error cuadrático promedio fue de 0.5244, mientras que el error porcentual absoluto promedio fue de 6.318 ± 4.8292 %. Con base en esta última información, y después de observar que el 86.38 % de los 492 compuestos utilizados para este ajuste se encuentran en un margen de ±10% del parámetro de red propuesto, se procedió a proponer los parámetros de red de las nuevas estructuras con valores desviados en ±10%, ±5% y 0% de la suma mencionada.

La propuesta de compuestos con estructura de tipo perovskita cúbica fue hecha con el código *simulate_compounds.py*, que evalúa la probabilidad de cristalizar como perovskita de aristotipo dada una composición y un intervalo para desviar el parámetro de red teórico. Los resultados de las evaluaciones se guardan como diccionarios de Python. Los compuestos explorados se trataron de perovskitas ternarias (puras).

Figura 5.9.2: Relación entre la suma de los radios atómicos de las especies en el sitio octaédrico y el vértice con el parámetro de red experimental en las perovskitas cúbicas.



Aquellas propuestas de compuestos con una probabilidad igual o mayor a 0.50 fueron consideradas como candidatas a cristalizar con una estructura tipo perovskita, o simplemente, como *predicciones*.

5.10 VALIDACIÓN CON CÁLCULOS MECANOCUÁNTICOS

5.10.1 CÁLCULOS DE ENERGÍA DE UN SOLO PUNTO CON CÚMULOS

Se construyeron cúmulos de las predicciones de compuestos con estructura perovskita hechas por la red neuronal. Estos cúmulos corresponden a una superestructura de 2x2x2 veces la celda unitaria. En dichos cúmulos hubo 71 átomos, ocho en sitios octaédricos, 27 en sitios cuboctaédricos y el resto en los vértices (izquierda de la Figura 5.10.1). Con estos cúmulos se implementaron cálculos de energía de un solo punto, mediante teoría de funcionales de la densidad (DFT, por su acrónimo en inglés), con el funcional PBE0 [78] y el conjunto base lanl2dz [79] con potenciales promedio para los electrones internos (*effective-core potential*). Estos cálculos se implementaron con el software TeraChem [80]. Los cálculos implementados tuvieron hasta 300 iteraciones para poder converger. Adicionalmente, los cálculos que después de 100 iteraciones tuvieron un error de

inversión directa en el subespacio iterativo [81] (DIIS, en inglés) mayor a 0.1 fueron interrumpidos.

Los cúmulos cuyos cálculos convergieron fueron considerados para ser validados con cálculos con condiciones de frontera periódica. En concreto, para una misma composición y diferentes parámetros de red, sólo se consideraron aquellos cúmulos que tuvieron la menor energía para ser validados en la segunda instancia.

5.10.2 OPTIMIZACIÓN DE LA GEOMETRÍA UTILIZANDO CONDICIONES DE FRONTERA PERIÓDICA

Para la optimización de la geometría, se construyó una superestructura de 2x2x2 que contuvo 40 átomos (derecha de la Figura 5.10.1). Esta diferencia en el número de átomos se debe a las condiciones de frontera periódica, lo cual requiere que se remuevan los átomos de tres caras adyacentes de la superestructura.

Los cálculos se implementaron, utilizando una mezcla de gaussianas y ondas planas, con el conjunto base DVZP-MOLOPT [82] y el funcional PBEsol [83]. El software utilizado fue cp2k [84].

Con este tipo de cálculos lo que se investigó fue la estabilidad del marco de octaedros con vértice compartido, que es característico de las estructuras perovskita. El grupo espacial de las estructuras que convergieron como perovskita se determinó con el software FINDSYM [85].

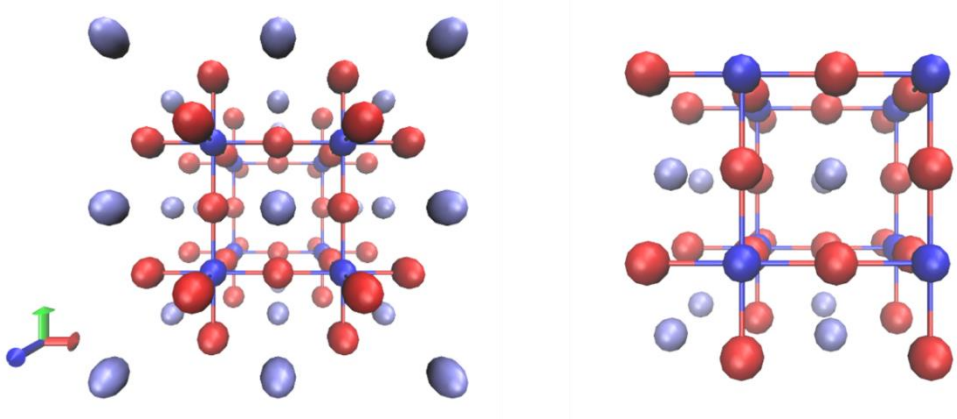


Figura 5.10.1: Estructuras utilizadas para los cálculos mecanocuánticos.

6. RESULTADOS Y ANÁLISIS

6.1 DISTRIBUCIÓN DE LOS COMPUESTOS ENCONTRADOS CON ESTRUCTURA PEROVSKITA

Todos los compuestos utilizados para el desarrollo de las redes neuronales se encuentran en el Apéndice C.

6.1.1 POR NÚMERO DE SITIOS OCUPADOS

Se encontraron en la base de datos *Crystallography Open Database* 1629 compuestos diferentes con estructura tipo perovskita. La distribución de estos compuestos con base en el número de sitios de Wyckoff se encuentra en Figura 6.1.1. En dicha Figura se diferencian cinco subconjuntos de compuestos con base en el número de sitios de Wyckoff. Los subconjuntos más abundantes fueron aquellos descritos con cuatro sitios (en rojo) y tres sitios de Wyckoff (en morado), los cuales se conformaron de 878 (53.90 %) y 553 compuestos (33.95 %), respectivamente. Ambos subconjuntos sumaron 1431 compuestos y representaron el 87.85 % de todas las perovskitas encontradas en la base de datos. Debido a que estos dos subconjuntos englobaron la mayor parte de todas las perovskitas

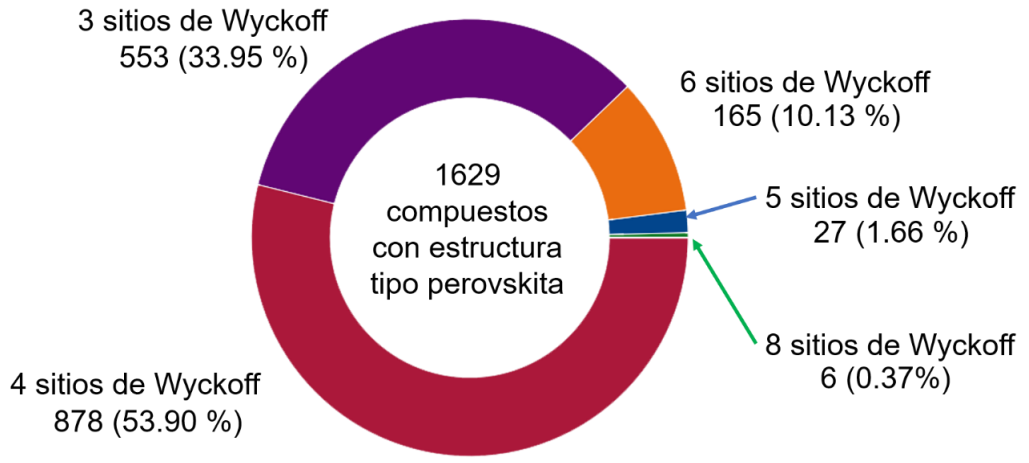


Figura 6.1.1: Distribución de los compuestos tipo perovskita en términos del número de sitios de Wyckoff.

encontradas, se creó la colección de compuestos descrita originalmente con tres o cuatro sitios de Wyckoff.

El tercer subconjunto de compuestos encontrado más abundante se conformó por perovskitas descritas con seis sitios de Wyckoff (Figura 6.1.1, en naranja). Este subconjunto contuvo 165 compuestos y representó el 10.33 % de la colección completa. Los subconjuntos menos abundantes fueron aquellos descritos con cinco (azul) y ocho (verde) sitios de Wyckoff, los cuales contribuyeron al total de perovskitas encontradas con 27 y 6 compuestos, respectivamente. Estos subconjuntos representaron el 1.66 y 0.37 % de todas las perovskitas encontradas, en el orden anteriormente mencionado. Por esta razón, se crearon, además, otras dos colecciones para desarrollar las redes neuronales. Una de estas colecciones contuvo compuestos descritos con hasta seis sitios de Wyckoff mientras que la otra contuvo hasta ocho sitios de Wyckoff. La colección conformada por compuestos descritos con hasta seis sitios de Wyckoff contuvo el 99.64 % de todos los compuestos encontrados con estructura tipo perovskita, y que correspondieron a 1623 compuestos. Naturalmente, la colección formada con compuestos con hasta ocho sitios de Wyckoff utilizó a todas las perovskitas halladas en la búsqueda dentro de la base de datos.

6.1.2 POR GRUPO ESPACIAL

En la Figura 6.1.2 se desglosa la distribución de los compuestos tipo perovskita encontrados con base en su grupo espacial. La información plasmada en la Figura 6.1.1 se retoma como el anillo interno de la Figura 6.1.2 para presentar la correspondencia entre el grupo espacial y la cantidad de sitios de Wyckoff que describe. Los porcentajes que se muestran en la Figura 6.1.2 son respecto al total de compuestos con estructura perovskita encontrados.

Los compuestos tipo perovskita descritos con tres sitios de Wcykoff correspondieron a los grupos espaciales cúbico $Pm\bar{3}m$ y trigonal $R\bar{3}c$, con 397 (24.37 % del total) y 156 (9.58 %) compuestos, respectivamente. Las perovskitas encontradas con cuatro sitios de Wyckoff se distribuyeron en su mayoría en los grupos espaciales $Pnma$ y $Fm\bar{3}m$, correspondientes a los sistemas cristalinos

ortorrómbico y cúbico. Ambos grupos espaciales suman 713 de los 878 compuestos encontrados de perovskitas con cuatro sitios de Wyckoff. Otros grupos espaciales encontrados dentro de este conjunto fueron: $C2/m$ (10 compuestos, sistema monoclínico), $Imma$ (15 compuestos, ortorrómbico), $P4/mbm$ (9 compuestos, sistema tetragonal), $I4/mcm$ (51 compuestos, tetragonal), $R\bar{3}$ (35 compuestos, trigonal) y $Im\bar{3}$ (45 compuestos, cúbico).

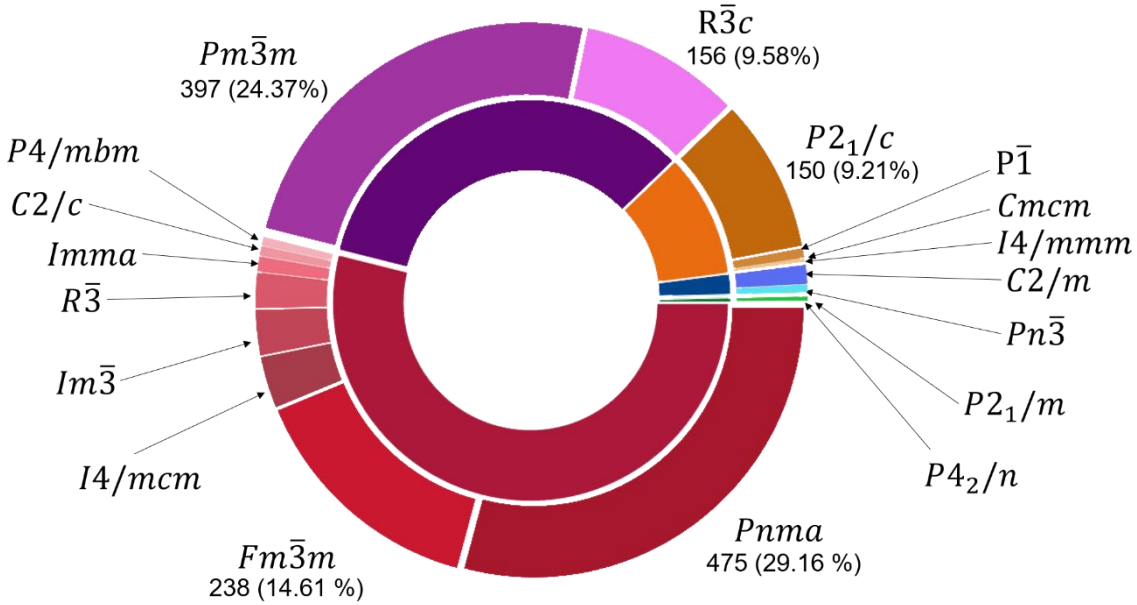


Figura 6.1.2: Distribución de los compuestos tipo perovskita en términos de su grupo espacial.

Además, los compuestos tipo perovskita descritos con seis sitios de Wyckoff correspondieron mayoritariamente al grupo espacial monoclínico $P2_1/c$, con 150 de los 165 compuestos de este conjunto. El resto de este tipo de compuestos se encontraron en los grupos espaciales triclínico $P\bar{1}$ (10 compuestos), ortorrómbico $Cmcm$ (4 compuestos) y tetragonal $I4/mmm$ (1 compuesto).

Los compuestos tipo perovskita descritos con cinco sitios de Wyckoff se distribuyeron en los grupos espaciales monoclínico $C2/m$ y cúbico $Pn\bar{3}$, con 19 y 8 compuestos, respectivamente. Finalmente, los compuestos tipo perovskita descritos con ocho sitios de Wyckoff correspondieron a los grupos espaciales monoclínico $P2_1/m$ y tetragonal $P4_2/n$, con 5 y un compuesto, respectivamente.

6.1.3 POR COMPOSICIÓN QUÍMICA

En la Figura 6.1.3 se muestra la presencia relativa de los elementos de la tabla periódica en los 1629 compuestos con estructura perovskita. Los elementos que estuvieron presentes en los compuestos tipo perovskita son los que están coloreados. Los porcentajes de presencia de cada elemento se desglosan en la Tabla 6.1.1.

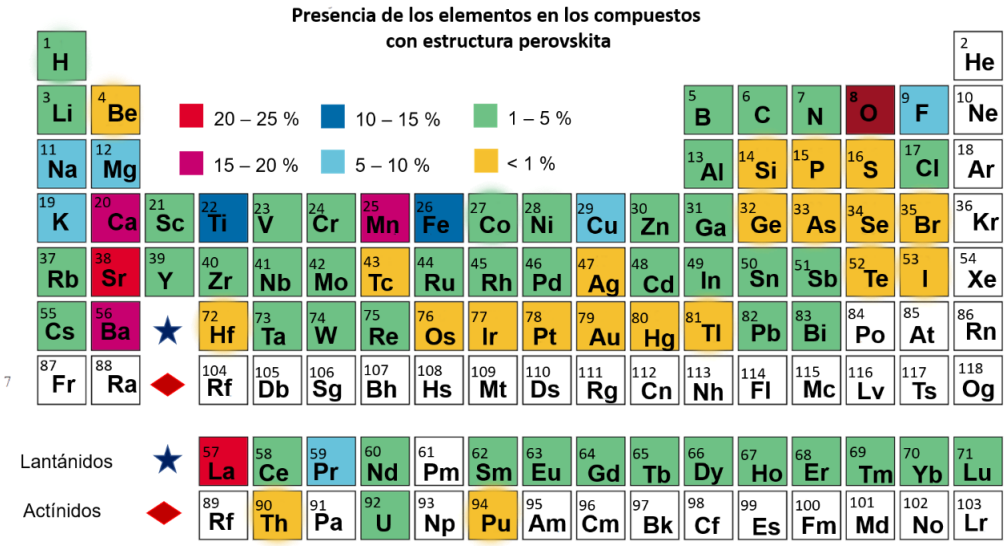


Figura 6.1.3: Presencia de los elementos en los compuestos con estructura perovskita. En color marrón se destacó al elemento con mayor presencia en los compuestos, que fue el oxígeno (83.61 %). En esta Figura 1 % equivale aproximadamente a 16 compuestos.

En la Figura 6.1.3 se destaca en color marrón al elemento con mayor presencia en los 1629 compuestos tipo perovskita, el cual fue el oxígeno. Este elemento se encontró en 1362 compuestos (83.61 %). En rojo se muestran a los elementos estroncio y lantano, los cuales se presentaron en 383 (23.51 %) y 355 (21.79 %) compuestos, respectivamente. En morado se muestran a los elementos manganeso, bario y calcio, que estuvieron presentes en 322 (19.77 %), 306 (15.41 %) y 251 (15.41 %) compuestos, respectivamente. En azul intenso se muestran a

Tabla 6.1.1: Presencia de los elementos en los compuestos tipo perovskita.

Elemento	Cantidad de compuestos	%
O	1362	83.61
La	383	23.51
Sr	355	21.79
Mn	322	19.77
Ba	306	18.78
Ca	251	15.41
Fe	232	14.24
Ti	190	11.66
Na	156	9.58
F	127	7.80
Mg	123	7.55
K	101	6.20
Pr	94	5.77
Cu	84	5.16
Nd	81	4.97
Co	80	4.91
N	78	4.79
Ga	77	4.73
Bi	72	4.42
Nb	71	4.36
Ni	68	4.17
Ta	66	4.05
B	65	3.99
Pb	64	3.93
Cr	60	3.68
In	58	3.56
U	58	3.56
Ce	56	3.44
V	55	3.38
Ru	54	3.31
W	53	3.25
Sc	52	3.19
Al	52	3.19
Mo	52	3.19
C	51	3.13
Y	50	3.07
Rh	46	2.82
Cs	46	2.82
Tb	45	2.76
Sm	41	2.52

Elemento	Cantidad de compuestos	%
Zr	39	2.39
Sn	34	2.09
Rb	33	2.03
Li	33	2.03
Er	31	1.90
Eu	31	1.90
Dy	30	1.84
Re	29	1.78
Sb	28	1.72
Yb	28	1.72
Cl	28	1.72
Ho	26	1.60
Tm	25	1.53
Pd	24	1.47
Gd	23	1.41
Cd	23	1.41
H	19	1.17
Zn	19	1.17
Lu	17	1.04
Ir	16	0.98
Ge	16	0.98
Au	15	0.92
Br	15	0.92
Pt	15	0.92
Si	14	0.86
I	13	0.80
Tl	11	0.68
Ag	11	0.68
Hg	10	0.61
Te	9	0.55
S	7	0.43
Se	7	0.43
Os	6	0.37
Hf	6	0.37
Th	5	0.31
P	3	0.18
Be	3	0.18
As	2	0.12
Tc	1	0.06
Pu	1	0.06

los elementos hierro y titanio, presentes en 232 (14.24 %) y 190 (11.66 %) compuestos, respectivamente.

En color azul claro se muestran al sodio, flúor, magnesio, potasio, praseodimio y cobre, con 156 (9.58 %), 127 (7.80 %), 123 (7.55 %), 101 (6.20 %), 94 (5.77 %) y 84 compuestos (5.16 %), respectivamente. El resto de los elementos se presentan en los colores verde y amarillo, que representan una presencia entre el 1 – 5 % y menor al 1 %, respectivamente. Se deja al lector que consulte la información de estos elementos en la Tabla 6.1.1.

Como complemento a lo presentado en la Tabla 6.1.1 la distribución de los compuestos tipo perovskita en términos de la cantidad de elementos diferentes en su fórmula se presenta en la Tabla 6.1.2. La información plasmada en dicha Tabla se construyó utilizando todas las perovskitas encontradas. Los porcentajes de presencia relativa en los compuestos, mostrados en la Tabla 6.1.2, cambiaron poco al considerar únicamente los compuestos tipo perovskita con hasta cuatro o seis sitios de Wyckoff.

Tabla 6.1.2: Distribución de los compuestos tipo perovskita en función de la cantidad de elementos distintos en su fórmula. El número de compuestos tipo perovskita encontrados fue de 1629.

	3 elementos diferentes	4 elementos diferentes	5 elementos diferentes	6 elementos diferentes
Compuestos tipo perovskita	483 (29.65 %)	853 (52.36 %)	276 (16.94 %)	17 (1.04 %)

A continuación, se presenta la distribución de los compuestos tipo perovskita encontrados en términos del tipo de compuesto inorgánico. Ninguno de los compuestos encontrados fue de tipo orgánico o de las llamadas perovskitas híbrido orgánico inorgánico (OIHP, por su acrónimo en inglés). Esta información se encuentra en la Tabla 6.1.3. La mayor parte de los compuestos tipo perovskita encontrados fueron óxidos, con 1327 compuestos, lo cual representó el 81.46 % de todos los encontrados. En menor cantidad, los siguientes tipos de compuestos más abundantes fueron los fluoruros (103 compuesto, 6.32 %), boruros (50, 3.07 %), nitruros (37, 2.27 %), cloruros (24, 1.47 %) y oxifluoruros (22, 1.35 %). Todos los

tipos de compuestos perovskita anteriormente mencionados suman 95.94 %. El resto de los compuestos fueron, hidruros, bromuros, oxinitruros, yoduros, carburos, sulfuros, compuestos metálicos, hidrofluoruros, bromocloruros, oxibromuros, oxicloruros, selenuros, oxiyoduros y fosfuros.

Los porcentajes reportados en la Tabla 6.1.3 poco cambiaron si se sólo se consideraban a los compuestos tipo perovskita con hasta seis o cuatro sitios de Wyckoff. Al considerar únicamente los compuestos tipo perovskita con hasta seis sitios de Wyckoff, la cantidad de óxidos cambió a 1322 y la de yoduros a 6. En adición, al considerar los compuestos tipo perovskita con hasta cuatro sitios de Wyckoff, los siguientes tipos de compuestos se ajustaron de la siguiente manera: óxidos, a 1156 compuestos; fluoruros, a 92; cloruros, a 15; hidruros, a 11; bromuros, a 9; yoduros, a 4; y la de compuestos metálicos cambió a 4.

Tabla 6.1.3: Distribución de los compuestos tipo perovskita encontrados en términos de su composición química.

Tipo de compuesto	Cantidad	% del total
Óxidos	1327	81.46
Fluoruros	103	6.32
Boruros	50	3.07
Nitruros	37	2.27
Cloruros	24	1.47
Oxifluoruros	22	1.35
Hidruros	13	0.80
Bromuros	10	0.61
Oxinitruros	7	0.43
Yoduros	7	0.43
Carburos	6	0.37
Sulfuros	6	0.37
Metálicos	5	0.31
Hidrofluoruros	4	0.25
Bromocloruros	2	0.12
Oxibromuros	2	0.12
Oxicloruros	1	0.06
Selenuros	1	0.06
Oxiyoduros	1	0.06
Fosfuros	1	0.06

6.2 COMPUESTOS TIPO NO PEROVSKITA EN LAS COLECCIONES

Las colecciones usadas en el entrenamiento de las redes neuronales artificiales se crearon de manera que en éstas hubiera una proporción 1:1 entre muestras verdades y falsas, es decir, compuestos tipo perovskita y no perovskita. Todas las colecciones contuvieron a los mismos compuestos de tipo perovskita dependiendo del número de sitios con el que se caracterizaban. Dicho en otras palabras, las colecciones de compuestos con hasta ocho o seis sitios contuvieron a todas los compuestos perovskita con hasta cuatro sitios de Wyckoff. Análogamente, la colección que caracterizó a los compuestos con ocho sitios contuvo a todos los compuestos perovskita con hasta seis sitios de Wyckoff.

En la Tabla 6.2.1 se desglosa la cantidad de compuestos de tipo no perovskita en cada colección con base en su número de sitios de Wyckoff. La proporción presentada en la Tabla 6.2.1 de los compuestos tipo no perovskita en términos de su número de sitios de Wyckoff es cercana a la mostrada en la Figura 6.1.1 para los compuestos tipo perovskita. Adicionalmente, los compuestos tipo no perovskita fueron en general los mismos en las tres colecciones. Particularmente, este tipo de compuestos con cinco y seis sitios de Wyckoff fueron exactamente los mismos en las colecciones de seis y ocho sitios. En la Figura 6.2.1 se muestran, a través de diagramas de Venn, las cantidades de compuestos no perovskita de tres y cuatro sitios de Wyckoff compartidos en las tres colecciones. Los compuestos de este tipo con cuatro sitios de Wyckoff fueron esencialmente los mismos en las tres colecciones mientras que esto sólo se observó con las colecciones de seis y ocho sitios para aquellos con tres sitios de Wyckoff. Esta diferencia en los compuestos no perovskita de tres sitios de Wyckoff entre las colecciones de seis y ocho sitios con la de cuatro sitios se debe al código utilizado en la función *create_collection* de *patolli.py* para seleccionar las muestras falsas en esta última colección. La función utilizada para la creación de la colección de cuatro sitios correspondió a las primeras etapas de desarrollo del programa *patolli.py.* Cabe mencionar que esta colección de cuatro sitios fue la utilizada para desarrollar la red neuronal artificial que se publicó en *Journal of Solid State Chemistry* [86].

Tabla 6.2.1: Distribución de los compuestos tipo no perovskita en términos de su número de sitios de Wyckoff.

COMPUESTOS TIPO NO PEROVSKITA					
	Con 3 sitios de Wyckoff	Con 4 sitios de Wyckoff	Con 5 sitios de Wyckoff	Con 6 sitios de Wyckoff	Con 8 sitios de Wyckoff
En la colección de cuatro sitios	552	879	-	-	-
En la colección de seis sitios	554	876	27	166	-
En la colección de ocho sitios	553	877	27	166	6

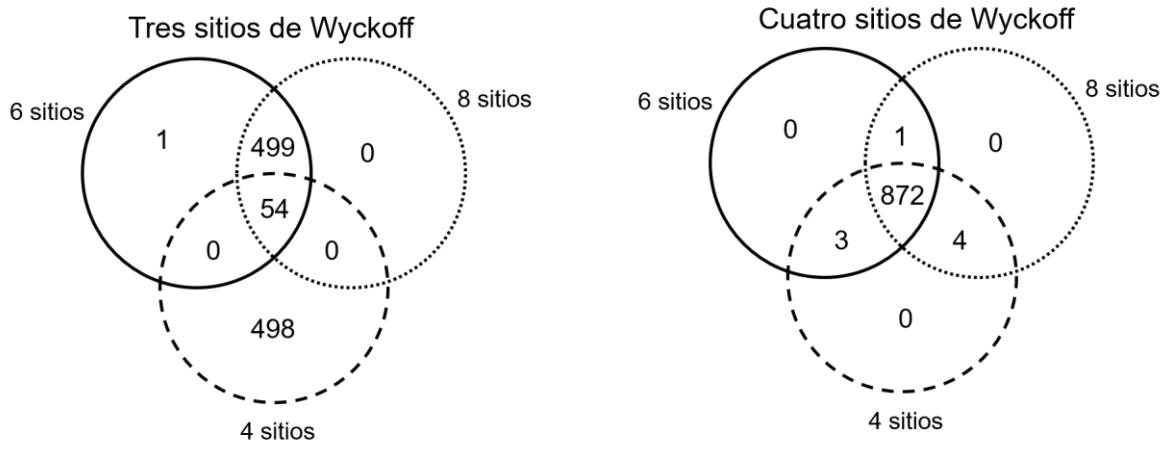


Figura 6.2.1: Se muestran las cantidades de compuestos tipo no perovskita de tres y cuatro sitios de Wyckoff compartidos en las tres colecciones.

En las Tablas 6.2.2 y 6.2.3 se muestra la distribución de los compuestos no perovskita en función del número de elementos diferentes en su fórmula. La Tabla 6.2.2 y la Tabla 6.2.3 refieren a los compuestos no perovskitas de las colecciones con ocho y cuatro elementos. La distribución para los mismos compuestos con la colección de seis sitios no se muestra ya que, como anteriormente se explicó, comparte esencialmente los mismos compuestos no perovskita que con la colección de ocho sitios. De hecho, los porcentajes que muestran las Tablas 6.2.2 y 6.2.3 son cercanos. Una diferencia notable entre esta distribución de compuestos con aquella mostrada en la Tabla 6.1.2, de compuestos perovskitas, es la presencia de compuestos con menos de 3 elementos diferentes. Esto se debe a que no se puso una restricción en la cantidad mínima de elementos diferentes para los compuestos no perovskita. Además, los porcentajes de compuestos con tres, cuatro y seis

elementos diferentes son similares entre compuestos perovskita (Tabla 6.1.2) y los compuestos no perovskita (Tablas 6.2.2 y 6.2.3).

Tabla 6.2.2: Distribución de los compuestos no perovskita en términos del número de elementos diferentes en la colección de ocho sitios. El total de este tipo de compuestos fue 1629.

	Menos de 3 elementos diferentes	3 elementos diferentes	4 elementos diferentes	5 elementos diferentes	6 elementos diferentes	Más de 6 elementos diferentes
Compuestos no perovskita	215 (13.20 %)	902 (55.37 %)	415 (25.48 %)	75 (4.60 %)	17 (1.04 %)	5 (0.30 %)

Tabla 6.2.3: Distribución de los compuestos no perovskita en términos del número de elementos diferentes en la colección de cuatro sitios. El total de este tipo de compuestos fue 1431.

	Menos de 3 elementos diferentes	3 elementos diferentes	4 elementos diferentes	5 elementos diferentes	6 elementos diferentes	Más de 6 elementos diferentes
Compuestos no perovskita	215 (15.02 %)	814 (56.88 %)	325 (22.71 %)	61 (4.26 %)	10 (0.70 %)	6 (0.42 %)

6.3 DESEMPEÑO DE LAS REDES NEURONALES ARTIFICIALES

A continuación se presentan los resultados de las mejores redes neuronales artificiales desarrolladas con las colecciones que caracterizan a los compuestos con cuatro, seis y ocho sitios. Se entrenaron diversas redes neuronales con diferentes números de capas y nodos por capa. Además, se probaron diferentes valores de regularización con Dropout y diferentes funciones de activación en los nodos de las capas ocultas. No obstante, las mejores redes neuronales se obtuvieron utilizando la función tangente hiperbólica en los nodos de las capas ocultas. En todos los casos, se utilizó la función sigmoide como función de activación del único nodo de la capa de salida.

Se consideró que las mejores redes neuronales artificiales desarrolladas con cada colección tenían las siguientes características:

- Tuvieron los valores más bajos de función de costo con las muestras de entrenamiento y de validación.
- La diferencia entre los valores finales de las funciones de costo de los conjuntos de entrenamiento y validación fueron las más pequeñas entre todas las redes neuronales.
- Tuvieron los puntajes más altos en la precisión de compuestos perovskita y no perovskita en los conjuntos entrenamiento-validación (TRAVAL), de prueba (TEST).

6.3.1 RED NEURONAL ARTIFICIAL PARA COMPUESTOS CARACTERIZADOS CON CUATRO SITIOS

La red neuronal artificial que mejor clasificó a los compuestos cristalinos entre perovskitas y no perovskitas tuvo dos capas ocultas, cada una con 132 nodos. Esta red neuronal artificial tuvo en total 22,177 parámetros:

- 4356 pesos (elementos de matriz) entre la capa de entrada y la primera capa oculta, además de 132 términos de sesgo en la primera capa oculta.

- 17,424 pesos entre las dos capas ocultas, además de los 132 términos de sesgo de la segunda capa oculta.
- 132 pesos entre la segunda capa oculta y la capa de salida; además de un término de sesgo en el nodo de la capa de salida.

Los nodos de las capas ocultas fueron optimizados utilizando una fracción de dropout de 0.40. Los valores de la función de costo de los conjuntos de entrenamiento y de validación, al término del entrenamiento, eran de 0.0900 y 0.0873, respectivamente. La diferencia entre ambos valores sugiere que el sobreentramiento es mínimo y que la red neuronal artificial desarrollada es capaz de generalizar lo "*aprendido*" a otras muestras no utilizadas durante su entrenamiento.

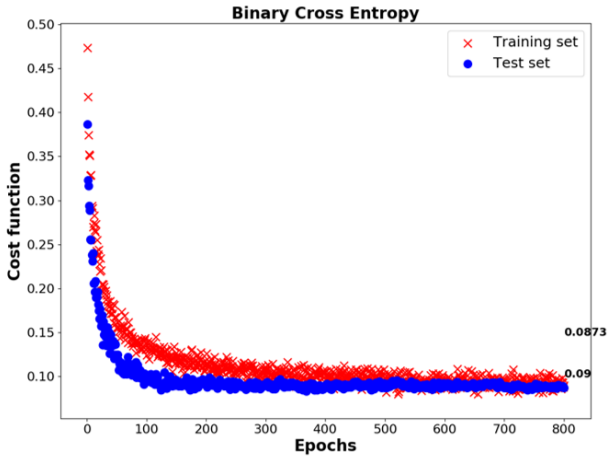


Figura 6.3.1: Gráfica de la función de costo de los conjuntos de entrenamiento y de validación contra las épocas utilizadas en el entrenamiento de la red neuronal artificial. Al término del entrenamiento, el valor de la función de costo fue de 0.0873 y 0.09 para los conjuntos de entrenamiento y de validación.

En las Figuras 6.3.2 y 6.3.3 (página 86) se muestran las matrices de confusión en la clasificación de los compuestos de los conjuntos TRAVAL y TEST. Las matrices de confusión permiten inspeccionar visualmente el número de muestras (compuesto) que el algoritmo (la red neuronal artificial) está confundiendo. En las filas se indica a los compuestos de la colección de cada tipo, es decir, se representa las etiquetas reales de las muestras del conjunto. En las columnas se representa a los compuestos que la red neuronal clasificó, o predijo, como de cierto tipo. Las predicciones hechas por la red correctamente se localizan en la diagonal

principal de las matrices. Las que se encuentran fuera corresponden a las predicciones erróneas.

En términos más estadísticos, los compuestos de tipo perovskita corresponden a los casos positivos mientras que aquellos de tipo no perovskita corresponden a casos negativos. Después de que la red neuronal clasificó, se pueden distinguir los siguientes casos en una matriz de confusión:

- Verdaderos Negativos (VN): Que son los compuestos clasificados correctamente por la red como no perovskitas (elemento de matriz de la primera fila, primera columna).
- Falsos Positivos (FP): Son los compuestos clasificados erróneamente por la red como perovskitas (elemento de matriz de la primera fila, segunda columna).
- Falsos Negativos (FN): Los compuestos clasificados erróneamente por la red como no perovskitas (elemento de matriz de la segunda fila, primera columna).
- Verdaderos Positivos (VP): Son los compuestos correctamente clasificados por la red como perovskitas (elemento de matriz de la segunda fila, segunda columna).

Con base en lo anterior, se definen las métricas Precisión y la Exhaustividad como sigue [87]:

$$Precision = \frac{VP}{VP + FP}$$

$$Exhaustividad = \frac{VP}{VP + FN}$$

Las definiciones presentadas anteriormente se formulan para los casos positivos, es decir, los compuestos tipo perovskita. No obstante, esas mismas definiciones se pueden aplicar a los casos negativos. Así en las Tablas de esta subsección se reportan las precisiones y exhaustividades para ambos tipos de compuestos.

La precisión es una métrica que se construye con base en las predicciones hechas por un algoritmo. Una precisión alta es indicador de que ha habido pocos falsos positivos (o negativos) en el historial de inferencias del algoritmo. Esta métrica, por lo tanto, es la más adecuada para evaluar la capacidad predictiva de una red ya que está relacionada con la probabilidad de que una nueva predicción resulte exitosa.

Por otro lado, la exhaustividad es una métrica que se construye a partir del conocimiento previo de un conjunto de cierto tipo de muestras. La exhaustividad mide la tasa con la que un algoritmo recuperó las muestras de dicho tipo después de clasificar.

Ambas métricas anteriormente presentadas son indicadores de lo bueno que es un clasificador, es decir, una red neuronal. Finalmente, una métrica que mide el compromiso que hay entre la precisión y la exhaustividad es el Valor-F1. Esta métrica está definida como:

$$Valor - F1 = 2 \cdot \frac{Precision \cdot Exhaustividad}{Precision + Exhaustividad}$$

Una red con un Valor-F1 igual a 1 se obtiene cuando la precisión y la exhaustividad valen ambos 1.

Después de esta breve digresión, se presentaran los resultados de dichas métricas en los conjuntos de entrenamiento-validación y de prueba de la colección de cuatro sitios. Se hará enfásis en los resultados obtenidos en precisión ya que el objetivo de este trabajo es predecir nuevos compuestos.

En la Tabla 6.3.1 se muestran las métricas de precisión, exhaustividad y valor-F1 del conjunto de entrenamiento-validación (TRAVAL). La matriz de confusión a partir de la cual se calcularon estas métricas se muestra en la Figura 6.3.2. Los valores obtenidos de estas métricas fueron muy homogéneos y en todos los casos mayores a 97 %. Este porcentaje se obtuvo sin importar el tipo de

compuesto o si sólo se consideraba compuestos con cierta cantidad de sitios de Wyckoff.

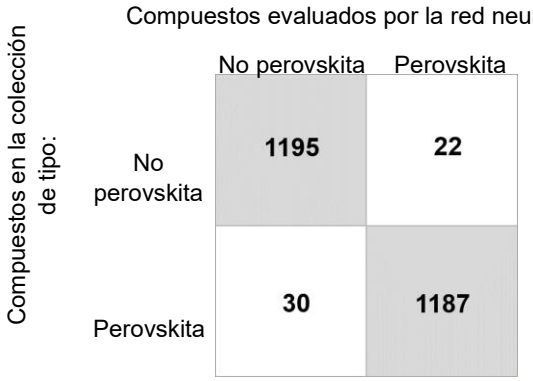


Figura 6.3.2: Matriz de confusión de los compuestos del conjunto TRAVAL de la colección caracterizada con cuatro sitios

Tabla 6.3.1: Métricas de Precisión, Exhaustividad y Valor-F obtenidas con los compuestos del conjunto TRAVAL de la colección caracterizada con cuatro sitios. En esta Tabla se diferencian estas métricas utilizando tanto todos los compuestos como por sitios de Wyckoff.

TODOS LOS COMPUESTOS (TRAVAL)				
	Número de Muestras	**Precisión (%)**	**Exhaustividad (%)**	**Valor-F (%)**
Perovskitas	1217	98.18	97.53	97.86
No perovskitas	1217	97.55	98.19	97.87
COMPUESTOS DE 3 SITIOS DE WYCKOFF				
Perovskitas	458	97.82	98.03	97.93
No perovskitas	471	98.09	97.88	97.98
COMPUESTOS DE 4 SITIOS DE WYCKOFF				
Perovskitas	759	98.40	97.23	97.81
No perovskitas	746	97.22	98.39	97.80

Al utilizar el conjunto de prueba, que fueron muestras diferentes al empleado en el entrenamiento y validación de la red neuronal artificial, las métricas que se obtienen se presentan en la Tabla 6.3.2. Utilizando todos los compuestos sin importar el número de sitios de Wyckoff las precisiones fueron 96.67 y 94.95 % para los compuestos perovskita y no perovskita, respectivamente. Estas precisiones difieren aproximadamente en 2 % de las obtenidas con el conjunto de entrenamiento-validación. Esta diferencia se corresponde con aquella observada en la Figura 6.3.1, con los valores de la función de costo cercanos entre las muestras de entrenamiento y de validación.

Una inspección más detallada en la misma Tabla 6.3.2 muestra que la red neuronal artificial predijo ligeramente mejor aquellos compuestos con tres sitios de Wyckoff que aquellos con cuatro sitios de Wyckoff: Las precisiones en la clasificación de compuestos con tres sitios de Wyckoff fueron 97.92 % y 98.75 % para los tipo perovskita y no perovskita, respectivamente; mientras que aquellos con cuatro sitios de Wyckoff fueron 95.61 y 92.75 % según sean de tipo perovskita o no perovskita. A pesar de estos resultados, se puede afirmar que la red neuronal entrenada cumple con el propósito de inferir nuevos compuestos tipo perovskita.

Los conjuntos de entrenamiento-validación y prueba se originaron de la colección de compuestos tipo perovskita y no perovskita en proporción 1:1. Esta colección se creó agotando todos los compuestos tipo perovskita, con hasta cuatro sitios de Wyckoff, que se encontraron en la base de datos *Crystallography Open Database*. Una vez entrenada y probada la red neuronal artificial, se realizó una segunda evaluación con todos los compuestos no perovskita que aún quedaban disponibles de la base de datos. Para esta segunda evaluación se utilizaron incluso compuestos con uno o dos sitios de Wyckoff en su cristalografía. En total, en esta segunda evaluación se emplearon 14,339 compuestos: 1189 fueron compuestos con un sitio de Wyckoff; 4307, con dos; 5442; con tres; y 3401 con cuatro. Previo a la evaluación con la red neuronal, se construyeron los rasgos descritos en la Tabla 5.3.1 de cada uno de los compuestos y posteriormente se escalaron mediante estandarización. Los resultados que se obtuvieron con la red neuronal después de

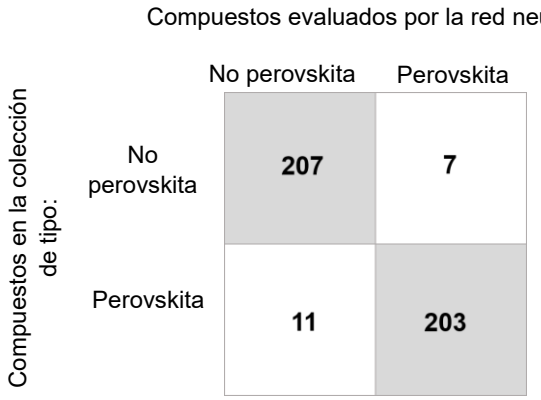


Figura 6.3.3: Matriz de confusión de los compuestos del conjunto TEST de la colección caracterizada con cuatro sitios.

Tabla 6.3.2: Métricas de Precisión, Exhaustividad y Valor-F obtenidas con los compuestos del conjunto TEST de la colección caracterizada con cuatro sitios. En esta Tabla se diferencian estas métricas utilizando tanto todos los compuestos como por sitios de Wyckoff.

TODOS LOS COMPUESTOS (TEST)				
	Número de Muestras	**Precisión (%)**	**Exhaustividad (%)**	**Valor-F (%)**
Perovskitas	214	96.67	94.86	95.75
No perovskitas	214	94.95	96.73	95.83
COMPUESTOS DE 3 SITIOS DE WYCKOFF				
Perovskitas	95	97.92	98.95	98.43
No perovskitas	81	98.75	97.53	98.14
COMPUESTOS DE 4 SITIOS DE WYCKOFF				
Perovskitas	119	95.61	91.60	93.56
No perovskitas	133	92.75	96.24	94.46

esta evaluación se encuentran en la Tabla 6.6.3. Estos resultados corresponden esencialmente de la exhaustividad en la recuperación de los compuestos de tipo no perovskita por la red neuronal después de clasificar.

Los valores de exhaustividad de la Tabla 6.3.3 con los compuestos de tres y cuatro sitios de Wyckoff se corresponden con los valores de la Tabla 6.3.2, sobre el conjunto de prueba, de los compuestos tipo no perovskita con la misma cantidad de sitios de Wyckoff. Cabe mencionar que la exhaustividad en la recuperación de los compuestos tipo no perovskita con uno o dos sitios de Wyckoff fue de 100.00 y 98.91 %. Estos resultados con compuestos de menos de tres sitios de Wyckoff merecen destacarse ya que la red no fue entrenada con compuestos de tipo no perovskita con menos de tres sitios de Wyckoff. En adición, se necesitan al menos tres sitios de Wyckoff en cualquier compuesto con la estructura de perovskita. Este resultado sugiere que la caracterización de los compuestos con los rasgos, plasmados en la Tabla 5.3.1, fue adecuada para el aprendizaje de la red neuronal.

Tabla 6.3.3: Exhaustividad en la recuperación de los compuestos hasta con cuatro sitios de Wyckoff de tipo no perovskita que quedaron disponibles de la base de datos. El número total de compuestos de este tipo utilizados fue 14,339.

COMPUESTOS NO PEROVSKITA			
1 sitio de Wyckoff (1189)	2 sitio de Wyckoff (4307)	3 sitio de Wyckoff (5442)	4 sitio de Wyckoff (3401)
100.00 %	98.91 %	96.31 %	96.09 %

6.3.2 RED NEURONAL ARTIFICIAL PARA COMPUESTOS CARACTERIZADOS CON SEIS SITIOS

La red neuronal que mejor clasificó a los compuestos cristalinos como perovskita o no perovskita tuvo una capa oculta con 1500 nodos; es decir, 10 veces la rasgos que caracterizan a los compuestos. Esta red neuronal tuvo en total 228,001 parámetros:

- 225,000 pesos (elementos de matriz) entre la capa de entrada y la capa oculta, además de los 1500 términos de sesgo de la capa oculta.
- 1500 pesos entre la capa oculta y la capa de salida, además del término de sesgo del nodo en la capa de salida.

La red neuronal fue entrenada con una fracción de Dropout de 0.60. Los valores finales de la función de costo entre los conjuntos de entrenamiento y de validación fueron de 0.1109 y 0.1187, respectivamente (Figura 6.3.4). La diferencia entre ambos valores finales sugiere que la red neuronal apenas fue sobreentrenada.

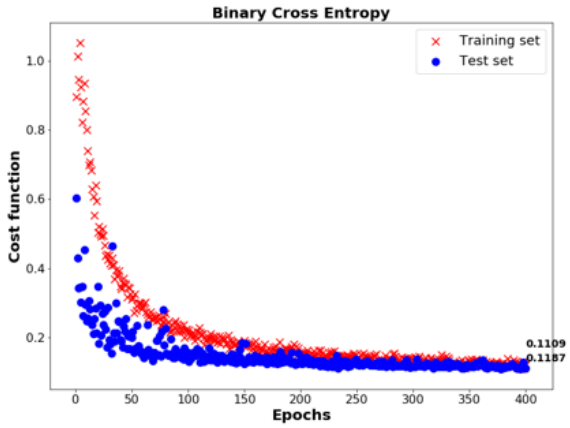


Figura 6.3.4: Gráfica de la función de costo de los conjuntos de entrenamiento y de validación contra las épocas utilizadas en el entrenamiento de la red neuronal artificial. Al término del entrenamiento, el valor de la función de costo fue de 0.1109 y 0.1187 para los conjuntos de entrenamiento y de validación.

En la Figura 6.3.5 se presenta la matriz de confusión que se obtiene con las muestras del conjunto de entrenamiento-validación una vez que la red neuronal fue entrenada. A partir de dicha matriz de confusión, se contruyen las métricas de Precisión, Exhaustividad y Valor-F1 que se presentan en la Tabla 6.3.4. En términos globales, utilizando todos los 1380 compuestos de cada tipo sin importar el número de sitios de Wyckoff, las métricas puntuaron arriba del 97 %. Con los compuestos de tipo perovskita, las precisiones que se alcanzaron fueron: 94.76 %, con 468 compuestos con tres sitios de Wyckoff; 99.05 %, con 748 compuestos con cuatro sitios de Wyckoff; 100.00 %, con 24 compuestos con cinco sitios de Wyckoff; y 99.26

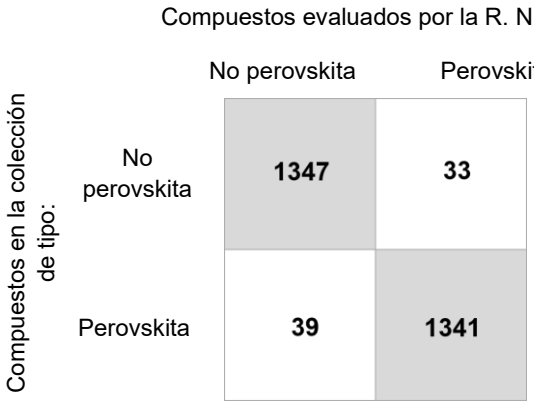


Figura 6.3.5: Matriz de confusión de los compuestos del conjunto TRAVAL de la colección caracterizada con seis sitios.

Tabla 6.3.4: Métricas de Precisión, Exhaustividad y Valor-F obtenidas con los compuestos del conjunto TRAVAL de la colección caracterizada con seis sitios. En esta Tabla se diferencian estas métricas utilizando tanto todos los compuestos como por sitios de Wyckoff.

TODOS LOS COMPUESTOS (TRAVAL)				
	Número de Muestras	**Precisión (%)**	**Exhaustividad (%)**	**Valor-F (%)**
Perovskitas	1380	97.60	97.17	97.39
No perovskitas	1380	97.19	97.61	97.40
COMPUESTOS DE 3 SITIOS DE WYCKOFF				
Perovskitas	468	94.76	96.58	95.66
No perovskitas	481	96.61	94.80	95.70
COMPUESTOS DE 4 SITIOS DE WYCKOFF				
Perovskitas	748	99.05	97.59	98.32
No perovskitas	735	97.59	99.05	98.31
COMPUESTOS CON 5 SITIOS DE WYCKOFF				
Perovskitas	24	100.00	100.00	100.00
No perovskitas	23	100.00	100.00	100.00
COMPUESTOS CON 6 SITIOS DE WYCKOFF				
Perovskitas	140	99.26	96.43	97.83
No perovskitas	141	96.55	99.29	97.90

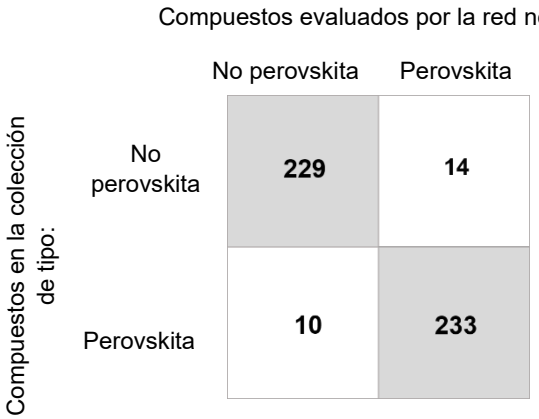


Figura 6.3.6: Matriz de confusión de los compuestos del conjunto TEST de la colección caracterizada con seis sitios.

Tabla 6.3.5: Métricas de Precisión, Exhaustividad y Valor-F obtenidas con los compuestos del conjunto TEST de la colección caracterizada con seis sitios. En esta Tabla se diferencian estas métricas utilizando tanto todos los compuestos como por sitios de Wyckoff.

TODOS LOS COMPUESTOS (TEST)				
	Número de Muestras	**Precisión (%)**	**Exhaustividad (%)**	**Valor-F (%)**
Perovskitas	243	94.33	95.88	95.10
No perovskitas	243	95.82	94.24	95.02
COMPUESTOS DE 3 SITIOS DE WYCKOFF				
Perovskitas	85	92.22	97.65	94.86
No perovskitas	73	97.06	90.41	93.62
COMPUESTOS DE 4 SITIOS DE WYCKOFF				
Perovskitas	130	96.12	95.38	95.75
No perovskitas	141	95.77	96.45	96.11
COMPUESTOS CON 5 SITIOS DE WYCKOFF				
Perovskitas	3	100.00	66.67	80.00
No perovskitas	4	80.00	100.00	88.89
COMPUESTOS CON 6 SITIOS DE WYCKOFF				
Perovskitas	25	92.31	96.00	94.12
No perovskitas	25	95.83	92.00	93.88

% con 140 compuestos con seis sitios de Wyckoff. Por otro lado, las precisiones que consiguieron con los compuestos de tipo no perovskita de este conjunto de entrenamiento-validación fueron: 96.61 %, con 481 compuestos de tres sitios de Wyckoff; 97.59 %, con 735 compuestos de cuatro sitios de Wyckoff; 100.00%, con 23 compuestos de cinco sitios de Wyckoff; y 96.55 %, con 141 compuestos de seis sitios de Wyckoff.

La matriz de confusión que se genera al clasificar las muestras del conjunto de prueba se muestra en la Figura 6.3.6. Estas muestras fueron diferentes a las usadas para optimizar la red neuronal. A partir de dicha matriz de confusión se contruyó la Tabla 6.3.5. Las precisiones alcanzadas con los compuestos tipo perovskita y no perovskita, sin importar la cantidad de sitios de Wyckoff que los compuestos tuvieran, fueron de 94.33 % y 95.82 %. La diferencia entre los valores mencionados con los reportados en el conjunto de entrenamiento-validación fueron de 3.27 y 1.37 %, respectivamente. Esta pequeña diferencia se corresponde con la la encontrada entre los valores de la función de costo de la Figura 6.3.4.

Una inspección más detallada en la precisión de los compuestos tipo perovskita del conjunto de prueba muestra que los valores de esta métrica no fueron homogéneos en términos de sitios de Wyckoff. La compuestos tipo perovskita con cuatro (85 compuestos) y seis sitios (25 compuestos) de Wyckoff tuvieron precisiones cercanas, de 92.22 y 92.31 %, respectivamente. La diferencia es de casi 7 % entre los compuestos de seis sitios de Wyckoff de los conjuntos entrenamiento-validación y de prueba. Con los 130 compuestos de cuatro sitios de Wyckoff alcanzada fue de 96.12 %, mientras que con los tres únicos compuestos de cinco sitios de Wyckoff del conjunto de prueba se consiguió una precisión del 100.00 %. Para los compuestos de tipo no perovskita la situación en términos de sitios de Wyckoff fue un poco más homogénea, a excepción de los compuestos con cinco sitios de Wyckoff, que fueron cuatro. Con estos compuestos se consiguió una precisión del 80 %. Con los 141 compuestos de cuatro sitios de Wyckoff se consiguió una precisión de 95.77 %; con los 25 compuestos de seis sitios de

Wyckoff se consiguió una precisión de 95.83 %; mientras que con 73 compuestos de tres sitos de Wyckoff la precisión fue de 97.06 %.

Una vez entrenada y probada la red neuronal se procedió a evaluarla con 21,418 compuestos de tipo no perovskita que todavía estaban disponibles de la base de datos. En la Tabla 6.3.6 se muestran los resultados de exhaustividad de este tipo de compuestos, con base en su número de sitios de Wyckoff. La valores de exhaustividad obtenidos fueron: 100.00 %, con 1189 compuestos de un sitio de Wyckoff; 96.68 %, con 4306 compuestos de dos sitios de Wyckoff; 90.83 %, con 5440 compuestos de tres sitios de Wyckoff; 96.89 %, de 3404 compuestos de 4 sitios de Wyckoff; 85.32 %, con 3625 compuestos de cinco sitios de Wyckoff; y 95.63 %, con 3454 compuestos de seis sitios de Wyckoff. Dichos valores de exhaustividad fueron cercanos a los reportados en la Tabla 6.3.5 (conjunto de prueba) para los compuestos de tipo no perovskita de tres, cuatro y seis sitios de Wyckoff. La disparidad entre los valores de las Tablas 6.3.5, de exhaustividad, y 6.3.6 de los compuestos de cinco sitios de Wyckoff sugiere que se debe incorporar más compuestos con esta cantidad de sitios originales en la colección. Adicionalmente, se puede sugerir que los rasgos utilizados en la caracterización de los compuestos fueron adecuados para que la red neuronal "*aprendiera*" que aquellos con menos de tres sitios de Wyckoff son de tipo no perovskita.

Tabla 6.3.6: Exhaustividad en la recuperación de los compuestos tipo no perovskita hasta con seis sitios de Wyckoff que quedaron disponibles de la base de datos. El número total de compuestos de este tipo utilizados fue 21,418.

COMPUESTOS NO PEROVSKITA					
1 sitio de Wyckoff (1189)	2 sitio de Wyckoff (4306)	3 sitio de Wyckoff (5440)	4 sitio de Wyckoff (3404)	5 sitios de Wyckoff (3625)	6 sitios de Wyckoff (3454)
100.00 %	96.68 %	90.83 %	96.89 %	85.32 %	95.63 %

3.6.3 RED NEURONAL ARTIFICIAL PARA COMPUESTOS CARACTERIZADOS CON OCHO SITIOS

La colección utilizada para entrenar esta red neuronal diferió, de aquella que caracterizó a los compuestos con seis sitios, en 12 compuestos. La caracterización de los compuestos cristalinos con dos sitios más dio lugar a que estos tuvieran 462 rasgos. La mejor red neuronal tuvo dos capas ocultas: la primera con 924 nodos y la segunda con 1386. Así, esta red neuronal tuvo ’7 , 49 parámetros para optimizar:

- 426,888 pesos (elementos de matriz) entre la capa de entrada y la primera capa oculta, además de 924 términos de sesgo en la primera capa oculta.
- ’ 8 ,664 pesos entre las capas ocultas, además de 386 términos de sesgo en la segunda capa oculta.
- 1386 pesos entre la segunda capa oculta y la capa de salida, además del término de sesgo en el único nodo de la capa de salida.

Debido a la gran cantidad de parámetros de esta red neuronal, fue necesario introducir una fracción de Dropout de 0.70. En otras palabras, el 70 % de los nodos de la capa oculta fueron “removidos” en cada bloque de una iteración. Lo anterior dio lugar a que la cantidad de épocas se extendiera hasta 800. Al final del entrenamiento y validación de la red neuronal, los valores de la función de costo fueron 0.2417 y 0.2692, respectivamente (Figura 6.3.7). Estos valores sugieren que las métricas que se obtuvieron con esta colección no serán tan altas como aquellas obtenidas con las colecciones de compuestos caracterizados con cuatro y seis sitios. Sin embargo, la pequeña diferencia entre los valores finales de la función de costo sugieren que hubo poco sobreentremiento de la red neuronal.

En la Figura 6.3.8 se muestra la matriz de confusión generada al clasificar las muestras del conjunto de entrenamiento-validación después de haber entrenado la red neuronal. Este conjunto tuvo en total 2770 compuestos, en proporción 1:1 entre perovskitas y no perovskitas. A partir de la matriz de confusión se calcularon las métricas de Precisión, Exhaustividad y Valor-F1. En términos generales, sin importar el número de sitios de Wyckoff, la precisión de los compuestos tipo

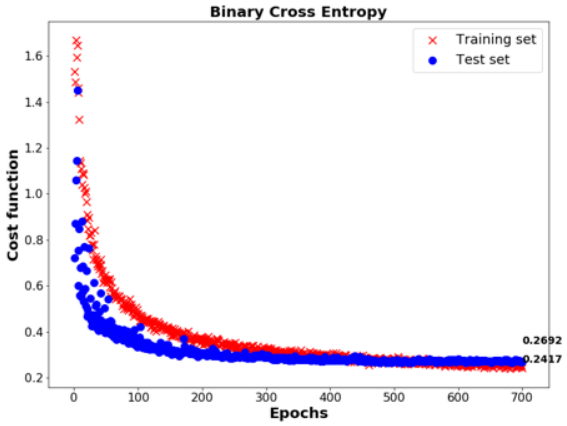


Figura 6.3.7: Gráfica de la función de costo de los conjuntos de entrenamiento y de validación contra las épocas utilizadas en el entrenamiento de la red neuronal artificial. Al término del entrenamiento, el valor de la función de costo fue de 0.2417 y 0.2692 para los conjuntos de entrenamiento y de validación.

perovskita fue de 94.18 %, mientras aquella de los compuestos de tipo no perovskita fue de 90.46 %. Aquí merece la pena comentar que esta colección que caracteriza a los compuestos con ocho sitios es, esencialmente, la misma que aquella que caracteriza a los compuestos con seis sitios. Por lo anterior, llama la atención esta disminución en la precisión, que se puede comparar con la presentada en la Tabla 6.3.4 para el conjunto de entrenamiento-validación. Este hecho sugiere que se necesitan enriquecer la colección de compuestos de ocho sitios con más muestras. En efecto, cuando se compara los Valores-F1 de los compuestos perovskita y no perovskita de las Tablas 6.3.4 y 6.3.7 se observa que, en promedio, la disminución en los puntajes es del 5%.

Una inspección más detallada en la Tabla 6.3.7 muestra que la incorporación de los compuestos de ocho sitios de Wyckoff no fue suficiente para que la red neuronal aprendiera a predecir compuestos de tipo perovskita con esa cantidad de sitios de Wyckoff. Con los compuestos de tipo no perovskita, la precisión obtenida corresponde al escenario de *adivinar*, ya que la precisión obtenida fue cercana al 50 %. No obstante, la exhaustividad con los seis compuestos de tipo no perovskita fue de 100 %.

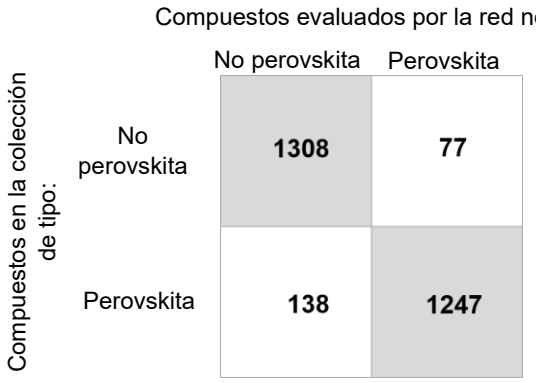


Figura 6.3.8: Matriz de confusión de los compuestos del conjunto TRAVAL de la colección caracterizada con ocho sitios.

Tabla 6.3.7: Métricas de Precisión, Exhaustividad y Valor-F obtenidas con los compuestos del conjunto TRAVAL de la colección caracterizada con ocho sitios. En esta Tabla se diferencian estas métricas utilizando tanto todos los compuestos como aquellos por sitios de Wyckoff.

TODOS LOS COMPUESTOS (TRAVAL)				
	Número de Muestras	**Precisión (%)**	**Exhaustividad (%)**	**Valor-F (%)**
Perovskitas	1385	94.18	90.04	92.06
No perovskitas	1385	90.46	94.44	92.41
COMPUESTOS DE 3 SITIOS DE WYCKOFF				
Perovskitas	470	88.42	84.47	86.40
No perovskitas	468	85.07	88.89	86.94
COMPUESTOS DE 4 SITIOS DE WYCKOFF				
Perovskitas	745	96.65	93.83	95.36
No perovskitas	745	94.02	97.05	95.51
COMPUESTOS CON 5 SITIOS DE WYCKOFF				
Perovskitas	24	87.50	87.50	87.50
No perovskitas	26	88.46	88.46	88.46
COMPUESTOS CON 6 SITIOS DE WYCKOFF				
Perovskitas	141	100.00	92.20	95.94
No perovskitas	140	92.75	100.00	96.22
COMPUESTOS CON 8 SITIOS DE WYCKOFF				
Perovskitas	5	0.00	0.00	0.00
No perovskitas	6	54.55	100.00	70.59

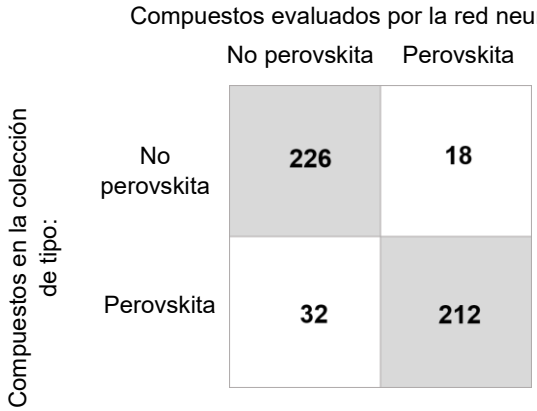


Figura 6.3.9: Matriz de confusión de los compuestos del conjunto TEST de la colección caracterizada con ocho sitios.

Tabla 6.3.8: Métricas de Precisión, Exhaustividad y Valor-F obtenidas con los compuestos del conjunto TEST de la colección caracterizada con ocho sitios. En esta Tabla se diferencian estas métricas utilizando tanto todos los compuestos como aquellos por sitios de Wyckoff.

TODOS LOS COMPUESTOS (TEST)				
	Número de Muestras	**Precisión (%)**	**Exhaustividad (%)**	**Valor-F (%)**
Perovskitas	244	92.17	86.89	89.45
No perovskitas	244	87.60	92.62	90.04
COMPUESTOS DE 3 SITIOS DE WYCKOFF				
Perovskitas	83	81.33	73.49	77.22
No perovskitas	85	76.34	83.53	79.78
COMPUESTOS DE 4 SITIOS DE WYCKOFF				
Perovskitas	133	96.92	94.74	95.82
No perovskitas	132	94.81	96.97	95.88
COMPUESTOS CON 5 SITIOS DE WYCKOFF				
Perovskitas	3	100.00	100.00	100.00
No perovskitas	1	100.00	100.00	100.00
COMPUESTOS CON 6 SITIOS DE WYCKOFF				
Perovskitas	24	100.00	91.67	95.65
No perovskitas	26	92.86	100.00	96.30
COMPUESTOS CON 8 SITIOS DE WYCKOFF				
Perovskitas	1	0.00	0.00	0.00
No perovskitas	0	0.00	0.00	0.00

En adición a lo ya comentado para los compuestos de ocho sitios de Wyckoff del conjunto de entrenamiento-validación, la precisión que se obtuvo con los compuestos de tipo perovskita en términos del número de sitios de Wyckoff fue la siguiente: 88.42 %, con 470 compuestos originalmente de tres sitios; 96.65 %, con compuestos originalmente de cuatro sitios; 87.50 %, con compuestos originalmente de cinco sitios; y 100.00 %, con compuestos originalmente de seis sitios. Con los compuestos de tipo no perovskita se obtuvieron los siguientes valores de precisión: 85.07%, con compuestos originalmente de tres sitios; 94.02 %, con compuestos originalmente de cuatro sitios; 88.46 %, con compuestos originalmente de cinco sitios; y 92.75 %, con compuestos originalmente de seis sitios.

En la Figura 6.3.9 se muestra la matriz de confusión de los compuestos del conjunto de prueba. A partir de esta Figura, se construyeron las métricas que se presentan en la Tabla 6.3.8. En general, la precisión alcanzada con los compuestos tipo perovskita y no perovskita fue 92.17 y 87.60 %, respectivamente. Una inspección más detallada en la Tabla 6.3.8 muestra los siguientes valores de precisión con los compuestos de tipo perovsktia en términos de sitios de Wyckoff: 81.33 %, con los compuestos de tres sitios; 96.92 %, con los compuestos de cuatro sitios; 100.00 %, con los compuestos de cinco sitios; 100.00 %, con los compuestos de seis sitios; y 0.00 %, con el único compuesto de ocho sitios. Analógamente, los valores de precisión obtenidos con los compuestos de tipo no perovskita fueron: 76.34 %, con los compuestos de tres sitios; 94.81 %, con los compuestos de cuatro sitios; 100.00 %, con los compuestos de cinco sitios; y 92.86 %, con los compuestos de seis sitios. En este conjunto de prueba, no hubo compuestos de tipo no perovskita con ocho sitios de Wyckoff. Esto se debe al algoritmo utilizado es la selección de las muestras.

Con base en lo anteriormente presentado para el conjunto de prueba, se puede observar que los resultados obtenidos con los compuestos de cuatro sitios de Wyckoff fueron esencialmente iguales a los mostrados en la Tabla 6.3.5. No obstante, también se observó que hubo un detrimento en las métricas utilizando compuestos con tres sitios de Wyckoff.

Después de entrenar, validar y probar la red neuronal, se evaluó con todos los compuestos de tipo no perovskita con hasta ocho sitios de Wyckoff todavía disponibles de la base de datos. En total, se utilizaron 27,486 compuestos para esta segunda evaluación. Los resultados se muestran en la Tabla 6.3.9, los cuales corresponden a exhaustividad. Cabe mencionar que durante el entrenamiento de la red neuronal no se utilizaron compuestos de tipo no perovskita con uno, dos o siete sitios de Wyckoff. No obstante, los valores de exhaustividad obtenidos fueron de 100.00 %, 99.93 % y 99.88 %, respectivamente. En adición, para los otros compuestos con diferente número de sitios de Wyckoff fueron: 86.05 %, con compuestos de tres sitios de Wyckoff; 94.86 %, con compuestos de cuatro sitios; 85.49 %, con compuestos de cinco sitios; 94.99 %, con compuestos de seis sitios; y 97.52 %, con compuestos de ocho sitios de Wyckoff. Estos valores obtenidos son cercanos a los de exhaustividad de los compuestos de tipo no perovskita presentados tanto en la Tabla 6.3.7 como en la Tabla 6.3.8.

Tabla 6.3.9: Exhaustividad en la recuperación de los compuestos hasta con ocho sitios de Wyckoff de tipo no perovskita que quedaron disponibles de la base de datos. El número total de compuestos de este tipo utilizados fue 27,486.

COMPUESTOS NO PEROVSKITA							
1 sitio de Wyckoff (1189)	2 sitio de Wyckoff (4306)	3 sitio de Wyckoff (5441)	4 sitio de Wyckoff (3403)	5 sitios de Wyckoff (3625)	6 sitios de Wyckoff (3454)	7 sitios de Wyckoff (3402)	8 sitios de Wyckoff (2666)
100.00 %	99.93 %	86.05 %	94.86 %	85.49 %	94.99 %	99.88 %	97.52 %

6.4 INFLUENCIA DE LOS RASGOS EN EL DESEMPEÑO DE LAS REDES NEURONALES ARTIFICIALES

Las métricas obtenidas en las pruebas hechas a las redes neuronales desarrolladas en este trabajo sugieren que hay correlación entre la caracterización de los compuestos cristalinos aquí implementada y el tipo de estructura cristalina (perovskita). Los rasgos construidos para cada compuesto cristalino tienen su origen en los factores geométricos y de empaquetamiento descritos en la sección 1.2.1 y en los entornos de los átomos que comparten un mismo sitio de Wyckoff (sección 1.1.3). De esta manera, el vector de datos de entrada de un compuesto cristalino alimentado a la red neuronal contiene información esencialmente de carácter estructural.

La caracterización implementada en este trabajo permitió que las redes neuronales evaluaran la probabilidad de un compuesto de cristalizar como perovskita no sólo en términos de su composición, sino también de su posible arreglo atómico. Para calcular los rasgos relacionados a los factores geométricos y de empaquetamiento bastó con conocer qué átomos ocuparon los sitios de Wyckoff, es decir, la composición promedio de éstos. En cambio, el cálculo de los rasgos que se derivaron de las funciones de localidad necesitó información sobre el arreglo de los átomos dentro de la celda unitaria. De esta manera, se pueden diseñar diferentes propuestas de compuestos, en términos de composición y su estructura, para ser evaluados con las redes neuronales desarrolladas. En adición, las evaluaciones hechas por las redes neuronales fueron más rápidas que las de los cálculos mecanocuánticos.

Por otro lado, la construcción de los rasgos de cada compuesto cristalino tiene un carácter combinacional (subsección 5.3.2). Dependiendo el número de sitios con el que se caracteriza una colección de compuestos cristalinos:

1. Se puede manejar compuestos cristalinos con hasta cierto número de sitios de Wyckoff y

2. Se obtiene una cantidad diferente de rasgos que depende del número de combinaciones que se pueden formar.

En el caso de la caracterización con cuatro sitios, el vector de entrada tuvo 33 componentes o rasgos; con seis sitios, 150; y con ocho sitios, 462. Este enfoque combinatorio tuvo la finalidad de normar la cantidad de rasgos en la caracterización de compuestos cristalinos con una cantidad definida de sitios, de manera que sean generales para cualquier tipo de estructura cristalina. Así, el usuario del código *patolli.py* no tendría que detenerse en la selección de los rasgos del vector de entrada. La importancia de los rasgos del vector de entrada se va adaptando durante el entrenamiento de la red neuronal, mediante la optimización de los pesos que conectan a los nodos entre capas. No obstante, es útil revisar qué rasgos tienen más influencia en la evaluación que hace una red neuronal.

Previo a la revisión sobre la influencia de los rasgos, cabe mencionar que en el Apéndice D se encuentran los histogramas de los rasgos utilizados en las colecciones de cuatro, seis y ocho sitios, junto con su promedio y desviación estándar. Dichos promedios y desviaciones estándar son adimensionales y fueron utilizados para la estandarización de los datos de entrada.

En la Figura 6.4.1 se muestra la matriz de correlación entre los rasgos con los que se caracterizó a los compuestos de la colección con cuatro sitios. Los elementos de matriz corresponden al coeficiente de correlación lineal de Pearson entre dos rasgos x_i y x_j, que se define como [88]:

$$\rho_{x_i,x_j} = \frac{cov(x_i, x_j)}{\sigma_{x_i}\sigma_{x_j}}$$

Donde $cov(x_i, x_j)$ es la covarianza entre los rasgos y σ es la desviación estándar del rasgo. El coeficiente de correlación de Pearson está constreñido al rango $[-1,1]$, donde 1 (azul intenso en la Figura 6.4.1) corresponde a una relación

proporcional y directa, mientras que -1 (rojo intenso en la Figura 6.4.1) corresponde a una relación proporcional e inversa.

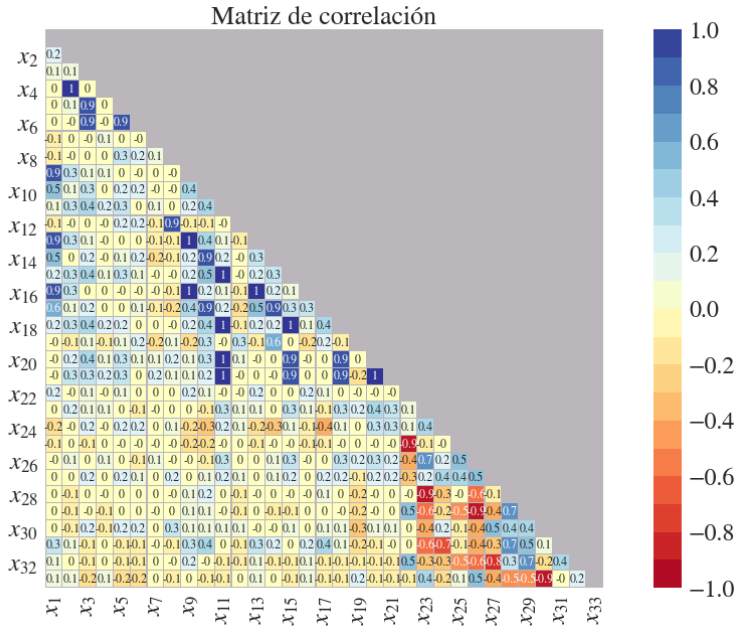


Figura 6.4.1: Matriz de correlación de Pearson de los datos de entrada de la colección de cuatro sitios.

En la colección de cuatro sitios no se tuvo ninguna correlación entre dos rasgos igual a la unidad, en términos absolutos. Los coeficientes de correlación más altos se obtuvieron entre los pares $x_{18}\left(\frac{r_1+r_4}{r_3+r_4}\right)$ y $x_{15}\left(\frac{r_1+r_3}{r_3+r_4}\right)$, con $\rho = 0.9881$; x_{18} y $x_{11}\left(\frac{r_1+r_2}{r_3+r_4}\right)$, $\rho = 0.9850$; y x_{15} y x_{11}, $\rho = 0.9845$, que correspondieron a rasgos de empaquetamiento. En adición a lo anterior, el rasgo x_{11} fue el que más correlación presentó con otros: con $x_{20}\left(\frac{r_2+r_3}{r_3+r_4}\right)$, $\rho = 0.9558$, y $x_{21}\left(\frac{r_2+r_4}{r_3+r_4}\right)$, $\rho = 0.9556$. Otros pares que tuvieron alta correlación fueron: $x_{13}\left(\frac{r_1+r_3}{r_2+r_3}\right)$ y $x_9\left(\frac{r_1+r_2}{r_2+r_3}\right)$, $\rho = 0.9694$; $x_{16}\left(\frac{r_1+r_4}{r_2+r_3}\right)$ y x_9, $\rho = 0.9734$, por lo tanto x_{16} y x_{13} estuvieron correlacionados con $\rho = 0.9789$; y el par $x_4\left(\frac{r_2}{r_3}\right)$ y $x_2\left(\frac{r_1}{r_3}\right)$, $\rho = 0.9520$. Con base en lo anterior, y estableciendo un criterio donde no haya una correlación entre dos rasgos mayor a 0.95, se puede reducir la dimensionalidad del vector de entrada en la colección de

cuatro sitios, al prescindir de siete de los diez rasgos mencionados en este párrafo, x_4, x_{13}, x_{15}, x_{16}, x_{18}, x_{20} y x_{21}.

En el Apéndice D se señalan con un asterisco los rasgos que presentaron una correlación menor a 0.95 para cada una de las colecciones. En la colección de seis sitios, que se caracterizó con un vector de 150 rasgos, se tuvieron 182 pares de rasgos con una correlación mayor a 0.95. Al eliminar aquellos rasgos redundantes, el vector de dicha colección puede reducirse a una dimensión de 80. Adicionalmente, en la colección de ocho sitios, caracterizada por un vector de 462 rasgos, se tuvieron 2611 pares con una correlación, en términos absolutos, mayor a 0.95. Dicho vector de datos de entrada puede reducirse a una dimensión de 103 al eliminar los rasgos redundantes. Debido a la cantidad de rasgos de los vectores de entrada de las colecciones de seis y ocho sitios, sus matrices de correlación fueron enormes y esa información se presenta de manera segmentada en el Apéndice E.

Las observaciones anteriores sobre la reducción de la dimensionalidad de los vectores de entrada de las colecciones se presentan a manera de sugerencia ya que, parte de que exista muchos pares de rasgos con alta correlación, se debe a la poca presencia de muestras con cierta cantidad de sitios de Wyckoff en las colecciones. Esto se hace evidente sobre todo en la colección de compuestos caracterizada con ocho sitios: debido a la poca presencia de compuestos con ocho sitios de Wyckoff en la colección, el promedio de los rasgos es cercano a cero y con una distribución muy estrecha. Por lo anterior, se puede esperar que varios pares de rasgos no presenten alta correlación al incrementar la cantidad de muestras con ese número de sitios de Wyckoff.

Se estudió la influencia de los rasgos en la precisión en la clasificación de las redes neuronales desarrolladas. Para esto, se igualaron a cero uno o un par de rasgos del vector de entrada de las muestras del conjunto TRAVAL, simultáneamente. Lo anterior es equivalente a ocultar ciertos rasgos del vector de entrada y revisar el efecto que tiene en las evaluaciones hechas por las redes neuronales. Los resultados de la red neuronal entrenada con la colección de cuatro

sitios se encuentran en las Figuras 6.4.2 y 6.4.3, para compuestos tipo perovskita y no perovskita, respectivamente. En dichas Figuras, la información se muestra de manera similar a la matriz de correlación. No obstante, la diagonal principal sí se representa y corresponde al ocultamiento (igualación a cero) de uno de los rasgos, mientras que los elementos debajo de la diagonal principal corresponden al ocultamiento de un par de rasgos.

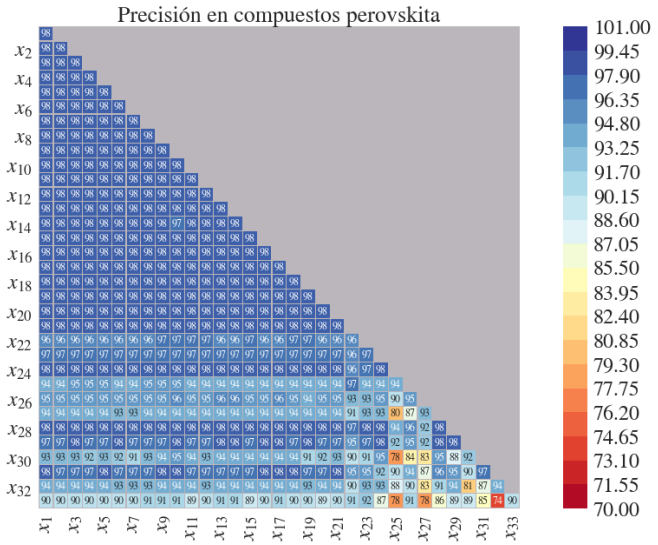


Figura 6.4.2: Se presenta la influencia de eliminar un rasgo, x_i, o un par de éstos en la precisión en la clasificación de los compuestos tipo perovskita con la red neuronal de cuatro sitios. En la diagonal de esta matriz se presenta el efecto de oculta uno de los rasgos, mientras que debajo de la diagonal el efecto es el que se consigue de oculta un par de rasgos.

El efecto que tuvo ocultar un rasgo del vector de entrada de la red neuronal de cuatro sitios fue distinto entre los compuestos tipo perovskita y no perovskita. Adicionalmente, las precisiones obtenidas de ocultar un rasgo relacionado con algún factor geométrico o de empaquetamiento fueron esencialmente los mismos (97 – 98 %) a los reportados en la Tabla 6.3.1, pero no con aquellos relacionados con las funciones de localidad. En los compuestos de tipo perovskita, prescindir de los rasgos $x_{33}(f_{43})$, $x_{30}(f_{34})$ y $x_{27}(f_{24})$ disminuyó la precisión a 90.04 %, 92.06 % y 93.38 %, respectivamente; mientras que en los compuestos de tipo no perovskita la precisión disminuyó en 67.60, 73.92, 75.81 y 77.10 % al prescindir de los rasgos x_{30}, x_{32} (f_{42}), x_{33} y x_{27}, respectivamente.

Al prescindir de un par de rasgos el efecto que se obtuvo fue mayor con los rasgos derivados de las funciones de localidad. Con las muestras de tipo perovskita, las precisiones más bajas que se obtuvieron fueron 74 %, al prescindir de x_{32} y x_{33}, simultáneamente; y 78 %, con los pares x_{25} (f_{21}) y x_{30}, x_{25} y x_{33}, y x_{27} y x_{33}. Con las muestras de tipo no perovskita las precisiones más bajas que se obtuvieron fueron 52 %, al eliminar el par x_{32} y x_{33}; 56 %, sin el par x_{28} (f_{31}) y x_{32}; y 59 %, sin los pares x_{28} y x_{33}, x_{29} (f_{32}) y x_{30}, y x_{31} y x_{32}. Además, la precisión en la clasificación de compuestos tipo no perovskita disminuyó notablemente al ocultar cualquier par que involucrara a los rasgos x_{27}, x_{30}, x_{32} y x_{33}.

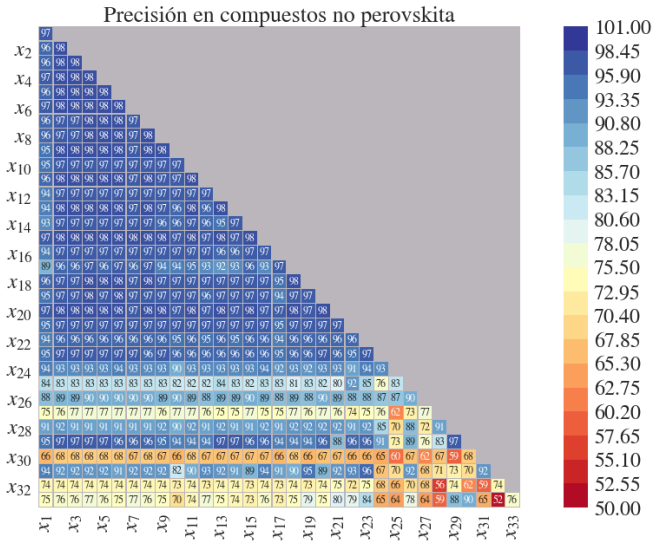


Figura 6.4.3: Se presenta la influencia de eliminar un rasgo, x_i, o un par de éstos en la precisión en la clasificación de los compuestos tipo no perovskita con la red neuronal de cuatro sitios. En la diagonal de esta matriz se presenta el efecto de oculta uno de los rasgos, mientras que debajo de la diagonal el efecto es el que se consigue de oculta un par de rasgos.

Los resultados anteriores sugieren que los rasgos relacionados a las funciones de localidad tienen mayor influencia en las decisiones de la red neuronal desarrollada que aquellos relacionados con los factores geométricos y de empaquetamiento. Esto también se observó en las redes neuronales de seis y ocho sitios, y se puede consultar en el Apéndice E.

6.5 INFERENCIA DE NUEVOS COMPUESTOS CON ESTRUCTURA PEROVSKITA

Con el código *simulate_compounds.py* se usó la red neuronal desarrollada de cuatro sitios para evaluar la probabilidad de que un compuesto cristalizara con la estructura aristotípica de la perovskita. Con este código se evaluaron diferentes propuestas de halogenuros ternarios ABX_3, Figura 6.5.1, (fluoruros, cloruros, bromuros y yoduros), donde A y B corresponden a los elementos en los sitios octaédrico y cuboctaédrico, respectivamente; mientras que X corresponde al halogenuro. Adicionalmente se probó con un mismo par de elementos que éstos ocuparan tanto los sitios octaédricos como cuboctaédricos de manera alternada. En total y, considerando la alternancia en la ocupación entre los sitios de las geometrías mencionadas, se evaluaron 696 halogenuros de cada tipo; es decir, 2784 compuestos diferentes. Los resultados de las inferencias hechas por la red neuronal se guardaron como diccionarios de Numpy y se pueden consultar en https://github.com/gomezperalta/phd_thesis/tree/master/Inferencias_ANN.

Aun cuando la red neuronal no toma en cuenta la información sobre los estados de oxidación de cada elemento, las propuestas evaluadas con el código *simulate_compounds.py* se diseñaron de manera que la combinación de elementos en la fórmula cumpliera la electroneutralidad: la suma de los estados de oxidación de los elementos de la fórmula debe ser cero. En efecto, el conjunto de elementos señalados con un círculo en azul, de la Figura 6.5.1, se propuso con un estado de oxidación de 1+, mientras que aquel conjunto de elementos señalados con un rombo en rojo se asumió que tenía un estado de oxidación de 2+.

Como se comentó en la metodología, para cada compuesto ABX_3 se utilizaron cinco parámetros de red diferentes. A este binomio de compuesto con parámetro de red se refiere en esta tesis como *propuesta*. De esta manera, para tipo de halogenuro se tuvieron 3480 propuestas inicialmente. Las propuestas que la red neuronal evaluó con probabilidad mayor o igual a 0.5 se consideraron como candidatos a cristalizar con la estructura tipo perovskita o, simplemente

predicciones. Después de evaluar las propuestas con la red neuronal el número de predicciones de compuestos fueron 3452, 3308, 3296 y 3001 de los tipos fluoruro, cloruro, bromuro y yoduro; es decir, se tuvieron en total 13,057 predicciones para ser validadas con cálculos de energía de un solo punto. En adición, la red neuronal descartó todas las propuestas de los compuestos Au*M*I_3; con *M* = Co, Hg, Ir, Mo, Ni, Os, Pd, Pt, Rh, Ru y Tc; así como PdAgI_3 y RhAuI_3. Así, se tuvieron 683 yoduros de tipo ABX_3 para ser validados con los cálculos químico cuánticos. En relación a los demás tipos de halogenuro, no se descartó por completo alguna composición, por lo que al menos hubo una predicción de la red neuronal con alguna de las cinco propuestas iniciales hechas para un mismo compuesto.

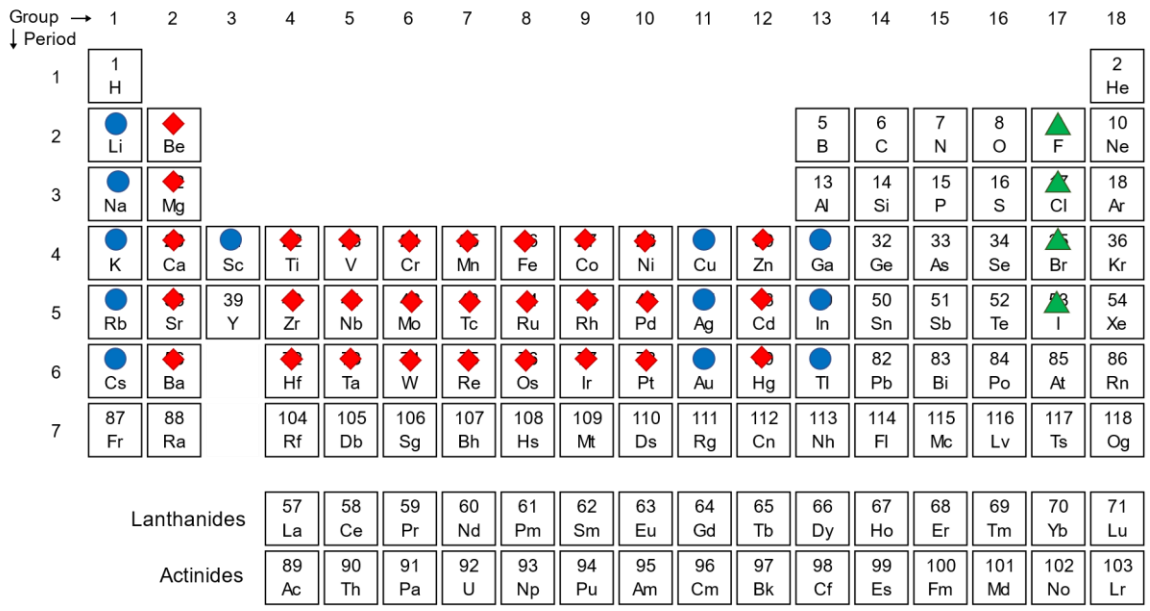


Figura 6.5.1: Elementos de la tabla periódica considerados en la propuesta de compuestos ABX_3 para su evaluación con la red neuronal. Para la formulación de dichas propuestas, se asumió que los elementos señalados con un círculo azul tenían un estado de oxidación 1+, mientras que aquellos con un rombo rojo tenían un estado de oxidación de 2+. Los elementos de ambos conjuntos ocuparon de manera alternada los sitios octaédrico y cuboctaédrico, lo cual dio origen a 720 compuestos diferentes para ser evaluados con la red neuronal.

Tras la implementación de los cálculos de energía de un solo punto, se tuvieron 4577 predicciones que convergieron de las 13057 hechas por la red neuronal. Esta cantidad de predicciones correspondió a 1373 compuestos diferentes: 316 fluoruros (1150 predicciones validadas), 337 cloruros (1132), 363

bromuros (1164), y 357 yoduros (1131). Este hecho, de que aproximadamente una tercera parte de las predicciones hechas por la red neuronal hayan sido validadas, refleja la importancia de implementar cálculos químico cuánticos pese a las buenas métricas obtenidas con la red neuronal: la red neuronal evalúa las propuestas de un compuesto cristalino de manera rápida y en términos estructurales; mientras que los cálculos químico cuánticos lo hacen desde un punto de vista electrónico. La lista de las predicciones que convergieron en los cálculos de energía de un solo punto se encuentra en el Apéndice F.

Por razones de tiempo de cómputo, sólo se implementó la optimización de la geometría con condiciones periódicas en la frontera de 874 de los 1373 compuestos tipo halogenuro. Para un compuesto ABX_3, se eligió la predicción con el parámetro de red con el que obtuvo la menor energía en los cálculos de energía de un solo punto. Tras esta validación, únicamente cien de los 874 compuestos no convergieron.

De los 774 compuestos que sí convergieron en este tipo de cálculos, se tuvieron 135 compuestos que convergieron sin conservar la estructura tipo perovskita; es decir, el marco de octaedros de vértice compartido no se preservó después de la optimización; 338 compuestos conservaron la estructura aristotípica de la perovskita, mientras que 301 compuestos adoptaron una estructura tipo perovskita distorsionada. La lista de los compuestos que convergieron con estructura perovskita, en cálculos con condiciones periódicas en la frontera, se encuentran en las Tablas 6.5.1 a 6.5.3. La división de los compuestos que convergieron en estructura perovskita obedece al elemento alojado en el sitio octaédrico: en la Tabla 6.5.1 se enlistan los compuestos donde dicho sitio está ocupado por elementos del bloque *s* de la tabla periódica; en la 6.5.2, se enlistan los compuestos con elementos del bloque *d* a excepción de los elementos los grupos 11 y 12 (cobre, plata, oro, zinc, cadmio y mercurio), los cuales aparecen junto al galio, indio y talio en la Tabla 6.5.3. En estas Tablas se presenta, además, el valor de la brecha energética entre bandas de los compuestos. Las brechas energéticas de los compuestos ABX_3 se grafican en función del par de elementos

A, B en las Figuras 6.5.2 – 6.5.5, según el tipo de halogenuro. Entre los compuestos de las Tablas 6.5.1 – 6.5.3 con estructura perovskita distorsionada, aquellos con los grupos espaciales 1 (sistema cristalino triclínico), 4 a 7 y 9 (monoclínico), 25 (ortorrómbico) y, 146 y 160 (trigonal) son candidatos a tener aplicación como materiales piezoeléctricos, piroeléctricos y de polarización de la luz (a excepción del grupo espacial 160).

En general, los compuestos de las Tablas 6.5.1 y 6.5.3 conservaron el grupo espacial 221; mientras que aquellos de la Tabla 6.5.2 presentaron distorsión en la estructura tipo perovskita tras la optimización de la geometría. El origen de esta distorsión en los compuestos de la Tabla 6.5.2 puede estar relacionado con la presencia de los orbitales de tipo *d* semillenos en el elemento del sitio octaédrico, los cuales ayudarían a estabilizar el marco de octaedros de vértice compartido. Debido a la fórmula de los compuestos de la Tabla 6.5.3 se sugiere que los elementos transicionales de la misma tabla poseen un estado de oxidación 1+, correspondiente a una configuración de capa llena, d^{10}. Lo mismo sucede con los elementos galio, indio y talio.

En cuanto a las brechas energéticas, cabe mencionar que se desconoce si los compuestos validados son de brecha directa o indirecta debido a que no se tuvo información sobre la estructura de bandas de los mismos. Como es bien sabido, los compuestos semiconductores de brecha directa son los que se necesitan en aplicaciones optoelectrónicas. Adicionalmente, se ha reportado que los valores de brecha entre bandas son en general subestimados por los cálculos químico cuánticos [89-92]. A pesar de esto, los cálculos son útiles en comparar diferentes compuestos en términos cualitativos. Respecto a esto, los compuestos de la Tabla 6.5.1 abarcan un margen de valores de brecha energética amplio (hasta de 7.88 eV con $KBeF_3$) comparado con los compuestos de la Tabla 6.5.2 (hasta 0.97 eV con $AgTiBr_3$) y 6.5.3 (3.75 eV con $TlZnF_3$). En adición, los compuestos de la Tabla 6.5.1 resultaron ser más atractivos en términos de tendencias a lo largo de una familia o serie de transición, siempre que hubiera una serie de compuestos presenta el mismo grupo espacial.

Respecto a los compuestos de la Tabla 6.5.1, los fluoruros con el aristotipo de perovskita dieron lugar a brechas entre bandas más amplias que sus contrapartes cloruros, bromuros y yoduros; siendo con este último tipo de compuestos los que presentaron las brechas energéticas más pequeñas, aunque con posibilidad de ser utilizados como materiales optoelectrónicos. En lo general, se observó que la brecha energética disminuyó en los compuestos de la Tabla 6.5.1, con la estructura del aristotipo, cuando los elementos más pesados de una familia de la tabla periódica ocuparon el sitio octaédrico (metales alcalinos o alcalinotérreos). La excepción a esta tendencia fueron los compuestos $LiMX_3$, con *X* = Cl, Br o I. Lo que se puede comentar adicionalmente sobre esta excepción es que en este tipo de compuestos el litio ocupa el sitio cuboctaédrico y que, considerando a los compuestos como completamente iónicos, este elemento y el berilio presentan los radios iónicos más pequeños de todos los elementos considerados. En los compuestos con estructura aristotípica de la Tabla 6.5.1 $KMCl_3$, $KMBr_3$, KMI_3, $GaMF_3$, $GaMCl_3$, $RbMBr_3$, $AgMF_3$, $AgMCl_3$, $AgMBr_3$, $CsMF_3$, $CsMCl_3$, $CsMBr_3$, y $CsMI_3$ se observó una tendencia inicialmente de incremento en la brecha entre bandas hasta alcanzar un máximo con el calcio. Sobre este hecho, se puede sugerir que obedece a un cambio en la relación de los radios entre los elementos de los sitios cuboctaédrico y octaédrico. No obstante, si esto así fuera, no debería de observarse la misma tendencia con los compuestos $CsMX_3$ pues el cesio es, de los elementos considerados, el de mayor radio iónico.

Con los compuestos de la Tabla 6.5.2 no fue posible identificar una tendencia a lo largo de una serie de transición, en parte porque los compuestos no presentaron el mismo grupo espacial para establecer una comparación; sin embargo, las brechas energéticas obtenidas sugieren que estos compuestos pueden ser de poco interés al menos para aplicaciones optoelectrónicas. Con los compuestos de la Tabla 6.5.3 se observó que la brecha entre bandas era menor con elementos más pesados de las familias 11, 12 y la familia del galio, con una tendencia de pasar de carácter de semiconductor al de conductor. En la familia del zinc se observó, en algunos casos, que la brecha energética del compuesto resultó ser mayor con el

galio que con el zinc, probablemente relacionado con las contracciones escándida y lantánida.

Figura 6.5.2: Fluoruros con estructura perovskita validados con cálculos de optimización de la geometría con condiciones periódicas en la frontera. En el eje *x* se señala al número atómico del elemento localizado en el sitio cuboctaédrico. De manera análoga, en el eje *y* se indica al elemento localizado en el sitio octaédrico. Las líneas quebradas verticales y horizontales señalan a cada periodo de la tabla periódica.

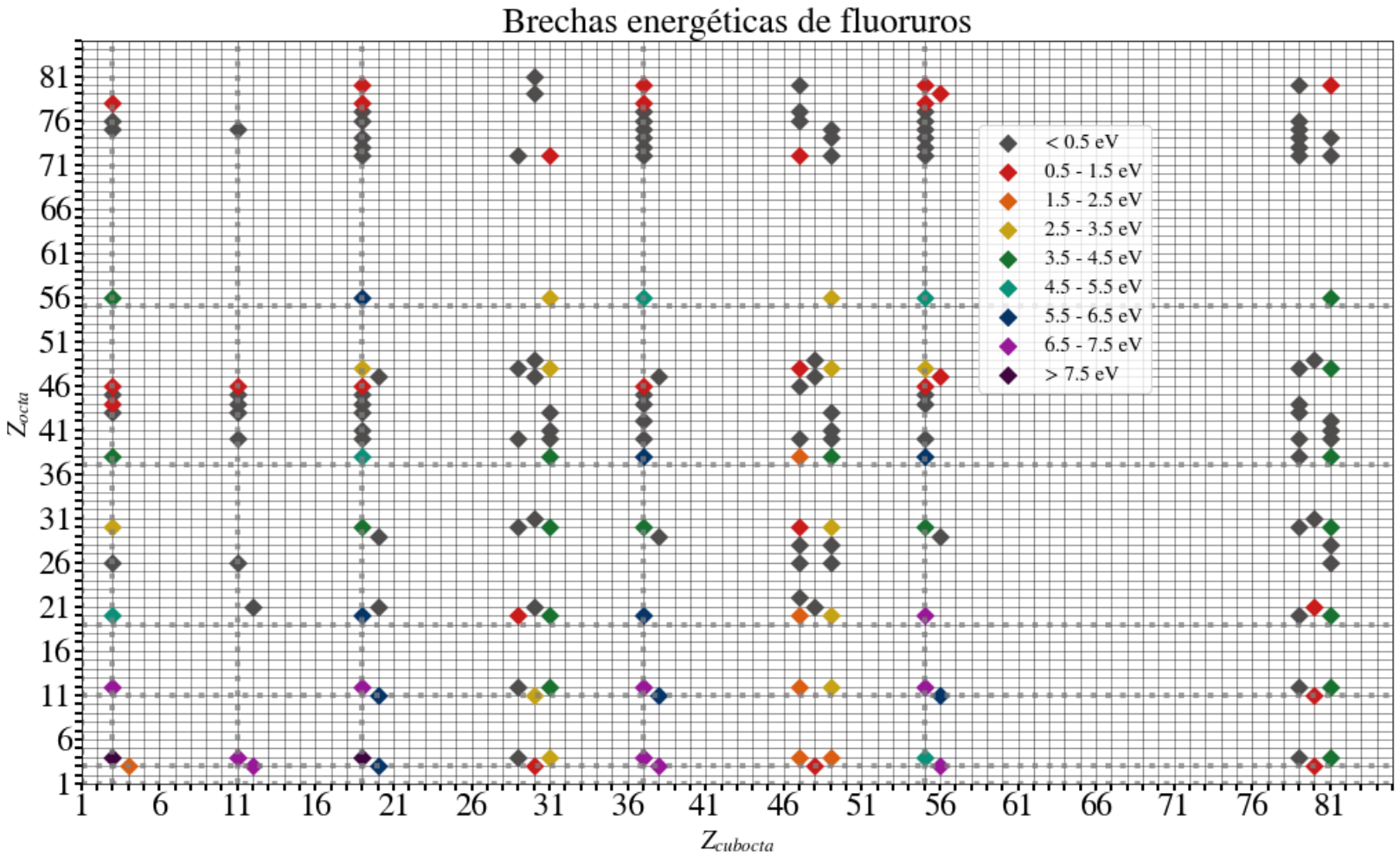


Figura 6.5.3: Cloruros con estructura perovskita validados con cálculos de ptimización de la geometría con condiciones periódicas en la frontera. En el eje *x* se señala al número atómico del elemento localizado en el sitio cuboctaédrico. De manera análoga, en el eje *y* se indica al elemento localizado en el sitio octaédrico. Las líneas quebradas verticales y horizontales señalan a cada periodo de la tabla periódica.

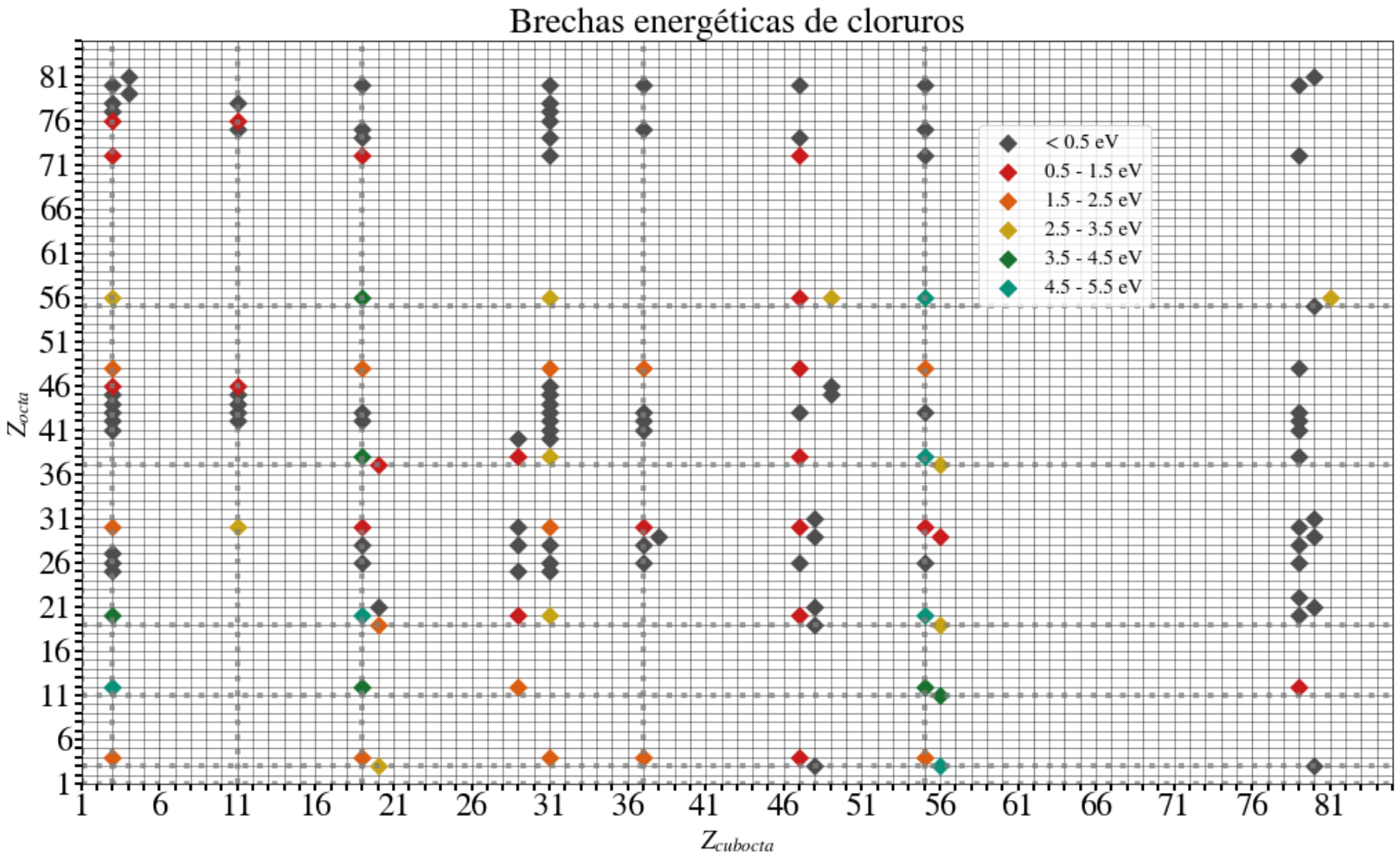


Figura 6.5.4: Bromuros con estructura perovskita validados con cálculos de optimización de la geometría con condiciones periódicas en la frontera. En el eje *x* se señala al número atómico del elemento localizado en el sitio cuboctaédrico. De manera análoga, en el eje *y* se indica al elemento localizado en el sitio octaédrico. Las líneas quebradas verticales y horizontales señalan a cada periodo de la tabla periódica.

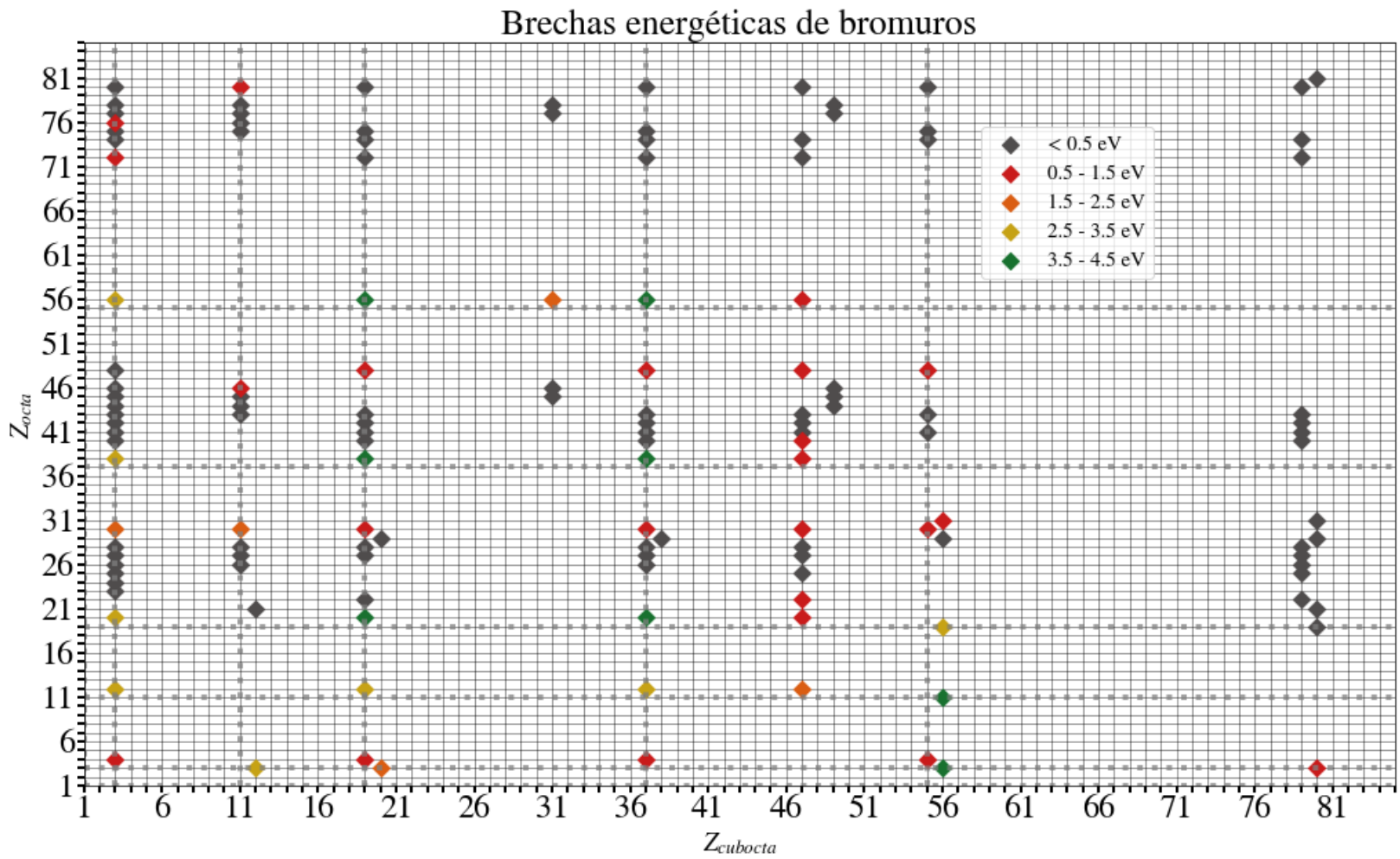


Figura 6.5.5: Yoduros con estructura perovskita validados con cálculos de optimización de la geometría con condiciones periódicas en la frontera. En el eje *x* se señala al número atómico del elemento localizado en el sitio cuboctaédrico. De manera análoga, en el eje *y* se indica al elemento localizado en el sitio octaédrico. Las líneas quebradas verticales y horizontales señalan a cada periodo de la tabla periódica.

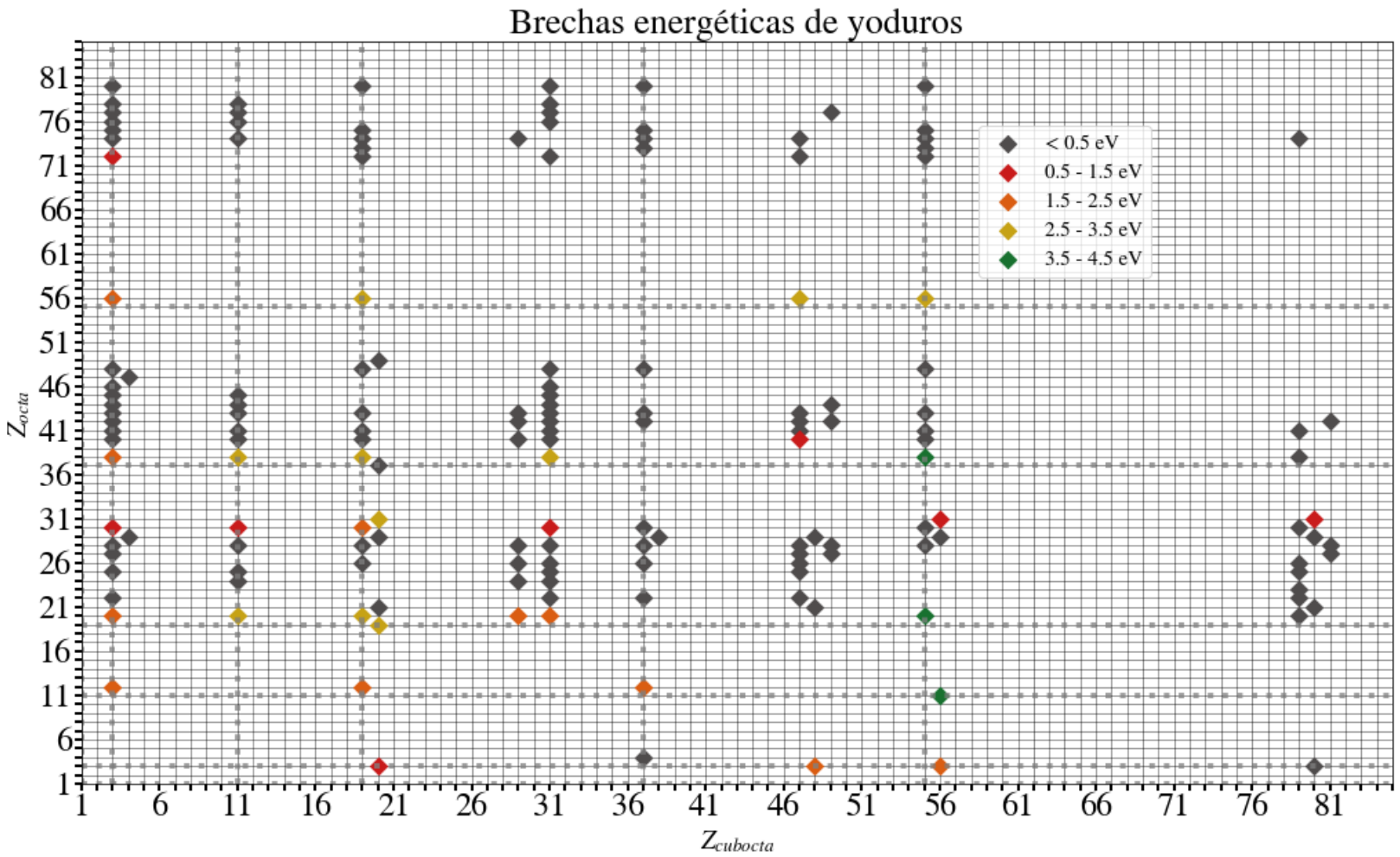


Tabla 6.5.1: Compuestos con estructura perovskita y validados por cálculos químico cuánticos. En la fórmula ABX_3, el elemento *B* (de tipo *s*) es el que aloja en el centro de los sitios octaédricos. Dentro de la Tabla, los compuestos se ordenan en términos del elemento *A*, que se aloja en los huecos que deja el marco de octaedros de vértice compartido, y el halogenuro.

Compuesto	Brecha energética (eV)	Grupo espacial
$LiBeF_3$	7.50	221
$LiMgF_3$	6.85	1
$LiCaF_3$	4.93	221
$LiSrF_3$	4.35	221
$LiBaF_3$	3.50	221
$LiBeCl_3$	2.05	221
$LiMgCl_3$	4.69	1
$LiCaCl_3$	3.70	221
$LiBaCl_3$	3.43	221
$LiBeBr_3$	0.54	221
$LiMgBr_3$	3.22	1
$LiCaBr_3$	2.98	221
$LiSrBr_3$	2.97	221
$LiBaBr_3$	2.93	221
$LiMgI_3$	2.07	1
$LiCaI_3$	2.29	221
$LiSrI_3$	2.33	221
$LiBaI_3$	2.36	221
$BeLiF_3$	1.57	221
$NaBeF_3$	7.43	221
$NaCaI_3$	3.41	1
$NaSrI_3$	3.37	1
$MgLiF_3$	6.82	1
$MgLiBr_3$	2.95	1
$KBeF_3$	7.88	221
$KMgF_3$	6.86	221
$KCaF_3$	6.22	221
$KSrF_3$	5.44	221
$KBaF_3$	5.60	1
$KBeCl_3$	2.05	221
$KMgCl_3$	4.31	1
$KCaCl_3$	4.76	221
$KSrCl_3$	4.44	221
$KBaCl_3$	4.17	221
$KBeBr_3$	0.56	221
$KMgBr_3$	3.21	1
$KCaBr_3$	3.87	221
$KSrBr_3$	3.73	221
$KBaBr_3$	3.59	221
$KMgI_3$	2.25	1
$KCaI_3$	3.14	221
$KSrI_3$	3.10	221
$KBaI_3$	3.05	221
$CaLiF_3$	6.50	221
$CaNaF_3$	5.88	1
$CaLiCl_3$	3.26	221
$CaKCl_3$	1.85	221
$CaRbCl_3$	1.27	221
$CaLiBr_3$	2.01	221
$CaLiI_3$	0.90	221
$CaKI_3$	2.91	166
$CaRbI_3$	0.47	221
$CuBeF_3$	-0.04	221
$CuMgF_3$	0.35	1
$CuCaF_3$	0.93	1
$CuMgCl_3$	1.64	1
$CuCaCl_3$	0.65	221
$CuSrCl_3$	0.76	221
$CuCaI_3$	1.90	1
$ZnLiF_3$	0.77	221
$ZnNaF_3$	3.38	221
$GaBeF_3$	3.33	221
$GaMgF_3$	3.63	6
$GaCaF_3$	3.95	221
$GaSrF_3$	3.87	221
$GaBaF_3$	2.95	221
$GaBeCl_3$	2.26	221
$GaCaCl_3$	3.37	221
$GaSrCl_3$	3.17	221
$GaBaCl_3$	2.84	221
$GaBaBr_3$	2.41	221
$GaCaI_3$	1.94	221
$GaSrI_3$	2.97	1
$RbBeF_3$	7.03	221
$RbMgF_3$	6.82	221
$RbCaF_3$	6.38	221
$RbSrF_3$	5.67	221
$RbBaF_3$	5.01	1
$RbBeCl_3$	2.00	221
$RbBeBr_3$	0.54	221
$RbMgBr_3$	2.61	2
$RbCaBr_3$	4.01	221
$RbSrBr_3$	3.85	221
$RbBaBr_3$	3.72	221
$RbBeI_3$	-0.09	221
$RbMgI_3$	1.98	1
$SrLiF_3$	7.40	221
$SrNaF_3$	6.05	9
$AgBeF_3$	1.80	221
$AgMgF_3$	1.76	221
$AgCaF_3$	2.11	221
$AgSrF_3$	1.95	221
$AgBeCl_3$	0.90	221
$AgCaCl_3$	1.37	221
$AgSrCl_3$	1.27	221
$AgBaCl_3$	1.07	221

Tabla 6.5.1 (cont.): Compuestos con estructura perovskita y validados por cálculos químico cuánticos. En la fórmula ABX_3, el elemento *B* (de tipo *s*) es el que aloja en el centro de los sitios octaédricos. Dentro de la Tabla, los compuestos se ordenan en términos del elemento *A*, que se aloja en los huecos que deja el marco de octaedros de vértice compartido, y el halogenuro.

Compuesto	Brecha energética (eV)	Grupo espacial
$AgMgBr_3$	1.94	1
$AgCaBr_3$	0.73	221
$AgSrBr_3$	0.73	221
$AgBaBr_3$	0.64	221
$AgBaI_3$	2.81	160
$CdLiF_3$	1.44	221
$CdLiCl_3$	-0.06	221
$CdKCl_3$	-0.03	221
$CdLiI_3$	1.62	2
$InBeF_3$	2.23	221
$InMgF_3$	2.72	1
$InCaF_3$	3.02	1
$InSrF_3$	3.58	221
$InBaF_3$	3.00	1
$InBaCl_3$	3.21	221
$CsBeF_3$	5.13	221
$CsMgF_3$	6.54	146
$CsCaF_3$	6.99	221
$CsSrF_3$	6.24	221
$CsBaF_3$	5.25	1
$CsBeCl_3$	1.88	221
$CsMgCl_3$	3.56	1
$CsCaCl_3$	5.33	221
$CsSrCl_3$	4.92	221
$CsBaCl_3$	4.64	221
$CsBeBr_3$	0.55	221
$CsCaI_3$	3.53	221
$CsSrI_3$	3.54	221
$CsBaI_3$	3.44	221
$BaLiF_3$	6.76	221
$BaNaF_3$	6.29	221
$BaLiCl_3$	4.72	221
$BaNaCl_3$	4.32	221
$BaKCl_3$	3.24	221
$BaRbCl_3$	2.57	221
$BaLiBr_3$	3.60	221
$BaNaBr_3$	4.27	1
$BaKBr_3$	2.52	221
$BaLiI_3$	2.33	221
$BaNaI_3$	3.65	221
$AuBeF_3$	-0.10	221
$AuMgF_3$	0.33	1
$AuCaF_3$	0.16	221
$AuSrF_3$	0.34	221
$AuMgCl_3$	1.13	1
$AuCaCl_3$	0.05	221
$AuSrCl_3$	0.05	221
$AuCaI_3$	-0.05	221
$AuSrI_3$	-0.05	221
$HgLiF_3$	0.94	1
$HgNaF_3$	1.37	1
$HgLiCl_3$	-0.06	221
$HgCsCl_3$	0.34	139
$HgLiBr_3$	1.49	1
$HgKBr_3$	0.45	14
$HgLiI_3$	-0.07	221
$TlBeF_3$	3.59	221
$TlMgF_3$	3.96	221
$TlCaF_3$	4.18	221
$TlSrF_3$	4.22	221
$TlBaF_3$	3.85	1
$TlBaCl_3$	3.38	221

Tabla 6.5.2: Compuestos con estructura perovskita y validados por cálculos químico cuánticos. En la fórmula ABX_3, el elemento *B* (del bloque *d*) es el que aloja en el centro de los sitios octaédricos. Dentro de la Tabla, los compuestos se ordenan en términos del elemento *A*, que se aloja en los huecos que deja el marco de octaedros de vértice compartido, y el halogenuro. Adicionalmente, para un mismo par *A*, *X*, las diferentes series de transición se señalan con un color sombreado gris o blanco.

Compuesto	Brecha energética (eV)	Grupo espacial
$LiFeF_3$	-0.09	221
$LiTcF_3$	0.04	221
$LiRuF_3$	0.72	1
$LiRhF_3$	-0.13	221
$LiPdF_3$	0.89	221
$LiReF_3$	0.11	2
$LiOsF_3$	0.06	221
$LiPtF_3$	0.54	221
$LiMnCl_3$	0.01	62
$LiFeCl_3$	0.08	13
$LiCoCl_3$	-0.08	59
$LiNbCl_3$	0.29	59
$LiMoCl_3$	0.48	11
$LiTcCl_3$	0.06	2
$LiRuCl_3$	-0.12	221
$LiRhCl_3$	-0.07	221
$LiPdCl_3$	0.77	221
$LiHfCl_3$	0.65	221
$LiOsCl_3$	0.73	13
$LiIrCl_3$	-0.06	62
$LiPtCl_3$	0.24	221
$LiVBr_3$	0.30	148
$LiCrBr_3$	-0.03	59
$LiMnBr_3$	0.01	1
$LiFeBr_3$	0.06	14
$LiCoBr_3$	-0.08	59
$LiNiBr_3$	0.34	204
$LiZrBr_3$	0.15	25
$LiNbBr_3$	0.09	11
$LiMoBr_3$	0.04	59
$LiTcBr_3$	0.00	1
$LiRuBr_3$	0.48	14
$LiRhBr_3$	-0.04	221
$LiPdBr_3$	0.15	221
$LiHfBr_3$	0.77	221
$LiWBr_3$	0.49	62
$LiReBr_3$	0.13	2
$LiOsBr_3$	0.70	14
$LiIrBr_3$	0.04	62
$LiPtBr_3$	0.38	62

Compuesto	Brecha energética (eV)	Grupo espacial
$LiTiI_3$	-0.09	221
$LiMnI_3$	-0.01	59
$LiCoI_3$	0.04	62
$LiNiI_3$	0.40	11
$LiZrI_3$	-0.02	221
$LiNbI_3$	0.22	62
$LiMoI_3$	0.02	148
$LiTcI_3$	-0.07	14
$LiRuI_3$	0.16	2
$LiRhI_3$	0.12	62
$LiPdI_3$	0.34	137
$LiHfI_3$	0.52	221
$LiWI_3$	0.03	1
$LiReI_3$	-0.02	11
$LiOsI_3$	0.49	14
$LiIrI_3$	0.13	62
$LiPtI_3$	0.17	137
NaFeF3	0.06	1
$NaZrF_3$	0.32	1
$NaTcF_3$	0.18	1
$NaRuF_3$	0.41	1
$NaRhF_3$	-0.08	1
$NaPdF_3$	0.53	1
$NaReF_3$	0.06	1
$NaMoCl_3$	0.32	1
$NaTcCl_3$	-0.01	1
$NaRuCl_3$	0.38	1
$NaRhCl_3$	-0.08	1
$NaPdCl_3$	0.60	1
$NaReCl_3$	0.10	1
$NaOsCl_3$	0.64	1
$NaPtCl_3$	0.47	1
$NaFeBr_3$	0.03	1
$NaCoBr_3$	0.05	1
$NaNiBr_3$	0.37	1
$NaTcBr_3$	0.00	1
$NaRuBr_3$	0.25	1
$NaRhBr_3$	0.01	1
$NaPdBr_3$	0.53	1
$NaReBr_3$	-0.01	1
$NaOsBr_3$	0.44	1
$NaIrBr_3$	0.18	1
$NaPtBr_3$	0.38	1

Tabla 6.5.2 (cont.): Compuestos con estructura perovskita y validados por cálculos químico cuánticos. En la fórmula ABX_3, el elemento *B* (del bloque *d*) es el que aloja en el centro de los sitios octaédricos. Dentro de la Tabla, los compuestos se ordenan en términos del elemento *A*, que se aloja en los huecos que deja el marco de octaedros de vértice compartido, y el halogenuro. Adicionalmente, para un mismo par *A*, *X*, las diferentes series de transición se señalan con un color sombreado gris o blanco.

Compuesto	Brecha energética (eV)	Grupo espacial
$NaCrI_3$	-0.03	1
$NaMnI_3$	-0.02	1
$NaNiI_3$	0.43	1
$NaZrI_3$	0.08	1
$NaNbI_3$	0.3	1
$NaTcI_3$	-0.03	1
$NaRuI_3$	0.16	1
$NaRhI_3$	-0.03	1
$NaWI_3$	0.03	1
$NaOsI_3$	0.27	1
$NaIrI_3$	0.30	1
$NaPtI_3$	0.37	1
$MgScF_3$	0.46	1
$MgScBr_3$	0.05	1
$KZrF_3$	-0.07	221
$KNbF_3$	-0.07	221
$KTcF_3$	-0.05	221
$KRuF_3$	-0.09	221
$KRhF_3$	-0.13	221
$KPdF_3$	0.93	221
$KHfF_3$	-0.04	221
$KTaF_3$	-0.08	221
KWF_3	-0.03	221
$KOsF_3$	-0.01	221
$KIrF_3$	-0.10	221
$KPtF_3$	0.62	221
$KFeCl_3$	-0.07	62
$KNiCl_3$	0.27	221
$KMoCl_3$	0.21	62
$KTcCl_3$	-0.21	62
$KHfCl_3$	0.56	67
$KWCl_3$	0.08	11
$KReCl_3$	0.36	14
$KTiBr_3$	-0.08	221
$KCoBr_3$	-0.07	62
$KNiBr_3$	0.34	204
$KZrBr_3$	0.15	59
$KNbBr_3$	0.26	62
$KMoBr_3$	0.12	62
$KTcBr_3$	0.03	62
$KHfBr_3$	-0.04	221
$KWBr_3$	0.12	62
$KReBr_3$	0.07	11

Compuesto	Brecha energética (eV)	Grupo espacial
$KFeI_3$	-0.06	62
$KNiI_3$	0.42	204
$KZrI_3$	-0.03	221
$KNbI_3$	0.27	62
$KTcI_3$	-0.05	62
$KHfI_3$	-0.04	221
$KTaI_3$	0.27	62
KWI_3	-0.01	59
$KReI_3$	0.03	11
$CaScF_3$	-0.05	221
$CaScCl_3$	0.13	221
$CaScI_3$	0.30	221
$CuZrF_3$	0.18	146
$CuHfF_3$	0.37	1
$CuMnCl_3$	-0.10	1
$CuNiCl_3$	0.07	6
$CuZrCl_3$	0.34	221
$CuCrI_3$	0.04	6
$CuFeI_3$	0.00	1
$CuNiI_3$	0.13	1
$CuZrI_3$	0.15	221
$CuMoI_3$	0.04	1
$CuTcI_3$	-0.11	1
$CuWI_3$	0.00	146
$ZnScF_3$	0.29	221
$GaZrF_3$	0.41	5
$GaNbF_3$	-0.06	221
$GaTcF_3$	0.10	2
$GaHfF_3$	0.66	6
$GaMnCl_3$	-0.01	6
$GaFeCl_3$	-0.06	1
$GaNiCl_3$	0.07	1
$GaZrCl_3$	-0.05	221
$GaNbCl_3$	0.22	11
$GaMoCl_3$	0.08	2
$GaTcCl_3$	0.07	1
$GaRuCl_3$	-0.01	1
$GaRhCl_3$	-0.09	221
$GaPdCl_3$	0.33	221
$GaHfCl_3$	-0.06	221
$GaWCl_3$	0.24	1
$GaOsCl_3$	0.30	59
$GaIrCl_3$	-0.10	221
$GaPtCl_3$	0.21	14

Tabla 6.5.2 (cont.): Compuestos con estructura perovskita y validados por cálculos químico cuánticos. En la fórmula ABX_3, el elemento *B* (del bloque *d*) es el que aloja en el centro de los sitios octaédricos. Dentro de la Tabla, los compuestos se ordenan en términos del elemento *A*, que se aloja en los huecos que deja el marco de octaedros de vértice compartido, y el halogenuro. Adicionalmente, para un mismo par *A*, *X*, las diferentes series de transición se señalan con un color sombreado gris o blanco.

Compuesto	Brecha energética (eV)	Grupo espacial
$GaRhBr_3$	-0.02	11
$GaPdBr_3$	0.12	221
$GaIrBr_3$	0.02	2
$GaPtBr_3$	0.21	221
$GaTiI_3$	-0.02	6
$GaCrI_3$	0.02	6
$GaMnI_3$	0.11	1
$GaFeI_3$	0.04	11
$GaNiI_3$	0.06	1
$GaZrI_3$	-0.04	221
$GaNbI_3$	0.10	6
$GaMoI_3$	-0.01	2
$GaTcI_3$	0.09	2
$GaRuI_3$	0.32	2
$GaRhI_3$	-0.02	6
$GaPdI_3$	0.18	204
$GaHfI_3$	-0.04	221
$GaOsI_3$	0.26	2
$GaIrI_3$	0.01	11
$GaPtI_3$	0.11	2
$RbZrF_3$	-0.06	221
$RbMoF_3$	-0.08	221
$RbRuF_3$	-0.09	221
$RbRhF_3$	-0.14	221
$RbPdF_3$	0.91	221
$RbHfF_3$	-0.05	221
$RbTaF_3$	-0.06	221
$RbWF_3$	-0.02	221
$RbReF_3$	0.23	2
$RbOsF_3$	-0.06	221
$RbIrF_3$	-0.10	221
$RbPtF_3$	0.63	221
$RbFeCl_3$	-0.02	62
$RbNiCl_3$	0.19	221
$RbNbCl_3$	0.10	15
$RbMoCl_3$	0.07	2
$RbTcCl_3$	0.15	62
$RbReCl_3$	0.32	14
$RbFeBr_3$	-0.06	11
$RbCoBr_3$	-0.04	63
$RbNiBr_3$	0.25	204
$RbZrBr_3$	0.29	135
$RbNbBr_3$	0.01	11
$RbMoBr_3$	0.07	62
$RbTcBr_3$	0.07	62
$RbHfBr_3$	-0.04	221
$RbWBr_3$	0.03	11
$RbReBr_3$	0.04	62
$RbTiI_3$	-0.02	221
$RbFeI_3$	-0.08	221
$RbNiI_3$	0.34	47
$RbMoI_3$	-0.06	11
$RbTcI_3$	0.02	221
$RbTaI_3$	-0.07	59
$RbWI_3$	0.11	62
$RbReI_3$	-0.01	11
$AgTiF_3$	-0.02	221
$AgFeF_3$	0.29	221
$AgNiF_3$	0.04	221
$AgZrF_3$	0.15	4
$AgPdF_3$	-0.05	221
$AgHfF_3$	0.79	1
$AgOsF_3$	0.00	229
$AgIrF_3$	-0.10	221
$AgFeCl_3$	0.08	62
$AgTcCl_3$	-0.01	14
$AgHfCl_3$	0.51	221
$AgWCl_3$	0.12	2
$AgTiBr_3$	0.97	221
$AgMnBr_3$	-0.11	7
$AgCoBr_3$	-0.06	1
$AgNiBr_3$	0.02	2
$AgZrBr_3$	0.70	221
$AgNbBr_3$	0.17	62
$AgMoBr_3$	-0.01	167
$AgTcBr_3$	0.08	14
$AgHfBr_3$	0.16	221
$AgWBr_3$	0.10	11
$AgTiI_3$	-0.02	47
$AgMnI_3$	-0.06	1
$AgFeI_3$	0.01	2
$AgCoI_3$	-0.06	148
$AgNiI_3$	0.10	11
$AgZrI_3$	0.58	221
$AgNbI_3$	0.23	62
$AgMoI_3$	0.03	148
$AgTcI_3$	-0.05	148
$AgHfI_3$	0.36	221
$AgWI_3$	0.14	221
$CdScF_3$	0.20	221
$CdScCl_3$	0.17	205
$CdScI_3$	0.24	7

Tabla 6.5.2 (cont.): Compuestos con estructura perovskita y validados por cálculos químico cuánticos. En la fórmula ABX_3, el elemento *B* (del bloque *d*) es el que aloja en el centro de los sitios octaédricos. Dentro de la Tabla, los compuestos se ordenan en términos del elemento *A*, que se aloja en los huecos que deja el marco de octaedros de vértice compartido, y el halogenuro. Adicionalmente, para un mismo par *A*, *X*, las diferentes series de transición se señalan con un color sombreado gris o blanco.

Compuesto	Brecha energética (eV)	Grupo espacial
$InFeF_3$	-0.07	221
$InNiF_3$	0.09	205
$InZrF_3$	-0.07	221
$InNbF_3$	-0.01	221
$InTcF_3$	-0.05	221
$InHfF_3$	0.45	2
$InWF_3$	-0.08	221
$InReF_3$	0.35	1
$InRhCl_3$	-0.10	221
$InPdCl_3$	0.11	60
$InRuBr_3$	0.24	62
$InRhBr_3$	-0.07	221
$InPdBr_3$	0.11	62
$InIrBr_3$	0.01	11
$InPtBr_3$	0.38	221
$InCoI_3$	-0.09	25
$InNiI_3$	-0.03	25
$InMoI_3$	0.04	11
$InRuI_3$	0.22	62
$InIrI_3$	0.02	11
$CsZrF_3$	-0.06	221
$CsRuF_3$	-0.08	221
$CsRhF_3$	-0.10	221
$CsPdF_3$	0.83	221
$CsHfF_3$	-0.05	221
$CsTaF_3$	-0.07	221
$CsWF_3$	-0.09	2
$CsReF_3$	-0.01	221
$CsOsF_3$	-0.07	221
$CsIrF_3$	-0.10	221
$CsPtF_3$	0.63	221
$CsFeCl_3$	0.01	74
$CsTcCl_3$	0.07	62
$CsHfCl_3$	0.26	12
$CsReCl_3$	0.19	2
$CsNbBr_3$	0.04	54
$CsTcBr_3$	0.08	62
$CsWBr_3$	0.10	14
$CsReBr_3$	0.06	2
$CsNiI_3$	0.19	221
$CsZrI_3$	0.02	221
$CsNbI_3$	0.15	62
$CsTcI_3$	0.05	221
$CsHfI_3$	-0.04	221
$CsTaI_3$	0.29	206
$CsWI_3$	0.02	11
$CsReI_3$	0.01	11

Compuesto	Brecha energética (eV)	Grupo espacial
$AuZrF_3$	0.40	225
$AuTcF_3$	0.05	4
$AuRuF_3$	0.22	4
$AuHfF_3$	0.03	221
$AuTaF_3$	-0.03	221
$AuWF_3$	0.25	221
$AuReF_3$	0.07	1
$AuOsF_3$	-0.05	221
$AuTiCl_3$	0.49	221
$AuFeCl_3$	0.14	14
$AuNiCl_3$	0.32	148
$AuNbCl_3$	-0.14	221
$AuMoCl_3$	0.13	2
$AuTcCl_3$	0.03	1
$AuHfCl_3$	0.08	221
$AuTiBr_3$	0.16	221
$AuMnBr_3$	-0.06	14
$AuFeBr_3$	0.16	14
$AuCoBr_3$	0.06	14
$AuNiBr_3$	-0.02	2
$AuZrBr_3$	0.03	221
$AuNbBr_3$	-0.13	221
$AuMoBr_3$	0.36	221
$AuTcBr_3$	0.14	14
$AuHfBr_3$	0.04	221
$AuWBr_3$	0.28	221
$AuTiI_3$	0.17	1
$AuVI_3$	0.17	167
$AuMnI_3$	0.01	1
$AuFeI_3$	0.11	58
$AuNbI_3$	-0.12	221
$AuWI_3$	-0.01	148
$HgScF_3$	0.95	221
$HgScCl_3$	0.26	221
$HgScBr_3$	0.34	221
$HgScI_3$	0.43	221
$TlFeF_3$	-0.07	221
$TlNiF_3$	0.25	221
$TlZrF_3$	-0.06	221
$TlNbF_3$	-0.01	221
$TlMoF_3$	-0.01	2
$TlHfF_3$	0.41	2
$TlWF_3$	-0.09	221
$TlCoI_3$	-0.03	62
$TlNiI_3$	0.18	62
$TlMoI_3$	0.07	62

Tabla 6.5.3: Compuestos con estructura perovskita y validados por cálculos químico cuánticos. En la fórmula ABX_3, el elemento *B* es el que aloja en el centro de los sitios octaédricos. Dentro de la Tabla, los compuestos se ordenan en términos del elemento *A*, que se aloja en los huecos que deja el marco de octaedros de vértice compartido, y el halogenuro.

Compuesto	Brecha energética (eV)	Grupo espacial
$LiZnF_3$	2.93	221
$LiZnCl_3$	1.53	221
$LiCdCl_3$	1.52	221
$LiHgCl_3$	-0.17	221
$LiZnBr_3$	2.01	146
$LiCdBr_3$	0.48	221
$LiHgBr_3$	-0.15	221
$LiZnI_3$	1.35	1
$LiCdI_3$	-0.05	221
$LiHgI_3$	-0.13	221
$BeAuCl_3$	-0.03	221
$BeTlCl_3$	-0.04	221
$BeCuI_3$	0.14	221
$BeAgI_3$	0.17	221
$NaZnCl_3$	2.73	1
$NaZnBr_3$	2.07	1
$NaHgBr_3$	0.68	1
$NaZnI_3$	1.36	1
$KZnF_3$	3.57	221
$KCdF_3$	2.83	221
$KHgF_3$	0.58	1
$KZnCl_3$	1.39	221
$KCdCl_3$	1.61	221
$KHgCl_3$	-0.19	221
$KZnBr_3$	1.20	167
$KCdBr_3$	0.65	221
$KHgBr_3$	-0.17	221
$KZnI_3$	1.88	1
$KCdI_3$	-0.06	221
$KHgI_3$	-0.13	221
$CaCuF_3$	-0.22	221
$CaAgF_3$	-0.18	221
$CaCuBr_3$	-0.16	221
$CaCuI_3$	-0.13	221
$CaGaI_3$	2.73	221
$CaInI_3$	-0.02	221
$CuZnF_3$	-0.02	221
$CuCdF_3$	-0.01	221
$CuZnCl_3$	0.25	221
$ZnGaF_3$	-0.04	9
$ZnAgF_3$	0.03	221
$ZnInF_3$	0.00	221
$ZnAuF_3$	0.36	221
$ZnTlF_3$	0.14	1
$GaZnF_3$	3.58	221
$GaCdF_3$	3.24	221
$GaZnCl_3$	1.58	221
$GaCdCl_3$	1.70	221
$GaHgCl_3$	-0.18	221
$GaZnI_3$	1.11	1
$GaCdI_3$	-0.05	221
$GaHgI_3$	-0.14	221

Compuesto	Brecha energética (eV)	Grupo espacial
$RbZnF_3$	3.61	221
$RbHgF_3$	0.59	2
$RbZnCl_3$	1.31	221
$RbCdCl_3$	1.58	221
$RbHgCl_3$	-0.17	221
$RbZnBr_3$	1.04	2
$RbCdBr_3$	0.62	221
$RbHgBr_3$	-0.17	221
$RbZnI_3$	-0.05	221
$RbCdI_3$	-0.06	221
$RbHgI_3$	-0.15	221
$SrCuF_3$	-0.23	221
$SrAgF_3$	-0.18	221
$SrCuCl_3$	0.00	221
$SrCuBr_3$	-0.05	140
$SrCuI_3$	-0.14	221
$AgZnF_3$	1.14	221
$AgCdF_3$	0.59	221
$AgHgF_3$	-0.20	221
$AgZnCl_3$	1.10	221
$AgCdCl_3$	1.14	221
$AgHgCl_3$	-0.16	221
$AgZnBr_3$	0.84	204
$AgCdBr_3$	0.72	221
$AgHgBr_3$	-0.11	221
$CdAgF_3$	-0.02	221
$CdInF_3$	0.15	205
$CdCuCl_3$	-0.02	221
$CdGaCl_3$	-0.05	1
$CdCuI_3$	0.12	1
$InZnF_3$	2.61	221
$InCdF_3$	3.06	221
$CsZnF_3$	3.52	221
$CsCdF_3$	3.04	221
$CsHgF_3$	0.71	221
$CsZnCl_3$	1.19	221
$CsCdCl_3$	1.53	221
$CsHgCl_3$	-0.17	221
$CsZnBr_3$	1.49	1
$CsCdBr_3$	0.59	221
$CsHgBr_3$	-0.16	221
$CsZnI_3$	-0.09	221
$CsCdI_3$	-0.07	221
$CsHgI_3$	-0.13	221
$BaCuF_3$	0.03	139
$BaAgF_3$	1.41	1
$BaAuF_3$	1.18	1
$BaCuCl_3$	0.66	221
$BaCuBr_3$	0.49	136
$BaGaBr_3$	0.75	221

Tabla 6.5.3 (cont.): Compuestos con estructura perovskita y validados por cálculos químico cuánticos. En la fórmula ABX_3, el elemento *B* es el que aloja en el centro de los sitios octaédricos. Dentro de la Tabla, los compuestos se ordenan en términos del elemento *A*, que se aloja en los huecos que deja el marco de octaedros de vértice compartido, y el halogenuro.

Compuesto	Brecha energética (eV)	Grupo espacial
$BaCuI_3$	-0.14	221
$BaGaI_3$	0.62	221
$AuZnF_3$	-0.08	221
$AuCdF_3$	-0.17	221
$AuHgF_3$	-0.19	221
$AuZnCl_3$	0.45	204
$AuCdCl_3$	-0.05	221
$AuHgCl_3$	-0.05	221
$AuHgBr_3$	-0.12	221
$AuZnI_3$	0.48	7
$HgGaF_3$	-0.04	221
$HgInF_3$	0.40	221
$HgCuCl_3$	-0.02	221
$HgGaCl_3$	-0.02	221
$HgTlCl_3$	-0.02	221
$HgCuBr_3$	0.06	221
$HgGaBr_3$	-0.01	221
$HgTlBr_3$	-0.04	221
$HgCuI_3$	0.09	221
$HgGaI_3$	0.59	1
$TlZnF_3$	3.75	221
$TlCdF_3$	3.67	221
$TlHgF_3$	1.21	221

6.6 COMENTARIOS FINALES, ALCANCES Y LIMITACIONES.

Las redes neuronales desarrolladas de seis y ocho sitios fueron más restrictivas en la evaluación de los 2784 compuestos propuestos en la sección anterior. Como se mencionó en la sección anterior, la red neuronal de cuatro sitios evaluó como candidatos a cristalizar con estructura tipo perovskita a 2771 compuestos; mientras que las redes neuronales de seis sitios y de ocho sitios encontraron 1621 y 647 compuestos, respectivamente. Otra diferencia entre las predicciones de las tres redes neuronales desarrolladas se encuentra en el número de compuestos que convergieron tras los cálculos de energía de un solo punto: sólo 777 de los 1621 compuestos predichos por la red neuronal de seis sitios convergieron (47.93 %); mientras que esto ocurrió con 294 de los 647 compuestos por la red neuronal de ocho sitios (45.44 %). Estos porcentajes de compuestos validados por cálculos de energía de un solo punto son similares al obtenido con la red neuronal de cuatro sitios, con la que se encontró que 49.55 % de los compuestos considerados como predicciones convergieron en cálculos de energía de un solo punto.

Una de las inquietudes que algún científico que se dedica a la síntesis de compuestos pudiera tener sobre los compuestos encontrados como viables es sobre de las condiciones de síntesis. Estas condiciones son también obviadas en cálculos químico cuánticos, los cuales son implementados en el vacío y en el cero absoluto. Incluso las redes neuronales desarrolladas consideran que los radios atómicos, necesarios para la construcción de los datos de entrada, son independientes de la temperatura y la presión. Esta información sobre las condiciones termodinámicas, ya sea durante la síntesis del compuesto o durante la medición, son raramente reportadas en los archivos CIF. Por ejemplo, 53 de los compuestos de tipo perovskita utilizados en la colección de cuatro sitios, reportaron en sus archivos CIF el intervalo de presión durante la síntesis (1 – 15 GPa) o al menos hicieron mención a que la síntesis ocurrió a presión alta o moderada.

Finalmente, se debe mencionar que los compuestos con estructura tipo perovskita se han considerado normalmente como de tipo iónico [2,15-16]. La diferencia de electronegatividad que se tomó en cuenta en las funciones de localidad pretende, en efecto, modelar el carácter iónico de estos compuestos. Adicionalmente, se hubiera esperado la utilización de radios iónicos [93] en lugar de los radios atómicos en la construcción de los datos de entrada de los compuestos. Una de las razones por la que no se usaron los radios iónicos se debe a que no se han reportado algunos valores para ciertos estados de oxidación de algunos elementos [93]. Además del estado de oxidación del elemento, los radios iónicos dependen tanto del número de coordinación como de la geometría definida por los primeros vecinos. Estas geometrías son fácilmente determinables para aquellos sitios muy simétricos, como los ya mencionados octaédricos y cuboctaédricos. Existen métodos que permiten estimar el estado de oxidación como el de valencia del enlace [6,94] (*Bond-Valence Method*, en inglés), que parametrizan la distancia entre el átomo central y los primeros vecinos. No obstante, no todos los parámetros para los posibles pares han sido reportados [95]. Otra alternativa para estimar el estado de oxidación y, consecuentemente el radio iónico, es mediante cálculos químico cuánticos [96-101], lo cual harían más costoso, computacionalmente, la construcción de los datos de entrada de un compuesto.

A pesar de este hecho, las precisiones en la clasificación de los compuestos por las redes neuronales fueron altas. Este resultado puede tener su origen el uso de las funciones de localidad en los datos de entrada, cuya influencia en la precisión de la clasificación de compuestos se mostró en la sección 6.4. Las funciones de localidad, además, podrían tener una relación con el estado de oxidación de un elemento, pues dependen de la distancia entre el par de átomos involucrados. No obstante, el desarrollo de un modelo que relacione las distancias entre un átomo central y sus vecinos con los radios iónicos puede contribuir al mejoramiento del mismo.

7. CONCLUSIONES

Más allá de los compuestos encontrados por las redes neuronales como candidatos de poseer una estructura de tipo perovskita, que fueron validados por cálculos químico cuánticos y que se enlistan en las Tablas 6.5.1 – 6.5.3, esto no hubiera sido posible sin una metodología como la desarrollada en este trabajo. Esta metodología se basó en la información contenida en los archivos CIF. Con dicha información es posible conocer el orden local alrededor de los átomos, lo que permite construir los datos de entrada de las redes neuronales artificiales. Estas redes neuronales clasifican eficientemente a los compuestos cristalinos como de tipo perovskita o no perovskita. Los datos de entrada construidos se basan en los diferentes entornos químicos de un cristal, definidos por los sitios de Wyckoff. La información contenida en los datos de entrada de la red neuronal es, esencialmente, de carácter estructural, por lo que es importante validar las predicciones hechas por las redes neuronales desde un enfoque electrónico. No obstante, las redes neuronales desarrolladas constituyen una herramienta para sondear el espacio de nuevos compuestos cristalinos de manera que el costo computacional sea reducido.

El autor de esta tesis espera que los usuarios de estas redes neuronales no encuentren una caja negra en la construcción de los datos de entrada de un compuesto, ya que éstos reflejan tanto factores estructurales y de empaquetamiento, así como ambientes químicos. Estos datos de entrada podrían ser calculados a lápiz y papel de manera directa, si así fuera el caso. El alcance de las redes neuronales, al ser alimentadas con este tipo de información, ha quedado reportado en las Tablas 6.3.1 – 6.3.9.

8. PERSPECTIVAS

La caracterización de los compuestos cristalinos implementada en este trabajo no está restringida únicamente a las estructuras de tipo perovskita y puede aplicarse a otros tipos de estructuras cristalinas. Esto se debe a la naturaleza de los datos de entrada. Adicionalmente, esta caracterización de compuestos puede aprovecharse para desarrollar redes neuronales artificiales con más de una salida.

El presente trabajo se restringió a encontrar nuevos compuestos con estructura tipo perovskita sin importar sus propiedades. El desarrollo de modelos para evaluar las propiedades de un nuevo material debe aprovechar la información de los sitios de simetría que existen en los materiales cristalinos.

Finalmente, se sugiere que esta metodología debe incorporarse en la currícula de químicos, físicos o científicos de materiales, interesados en la búsqueda de nuevos compuestos cristalinos.

9. REFERENCIAS

1. W. Borchardt-Ott *Kristallographie. Eine Einführung für Naturwissenschaftler*. 7te Auflage, Springer, **2009**. ISBN 978-3-540-78270-4
2. U. Müller, *Inorganic Structural Chemistry* 2nd Edition John Wiley & Sons **2006** ISBN-10: 0-470-01864-X
3. D. E. Sanders *Introducción a la Cristalografía* Ed. Reverté S. A. **1984** ISBN: 84-291-4150-2
4. C. Giacovazzo, H. L. Monaco, D. Viterbo, F. Scordari, G. Gilli, G. Zanotti, M. Catti *Fundamentals of Crystallography* Oxford University Press **2000** ISBN: 0 19 855578 4
5. *International Tables for Crystallography* Vol. A: *Space-Group Symmetry* **2005** Springer
6. I. D. Brown *The Chemical Bond in Inorganic Chemistry* Oxford University Press **2002** ISBN: 9780198508700
7. A. R. Chakmouradian, P. M. Woodward *Phys. Chem. Minerals* **2014**, 41, 387-391
8. K. Uchino *Sci. Tech. Adv. Mater.* **2015**, 16:4, 046001
9. C. A. Hancock, J.M. Porras-Vazquez, P.J. Keenan, P.R. Slater *Dalton Trans*. **2015**, 44, 10559-10569
10. Berdnorz, J.G. “Perovskites and their new role in oxide electronics. High T c superconductors and related transition metal oxides”. **2007** Springer. ISBN: 978-3-540-71023-3
11. M. A. Green, A. H. Baillie, H. J. Snaith *Nat. Photonics* **2014**, 8, 506 – 514
12. A. Kojima, K. Teshima, Y. Shirai, T. Miyasaka *J. Am. Chem. Soc.* **2009**, 131, 6050 – 6051
13. F. Sahli, J. Werner, B. A. Kamino, M. Bräuninger, R. Monnard, B. Paviet-Salomon, L. Barraud, L. Ding, J. J. Diaz Leon, D. Sacchetto, G. Cattaneo, M. Despeisse, M. Boccard, S. Nicolay, Q. Jeangros, B. Niesen, C. Ballif *Nature Mater.* **2018**, 17, 820-826.
14. P. M. Woodward, *Acta Cryst. B* **1997**, 53, 32-43
15. L. Pauling *J. Am. Chem. Soc.* **1929**, 51,1010 – 1026
16. V. M. Golschmidt, *Naturwissenchaften* **1926**, 21, 477-485

17. A. M. Glazer, *Acta Cryst. B* **1972**, 28, 3384-3385
18. C. J. Howard, H. T. Stokes *Acta Cryst. B* **1998**, 54, 782 – 789
19. M. W. Lufaso, P. M. Woodward *Acta Cryst. B*, **2001**, 54, 725-738
20. J. Schmidhuber *Front. Electr. Electron. Eng. Chine* **2010,** 5, 347-362
21. M. Furmankiewicz, A. Soltysik-Piorunkiewicz, P.Ziuzianski, *Last Trends on Systems* **2014** Vol. 3, 551 – 556 ISBN: 978-1-61804-244-6
22. T. Stonier *Beyond Information* **1992** Springer, London. ISBN 978-1-4471-1835-0
23. A. M. Turing, *Mind*, **1950**, 59, 433-460
24. a) https://ai.googleblog.com/2018/05/duplex-ai-system-for-natural-conversation.html (Consultado el 30 de mayo de 2020), b) https://www.cnet.com/features/google-assistant-duplex-at-io-could-become-the-most-lifelike-ai-voice-assistant-yet/ (Consultado el 30 de mayo de 2020), c) https://youtu.be/D5VN56jQMWM (Consultado el 30 de mayo de 2020).
25. J. Kleesiek, J. M. Murray, C. Strack, G. Kaissis *Radiologe* **2020**, 60, 24-31
26. E. Rich *Artificial Intelligence* McGraw-Hill **1983** ISBN 10:0-070-52261-8
27. http://www.nomodes.com/Larry_Tesler_Consulting/Adages_and_Coinages.html (Consultado el 11 de mayo de 2020).
28. M. I. Jordan, T. M. Mitchell *Science* **2015,** 349, 255 – 261.
29. A. L. Samuel *IBM J. Res. Develop.* **1959**, 3, 210-229
30. Tom M. Mitchell *Machine Learning* McGraw-Hill **1997** *ISBN-10: 0070428077*
31. *a)* R. A. Fischer *Annals of Eugenics* **1936**, 7, 179 – 188; b) https://archive.ics.uci.edu/ml/datasets/iris (Consultado el 18 de mayo de 2020)
32. C. M. Bishop *Pattern Recognition and Machine Learning* **2006** Springer ISBN-10: 0-387-31073-8
33. W. S. McCulloch, W. Pitts *Bull. Math. Biophys.* **1943**, 5, 115 – 133
34. F. Rosenblatt *The Perceptron. A perceiving and recognizing Automaton*. Cornell Aeronautical Laboratory, Inc. 1957
35. Y. LeCun, Y. Bengio, G. Hinton *Nature* **2015**, 521, 436 – 444
36. a) M. A. Nielsen *Neural Networks and Deep Learning*. Determination Press, 2015 (Disponible total y legalmente en http://neuralnetworksanddeeplearning.com/, Consultado el 23 de mayo de 2020); b) 2. S. Raschka, V. Mirjalili *Python Machine Learning* PacktPub 2nd Edition, **2017** ISBN-10: 9781787125933
37. Y. LeCun, L. Bottou, Y. Bengio, P. Haffner *Gradient-Based Learning Applied to Document Recognition* Proc. of the IEEE **1998**
38. S. Ruder, arXiV: 1609.04747
39. D. E. Rumelhart, G. E. Hinton, R. J. Williams *Nature* **1986**, 323, 533 – 536
40. G. Montavon, G. B. Orr, K.R. Müller *Neural Networks: Tricks of the Trade* 2nd Edition Springer-Verlag **2012** ISBN: 978-3-642-35288-1
41. S. Lorenz, A. Gross, M. Scheffler, *Chem. Phys. Lett.* **2004**, 395, 210 – 215
42. J. Behler, M. Parrinello, *Phys. Rev. Lett.* **2007**, 98, 146401
43. N. Artrith, A. Urban, *Comp. Mater. Science*, **2016**, 114, 135 – 150
44. K. Hansen, G. Montavon, F. Biegler, S. Fazli, M. Rupp, M. Scheffler, O. A. von Lilienfeld, A. Tkatchenko, K. R. Müller, *J. Chem. Theory Comput.* **2013**, 9, 3404 – 3419
45. M. Rupp, A. Tkatchenko, K. R. Müller, O. A. von Lilienfeld, *Phys. Rev. Lett.*, **2012**, *108*, 058301
46. F. Faber, A. Lindmaa, O. A. von Lilienfeld, R. Armiento, *Int. J. Quantum Chem.* **2015**, 115, 1094 – 1101
47. B. Schölkopf, Z. Luo, V. Vovk, *Empirical Inference*. Springer, **2013**. ISBN: 978-3-642-41135-9
48. A. Jain, S. P. Ong, G. Hautier, W. Chen, W. D. Richards, S. Dacek, S. Cholia, D. Gunter, D. Skinner, G. Ceder, K. A. Persson, *APL Mater.* **2013**, 1, 011002
49. J. Schmidt, H. Shi, P. Borlido, L. Chen, S. Botti, M. A. L. Marques *Chem. Mater.* **2017**, 29, 5090 – 5103
50. a) https://mathworld.wolfram.com/ConvexHull.html (Consultado el 23 de noviembre de 2020); b) https://en.wikipedia.org/wiki/Convex_hull (Consultado el 23 de noviembre de 2020).
51. A. V. Fedorov, I. V. Shamanaev *Mol. Inf.* **2017**, 36, 1600162
52. G. Thimm, *Acta Cryst. A,* **2009**, 65, 213 – 226
53. W. Ye, C. Chen, Z. Wang, I. H. Chu, S. P. Ong, *Nature Commun.* **2018**, 9, 3800

54. S. G. Javed, A. Khan, A. Majid, A. M. Mirza, J. Bashir *Comput. Mater. Science,* **2007**, 39, 627 – 634
55. A. Majid, A. Khan, T. S. Choi, *Comput. Mater. Science* **2011**, 50, 1879 – 1888
56. A Majid, A. Khan, G. Javed, A. M. Mirza *Comput. Mater. Science* **2010**, 50, 363 – 372
57. G. Pilania, P. V. Balachandran, C. Kim, T. Lookman, *Front. Mater.* **2016**, 3, 19
58. G. Pilania, P. V. Balachandran, J. E. Gubernatis, T. Lookman, *Acta Cryst. B,* **2015**, 71, 507 – 513
59. C. Cortes, V. Vapnik, *Machine Learning*, **1995**, 20, 273 – 297
60. W. Noble, *Nat. Biotechnology,* **2006**, 24, 1565 – 1567
61. A. Natekin, A. Knoll, *Front. Neurobot.*, **2013**, 7, 21
62. C. Li, X. Lu, W. Ding, L. Feng, Y. Gao, Z. Guo, *Acta Cryst. B*, **2008**, 64, 702 – 707
63. O. Isayev, C. Oses, C. Toher, E. Gossett, S. Curtarolo, A. Tropsha, *Nature Comm.* **2017**, *8*, 15679
64. S. Curtarolo, W. Setyawan, G. L. W. Hart, M. Jahnatek, R. V. Chepulskii, R. H. Taylor, S. Wang, J. Xue, K. Yang, O. Levy, M. J. Mehl, H. T. Stokes, D. O. Demchenko, D. Morgan, *Comput. Mater. Science*, **2012**, *58*, 218 – 226
65. Z. W. Salsburg, *J. Chem. Educ.*, **1966**, 43, 353 – 357
66. B. Cordero, V. Gómez, A. E. Platero-Prats, M. Revés, J. Echeverría, E. Cremades, F. Barragán, S. Alvarez, *Dalton Trans.*, **2008**, 2832 - 2838
67. T. Xie, J. C. Grossman, *Phys. Rev. Lett.* **2018**, 120, 145301
68. Materials Genome Initiative, Strategic Plan, National Science Technology Council,USA, 2014. https://obamawhitehouse.archives.gov/mgi (Consultado el 30 de junio de 2020)
69. Report of the Clean Energy Materials Innovation Challenge Expert Workshop,Mission Innovation, 2018 http://mission-innovation.net/wp-content/uploads/2018/01/Mission-Innovation-IC6-Report-Materials-Acceleration-Platform-Jan-2018.pdf (Consultado el 30 de junio de 2020)
70. J. Hachmann, R. Olivares-Amaya, S. Atahan-Evrenk, C. Amador-Bedolla, R. S. Sánchez-Carrera, A. Gold-Parker, L. Voigt, A. M. Brockway, A. Aspuru-Guzik *J. Phys. Chem. Lett.* **2011**, 2, 2241 - 2251
71. „Vom Material zur Innovation. Rahmenprogramm zur Förderung der Materialforschung“ Bundesministerium für Bildung und Forschung, 2015. https://www.bmbf.de/upload_filestore/pub/Vom_Material_zur_Innovation.pdf (Consultado el 30 de junio de 2020)
72. a) Gražulis, S., Chateigner, D., Downs, R. T., Yokochi, A. T., Quiros, M., Lutterotti, L., Manakova, E., Butkus, J., Moeck, P. & Le Bail, A., *J. Appl. Cryst*., **2009**, 42, 726-7 9 b) Gražulis, S., Daškevič, A., Merkys, A., Chateigner, D., Lutterotti, L., Quirós, M., Serebryanaya, N. R., Moeck, P., Downs, R. T. & LeBail, *A., Nucleic Acids Research*, **2012**, 40, D420-D427
73. S. P. Ong, W. D. Richards, A. Jain, G. Hautier, M. Kocher, S. Cholia, D. Gunter, V. L. Chevrier, K. A. Persson, G. Ceder, *Comp. Mat. Science*, **2013**, 68, 314 – 319
74. M. Rahm, R. Hoffmann, N. W. Ashcroft, *Chem. Eur. J*., **2016**, 22, 14625 – 14632
75. W. M. Haynes. CRC Handbook of Chemistry and Physics. 100 Key Points. CRC Press, London, 95th edition, **2014**
76. N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, *J. Machine Learning Res.,* 2014, **15**, 1929 – 1958
77. D. P. Kingma, J. L. Ba, Proceedings of the 3rd International Conference on Learning Representations (ICLR) 2014 arXiV: 1412.6980, 2014
78. C. Adamo, V. Barone, *J. Chem. Phys.*, **1999**, 110, 6158 – 6170
79. a) P. J. Hay, W. R. Wadt, *J. Chem. Phys.*, **1985**, 82, 270 – 283; b) P. J. Hay, W. R. Wadt, *J. Chem. Phys.*, **1985**, 82, 284 – 298; c) P. J. Hay, W. R. Wadt, *J. Chem. Phys.*, **1985**, 82, 299 – 310
80. a) I.S. Ufimtsev, T. J. Martínez*, J. Chem. Theo. Comp*., **2009**, 5, 2619; b) A. V. Titov, I. S. Ufimtsev, N. Luehr, T. J. Martínez, *J. Chem. Theo. Comp*., **2013**, 99, 213; c) C. Song, L. – P. Wang, T. J. Martínez, *J. Chem. Theo. Comp.,* **2016**, 12, 92 d) J. Kästner, J.M. Carr, T.W. Keal, W.Thiel, A. Wander, P. Sherwood, *J. Phys. Chem. A*, **2009**, 113, 11856
81. P. Pulay *Chem. Phys. Lett.*, **1980**, 73, 393
82. a) S. Goedecker, M. Teter, J. Hutter, *Phys. Rev. B*, **1996**, 54, 1703-1710; b) C. Hartwigsen, S. Goedecker, J. Hutter, *Phys. Rev. B*, **1998**, 58, 3641-3662; c) M. Krack, *Theor. Chem. Acc.*, **2005**, 114, 145-152

83. a) John P. Perdew, A. Ruzsinsky, G. I. Csonka, O. A. Vydrov, G. E. Scuseria, L. A. Constantin, X. Zhuo, K. Burke, *Phys. Rev. Lett.,* **2008**, 100, 136406; b) John P. Perdew, A. Ruzsinsky, G. I. Csonka, O. A. Vydrov, G. E. Scuseria, L. A. Constantin, X. Zhuo, K. Burke, *Phys. Rev. Lett.,* **2009***,* 102, 039902
84. a) The CP2K Developers Group. Disponible en https://www.cp2k.org/ (Consultado el 30 de junio de 2020) b) J. Hutter, M. Ianuzzi, F. Schiffmann, J. VandeVondele *WIRE – Computational Molecular Science*, **2014**, 4, 15 – 25
85. a) H. T. Stokes, D. M. Hatch, and B. J. Campbell, FINDSYM, ISOTROPY Software Suite, iso.byu.edu. b) H. T. Stokes and D. M. Hatch *J. Appl. Cryst.*, **2005**, 38, 237-238.
86. J.I. Gómez-Peralta, X. Bokhimi *J. Solid State Chemistry* **2020**, 285, 121253
87. a) https://en.wikipedia.org/wiki/Precision_and_recall (Consultado el 23 de noviembre de 2020)b) *Structuring Machine Learning Projects.* Curso 3 del programa de especialización en Aprendizaje Profundo. Disponible en https://www.coursera.org/
88. P. Sedgwick *BMJ* **2012**, 345, e4483
89. F. Bechstedt, F. Fuchs, G. Kresse *Phys. Status Solidi B*, **2009**, 246, 1877 – 1892
90. X. Zheng, A. J. Cohen, P. Mori-Sánchez, Z. Hu, W. Yang, *PRL*, **2011**, 107, 026403
91. R.J. Hasnip, K. Refson, M.I.J. Probert, J.R. Yates, S.J. Clark, C.J. Pickard *Phil. Trans. R. Soc. A,* **2014**, 372, 20130270
92. P. Verma, D. Truhlar, *Trends in Chemistry*, **2020**, 2, 302 – 318
93. a) R. D. Shannon, C. T. Prewitt, *Acta Cryst. B*, **1969**, 25, 925 – 946, b) R. D. Shannon, *Acta Cryst. A*, **1976**, 32, 751-767
94. I. D. Brown, K. R. Poeppelmeier, *Bond Valences*. Springer-Verlag, **2014** ISBN: 978-3-642-54968-7 DOI: 10.1007/978-3-642-54968-7
95. https://www.iucr.org/resources/data/datasets/bond-valence-parameters (Consultado el 23 de noviembre de 2020)
96. R. S. Mulliken, *J. Chem. Phys.*, **1955**, 23, 1833 – 1840
97. A. E. Reed, R. B. Weinstock, F. Weinhold, *J. Chem. Phys.*, **1985**, 83, 735 – 746
98. F. L. Hirshfeld, *Theoretica Chimica Acta*, **1977**, 44, 129 – 138
99. R. F. W. Bader, *Chem. Rev.* **1991**, 5, 893 – 828
100. T. A. Manz, D. S. Sholl, *J. Chem Theory Comput.* **2012**, 8, 2844 – 2867
101. F. Jensen, *An Introduction to Computational Chemistry*. 3rd Edition, Wiley. **2017.** ISBN-10: 1118825993

APÉNDICE A: CÓDIGOS

Éstos están también disponibles en los repositorios de GitHub:
https://github.com/gomezperalta/patolli
https://github.com/gomezperalta/perovskites_simulator
https://github.com/gomezperalta/phd_thesis

A.1 *cif_fixer.sh*

```
#!/bin/bash

for linea in `cat control.txt`;
do
        variable=$(echo $linea | cut -d "." -f 1)
        sed -i 's/HE/He/g' $variable.cif
        sed -i 's/LI/Li/g' $variable.cif
        sed -i 's/BE/Be/g' $variable.cif
        sed -i 's/NE/Ne/g' $variable.cif
        sed -i 's/NA/Na/g' $variable.cif
        sed -i 's/MG/Mg/g' $variable.cif
        sed -i 's/AL/Al/g' $variable.cif
        sed -i 's/SI/Si/g' $variable.cif
        sed -i 's/CL/Cl/g' $variable.cif
        sed -i 's/AR/Ar/g' $variable.cif
        sed -i 's/CA/Ca/g' $variable.cif
        sed -i 's/SC/Sc/g' $variable.cif
        sed -i 's/TI/Ti/g' $variable.cif
        sed -i 's/CR/Cr/g' $variable.cif
        sed -i 's/MN/Mn/g' $variable.cif
        sed -i 's/FE/Fe/g' $variable.cif
        sed -i 's/CO/Co/g' $variable.cif
        sed -i 's/NI/Ni/g' $variable.cif
        sed -i 's/CU/Cu/g' $variable.cif
        sed -i 's/ZN/Zn/g' $variable.cif
        sed -i 's/GA/Ga/g' $variable.cif
        sed -i 's/GE/Ge/g' $variable.cif
        sed -i 's/AS/As/g' $variable.cif
        sed -i 's/SE/Se/g' $variable.cif
        sed -i 's/BR/Br/g' $variable.cif
        sed -i 's/KR/Kr/g' $variable.cif
        sed -i 's/RB/Rb/g' $variable.cif
        sed -i 's/SR/Sr/g' $variable.cif
        sed -i 's/ZR/Zr/g' $variable.cif
        sed -i 's/NB/Nb/g' $variable.cif
        sed -i 's/MO/Mo/g' $variable.cif
        sed -i 's/TC/Tc/g' $variable.cif
        sed -i 's/RU/Ru/g' $variable.cif
        sed -i 's/RH/Rh/g' $variable.cif
        sed -i 's/PD/Pd/g' $variable.cif
        sed -i 's/AG/Ag/g' $variable.cif
        sed -i 's/CD/Cd/g' $variable.cif
        sed -i 's/IN/In/g' $variable.cif
```

```
        sed -i 's/SN/Sn/g' $variable.cif
        sed -i 's/SB/Sb/g' $variable.cif
        sed -i 's/TE/Te/g' $variable.cif
        sed -i 's/XE/Xe/g' $variable.cif
        sed -i 's/CS/Cs/g' $variable.cif
        sed -i 's/BA/Ba/g' $variable.cif
        sed -i 's/LA/La/g' $variable.cif
        sed -i 's/CE/Ce/g' $variable.cif
        sed -i 's/PR/Pr/g' $variable.cif
        sed -i 's/ND/Nd/g' $variable.cif
        sed -i 's/PM/Pm/g' $variable.cif
        sed -i 's/SM/Sm/g' $variable.cif
        sed -i 's/EU/Eu/g' $variable.cif
        sed -i 's/GD/Gd/g' $variable.cif
        sed -i 's/TB/Tb/g' $variable.cif
        sed -i 's/DY/Dy/g' $variable.cif
        sed -i 's/HO/Ho/g' $variable.cif
        sed -i 's/ER/Er/g' $variable.cif
        sed -i 's/TM/Tm/g' $variable.cif
        sed -i 's/YB/Yb/g' $variable.cif
        sed -i 's/LU/Lu/g' $variable.cif
        sed -i 's/HF/Hf/g' $variable.cif
        sed -i 's/TA/Ta/g' $variable.cif
        sed -i 's/RE/Re/g' $variable.cif
        sed -i 's/OS/Os/g' $variable.cif
        sed -i 's/IR/Ir/g' $variable.cif
        sed -i 's/PT/Pt/g' $variable.cif
        sed -i 's/AU/Au/g' $variable.cif
        sed -i 's/HG/Hg/g' $variable.cif
        sed -i 's/TL/Tl/g' $variable.cif
        sed -i 's/PB/Pb/g' $variable.cif
        sed -i 's/BI/Bi/g' $variable.cif
        sed -i 's/PO/Po/g' $variable.cif
        sed -i 's/AT/At/g' $variable.cif
        sed -i 's/RN/Rn/g' $variable.cif
        sed -i 's/FR/Fr/g' $variable.cif
        sed -i 's/RA/Ra/g' $variable.cif
        sed -i 's/AC/Ac/g' $variable.cif
        sed -i 's/TH/Th/g' $variable.cif
        sed -i 's/PA/Pa/g' $variable.cif
done
```

A.2 *create_dataframe.py*

```
import pandas as pd
import numpy as np
import pymatgen
import Wyckoff_finder as wf
import time

directorio = input('Provide the directory where the CIF-files are:'+ \
            '\n')
cif_list = input('Provide the txt-file where the CIF-file list is (no-extension)' +\
              '\n')

df = pd.read_csv(cif_list + '.txt', header=None)
df = df.rename(columns={0:'cif'})

formulas = list()
sgnums = list()
element_list = list()
wyck_list = list()
site_list = list()
atom_list = list()
not_added_cifs = list()

diccio = np.load('WyckoffSG_dict.npy').item()['wyckmul']

samples = df.shape[0]

start = time.time()
for row in range(df.shape[0]):

    try:
        estructura = pymatgen.Structure.from_file(directorio + \
                                        str(df['cif'][row]) + \
                                        '.cif')

        sgnum =
pymatgen.symmetry.analyzer.SpacegroupAnalyzer(estructura).get_space_group_number()
        formula = estructura.composition.reduced_formula
        elements = len(estructura.composition.elements)
        wyckdict = wf.wyckoff_occupation(ruta = directorio,
                                archivo = str(df['cif'][row]))
        sites = len(wyckdict)

        sg_diccio = diccio[str(sgnum).zfill(3)]

        atoms = 0
        for item in range(len(wyckdict)):

            label = wyckdict[item].keys()
            label = list(label)[0]
            fracsum = np.sum(list(wyckdict[item][label].values()))
            multiplicity = int(sg_diccio[label[0]])
```

```
            atoms += multiplicity*fracsum

        except:
            formula = None
            sgnum = None
            elements = None
            wyckdict = None
            sites = None
            atoms = None
            not_added_cifs += [df['cif'][row]]

        formulas += [formula]
        sgnums += [sgnum]
        element_list += [elements]
        wyck_list += [wyckdict]
        site_list += [sites]
        atom_list += [atoms]

df['formula'] = formulas
df['sgnum'] = sgnums
df['elements'] = element_list
df['WyckOcc'] = wyck_list
df['sites'] = site_list
df['atoms'] = atom_list

df = df[df['WyckOcc'] != None].reset_index(drop=True)

if len(not_added_cifs) != 0:
    with open('not_added_cif.txt','w') as f:
        for item in not_added_cifs:
            f.write(str(item)+'\n')
        f.close()

df.to_csv('cod_dataframe.csv',index=None)
df.to_pickle('cod_dataframe.pkl')

print('Process lasted',np.round(time.time()-start,2),
      's to treat',samples,'samples')
```

A.3 *Wyckoff_finder.py*

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Thu Oct  5 17:45:29 2017

@author: iG
"""
import pandas as pd
import pymatgen as pg
import numpy as np
import os
from pymatgen.symmetry.analyzer import SpacegroupAnalyzer

def drop_characters(in_str):
    '''
    Elimina los caracteres numericos y los signos + y -.
    Esta funcion se tomo de stackoverflow.com
    Parametros: string
    Regresa: string sin digitos y signos +-
    '''
    char_list = "1234567890+-"
    for char in char_list:
        in_str = in_str.replace(char, "")

    return in_str

def move_element(odict, thekey, newpos):
    '''
    Esta funcion se tomo de stackoverflow.com
    Tiene como objetivo cambiar de posicion la clave de un diccionario.
    Parametros:
        odict: diccionario
        thekey: la clave que se quiere mover
        newpos: posicion a la que se quiere mover
    Regresa:
        diccionario con la clave en la posicion requerida
    '''

    odict[thekey] = odict.pop(thekey)
    i = 0
    for key, value in odict.items():
        if key != thekey and i >= newpos:
            odict[key] = odict.pop(key)
        i += 1
    return odict

def wyckoff_occupation(ruta='/home/ivan/Alles/',archivo='1010902'):

    '''
    Esta funcion permite encontrar tanto todas las multiplicidad y las etiquetas de los sitios
    de Wyckoff. Se vale de la libreria de pymatgen para conseguir este fin
```

```
    Parametros:
        ruta(string): Direccion donde se encuentra el archivo cif.
        archivo(string): Nombre del archivo cif.
    Regresa:
        Diccionario de los elementos de la formula con un array de todas las multiplicidades
        y sitios de Wyckoff
    '''

    estructura=pg.Structure.from_file(str(ruta)+str(archivo)+'.cif')
    eqpos=SpacegroupAnalyzer(estructura).get_symmetry_dataset()

    df=pd.DataFrame({'eq_atoms': eqpos['equivalent_atoms'],'Wyckoff': eqpos['wyckoffs']})

    ocupacion=[]
    clave={}
    for i in range(len(estructura)):
        ocupacion.append(estructura[i].species_string.split(','))
        #print(ocupacion)
        for occ in range(len(estructura[i].species_string.split(','))):
            especie=estructura[i].species_string.split(',')[occ].split(':')[0].lstrip(' ')
            clave.setdefault(i,[]).append(especie)

    df['elemento']=drop_characters(df['eq_atoms'].map(clave))

    df=df.groupby(['Wyckoff','eq_atoms']).size().reset_index(name='multiplicidad')

    df=df[['multiplicidad','Wyckoff','eq_atoms']]
    df=df.drop_duplicates()
    df=df[['eq_atoms','Wyckoff']]
    df=df.reset_index(drop=True)
    df=df.drop_duplicates()

    dicc={}
    for item in range(len(ocupacion)):

            frac={}
            if len(ocupacion[item]) == 1:
                try:
                    frac[ocupacion[item][0].split(':')[0].lstrip(' ')] =
np.round(float(ocupacion[item][0].split(':')[1].lstrip(' ')),5)
                except:
                    frac[ocupacion[item][0]] = 1

            else:
                for Z in ocupacion[item]:
                    frac[Z.split(':')[0].lstrip(' ')] = np.round(float(Z.split(':')[1].lstrip(' ')),5)

            dicc[item]=frac

    df=df[['Wyckoff', 'eq_atoms']]
    df['eq_atoms']=df['eq_atoms'].map(dicc)

    diccionario={}
```

```
    for row in range(len(df)):
        diccionario[row]={df['Wyckoff'][row]:df['eq_atoms'][row]}

    return diccionario


def wyckoff_positions(ruta='/home/ivan/Alles/',archivo='1010902'):

    '''
    Esta funcion permite encontrar tanto todas las multiplicidad y las etiquetas de los sitios
    de Wyckoff. Se vale de la libreria de pymatgen para conseguir este fin
    Parametros:
        ruta(string): Direccion donde se encuentra el archivo cif.
        archivo(string): Nombre del archivo cif.
    Regresa:
        Diccionario de los elementos de la formula con un array de todas las multiplicidades
        y sitios de Wyckoff
    '''

    estructura=pg.Structure.from_file(str(ruta)+str(archivo)+'.cif')
    eqpos=SpacegroupAnalyzer(estructura).get_symmetry_dataset()
    archivo=SpacegroupAnalyzer(estructura).get_conventional_standard_structure()
    text=str(archivo)

    sitios=int(text.split('\n')[4].split('(')[1].split(')')[0])

    abc=[float(item) for item in list(filter(None,text.split('\n')[2].split(':')[1].split(' ')))]
    angles=[float(item) for item in list(filter(None,text.split('\n')[3].split(':')[1].split(' ')))]
    lista=text.split('\n')[-sitios:]

    newlist=[list(filter(None,line.split(' '))) for line in lista]
    newlist = [[item[0]] + [str(item[1:-3])] +  item[-3:] for item in newlist]
    newlist=np.asarray(newlist)
    #print(archivo)
    motif=pd.DataFrame(newlist)[[1,2,3,4]]
    motif[2]=[float(i) for i in motif[2].values]
    motif[3]=[float(i) for i in motif[3].values]
    motif[4]=[float(i) for i in motif[4].values]

    volumen=abc[0]*abc[1]*abc[2]*np.sqrt(1-(np.cos(np.deg2rad(angles[0])))**2-
(np.cos(np.deg2rad(angles[1])))**2-
(np.cos(np.deg2rad(angles[2])))**2+2*np.cos(np.deg2rad(angles[0]))*np.cos(np.deg2rad(angles[1]))*n
p.cos(np.deg2rad(angles[2])))


    df=pd.DataFrame({'eq_atoms': eqpos['equivalent_atoms'],'Wyckoff': eqpos['wyckoffs']})

    df=df.groupby(['Wyckoff','eq_atoms']).size().reset_index(name='multiplicidad')
    df=df[['multiplicidad','Wyckoff','eq_atoms']]
    df=df.drop_duplicates()
    df=df[['eq_atoms','Wyckoff']]
    df=df.reset_index(drop=True)
    df=df.drop_duplicates()
```

```
    #print(df)
    #motif=pd.read_csv('motif.csv', header=None, delim_whitespace=True)[[1,2,3,4]]
    motif.columns = np.arange(len(motif.columns))
    df=pd.concat([df,(motif.loc[list(df['eq_atoms'].values),1:]).reset_index(drop=True)], axis=1,
join='inner')
    #print(df)
    idx = sorted(df['eq_atoms'].values) + [len(motif)]
    #print(idx)
    diccionario={}
    for row in range(len(df)):
        #vector=np.around(df.iloc[row,2:].values.astype(np.double),decimals=4)
        init = df['eq_atoms'][row].item()
        finit = idx[idx.index(init) + 1]
        #print(init,finit)
        diccionario[row]={df['Wyckoff'][row] : motif.iloc[init:finit,-3:].values}

    return diccionario,motif, angles, abc

def wyckoff_finder(ruta='/home/ivan/Alles/',archivo='1010902'):

    '''
    Esta funcion permite encontrar tanto todas las multiplicidad y las etiquetas de los sitios
    de Wyckoff. Se vale de la libreria de pymatgen para conseguir este fin
    Parametros:
        ruta(string): Direccion donde se encuentra el archivo cif.
        archivo(string): Nombre del archivo cif.
    Regresa:
        Diccionario de los elementos de la formula con un array de todas las multiplicidades
        y sitios de Wyckoff
    '''

    estructura=pg.Structure.from_file(str(ruta)+str(archivo)+'.cif')
    eqpos=SpacegroupAnalyzer(estructura).get_symmetry_dataset()

    df=pd.DataFrame({'eq_atoms': eqpos['equivalent_atoms'],'Wyckoff': eqpos['wyckoffs']})

    clave={k:v for k,v in enumerate([[occ.split(':')[0].lstrip(' ') for occ in
estructura[i].species_string.split(',')] for i in range(len(estructura))])}

    df['elemento']=drop_characters(df['eq_atoms'].map(clave))
    df=df.groupby(['Wyckoff','eq_atoms']).size().reset_index(name='multiplicidad')
    df=df[['multiplicidad','Wyckoff','eq_atoms']]
    df=df.drop_duplicates()
    df['mult_wyc']=df['multiplicidad'].map(str)+df['Wyckoff']
    df=df[['eq_atoms','mult_wyc']]
    df=df.reset_index(drop=True)
    df=df.drop_duplicates()

    dicc1={k:[v] for k,v in zip(df['eq_atoms'],df['mult_wyc'])}

    dicc={}
    for keys in dicc1.keys():
        for el in range(len(clave[keys])):
```

```
            dicc.setdefault(clave[keys][el].rstrip('0123456789+-.'),[]).append(dicc1[keys])
    return dicc
```

A.4 patolli.py

```
# -*- coding: utf-8 -*-
"""
Created on Fri Mar 16 12:24:33 2018

@author: iG
"""

import keras.layers as layers
import keras.models as models
import keras.utils as kutils
import keras.callbacks as callbacks
import keras.optimizers as optimizer
from sklearn.utils import shuffle
from sklearn.metrics import confusion_matrix
from sklearn.metrics import precision_recall_fscore_support as PRFS

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import math
import itertools
import time
import copy
import os

np.random.seed(10)

def lab2symm(sgnum='001',label='a'):
    """
    Convierte los simbolos de Wyckoff
    en sus multiplicidades.
    Necesita que el archivo WyckoffSG_dict.npy
    esté en la misma carpeta donde se tiene
    patolli.py
    Parametros:
        sgnum: string con el numero de grupo espacial
               en formato 000 (ejemplo: '221','062','002').
        label: string con algun simbolo de Wyckoff
               del grupo espacial proporcionado
    Regresa:
        string con la multiplicidad del simbolo de Wyckoff
        ingresado
    """
    wyck_dic=np.load('WyckoffSG_dict.npy').item()['wycksym']
    return wyck_dic[sgnum].get(label)

def create_dictionary(file='dictionary'):
    """
```

```
    Parametros:
       file: un archivo de texto que contiene los grupos
          espaciales con sus respectivos sitios de Wyckoff
          ocupados. Este archivo de texto se transforma a
          un diccionario de Python.
    Regresa:
       diccio: Diccionario de Python.
    """
    start=time.time()
    f=list(filter(None,open(str(file)+'.txt','r').read().split('\n')))

    sg_ikeys=[f.index(sg) for sg in f if 'spacegroup' in sg]+[len(f)]
    sg_keys=[str(int(sg.split(':')[1])).zfill(3) for sg in f if 'spacegroup' in sg]

    diccio={}

    for item in range(len(sg_ikeys)-1):
       text=f[sg_ikeys[item]+1:sg_ikeys[item+1]]
       option=[text.index(i) for i in text if 'option' in i]+[len(text)]

       dicc_1={}
       for inneritem in range(len(option)-1):
          innertext=text[option[inneritem]+1:option[inneritem+1]]

          values=[]
          for letra in innertext:

             temp_values=[i for j in [s.split(',') for s in [letra.split(':')[1].replace(' ','')]] for i in j]
             values=values+temp_values
          dicc_1[inneritem]=[lab2symm(sgnum=sg_keys[item],label=j) for j in sorted(values)]

       diccio[sg_keys[item]]=dicc_1
    print('Crystal structure definition text file already converted into a Python dictionary in',
         round(time.time()-start,2),' s')
    return diccio

def ctrl_dictionary(archivo='model_control_file'):
    """
    Parametros: Un archivo de texto que contiene todas las caracteristicas
             (hiperparametros) con los que se entrenaran las RR. NN. AA.
             Este archivo es convertido a un diccionario de Python,
             el cual es usado por la funcion entrenador.
    Regresa: Un diccionario de Python, requerido por la funcion entrenador.
    """
    f=list(filter(None,open(str(archivo)+'.txt','r').read().split('\n')))

    sg_ikeys=[f.index(sg) for sg in f if 'NAME' in sg]+[len(f)]

    diccio={}
    for item in range(len(sg_ikeys)-1):
       text = f[sg_ikeys[item]:sg_ikeys[item+1]]
       key = [entry.split(':')[0] for entry in text]
       value = [entry.split(':')[1] for entry in text]
```

```
        diccio[item] = {k:v for k,v in zip(key,value)}

    return diccio

def create_collection(database='/home/bokhimi/.cod/cod-db.pkl', sites=-1, elements=-1,
                maxatoms=-1, dictionary='diccionario'):
    """
    Parameters:
        database: A pickle file which has information about the space group
                and the occupied sites in the structure.
                It must be specified with extension
        dictionary: A txt - file which contains the spacegroups and the symmetry
                sites which define a given structure.
    Returns:
        final_data: A pandas DataFrame which contains True and False samples.
    """


    diccionario = create_dictionary(file=str(dictionary))

    sitios=max([len(j) for i in diccionario.values() for j in i.values()])
    #print(sitios)
    start=time.time()
    print('Loading database. This may take some time...')
    df = pd.read_pickle(database)
    print('Database loaded. This process took ',np.round(time.time()-start,2))

    if sites == -1:
        df = df[df['sitios'] <= sitios].reset_index(drop=True)
    else:
        df = df[df['sitios'] <= sites].reset_index(drop=True)

    if maxatoms != -1:
        df = df[df['atoms'] <= maxatoms].reset_index(drop=True)
    '''
    if elements != -1:
        data = df[df['elements'] >= elements].reset_index(drop=True)
    else:

    '''
    data = df

    trydata=data.loc[data['sgnum'].isin([int(i) for i in \
                list(diccionario.keys())])].reset_index(drop=True)
    wyck_dic=np.load('WyckoffSG_dict.npy').item()['wycksym']

    trydata['labels'] = [[list(item.keys())[0] for item in \
          list(trydata['WyckOcc'][row].values())] for row in range(len(trydata))]

    trydata['symmetry'] = [[wyck_dic[str(trydata['sgnum'][row]).zfill(3)].get(letra) \
          for letra in trydata['labels'][row]] for row in range(len(trydata))]
```

```
    target=[]
    for row in range(len(trydata)):

        spacegroup = str(trydata['sgnum'][row]).zfill(3)
        comparador=list(diccionario[spacegroup].values())

        if trydata['symmetry'][row] in comparador:
            target.append(True)
        else:
            target.append(False)

    target=pd.Series(target,name='target')
    trydata=trydata.join(target)
    print('True data identified')

    data_true=trydata[trydata['target'] == True].reset_index(drop=True)
    exclude_cif=trydata[trydata['target'] == True]['cif']
    sitios=list(set(data_true['sitios']))
    data_rem=df.loc[~df['cif'].isin(exclude_cif)].reset_index(drop=True)

    final_data=data_true[['cif','formula','WyckOcc','sgnum','sitios',
                          'atoms','elements','target']]
    for item in sitios:
        data_temp=data_rem[data_rem['sitios'] == item].reset_index(drop=True)

        cantidad=len(data_true[data_true['sitios'] == item])


vector=np.random.permutation(np.random.permutation(np.random.permutation(np.arange(len(data_te
mp)))))[:cantidad]
        data_temp=data_temp.take(vector)
        data_temp['target']=False
        final_data=pd.concat((final_data, data_temp), ignore_index=True)

    print('Crystal compounds collection to train the ANNs created in',
          round(time.time()-start,2),' s')

    final_data = final_data[['cif','formula','sgnum','WyckOcc','sitios',
                             'elements','atoms','target']]

    if elements != -1:
        false_positive = final_data[final_data['target'] == True][final_data['elements'] < elements].index
        if len(false_positive) != 0:
            numcif_todrop = len(false_positive)
            false_todrop = np.random.choice(final_data[final_data['target'] == False].index,numcif_todrop)
            idx_todrop = np.append(false_positive, false_todrop)

            final_data = final_data.drop(final_data.index[idx_todrop])
            final_data = final_data.reset_index(drop=True)

    final_data.to_csv('compounds_collection.csv', index=None)

    return final_data
```

```
def raw_features_extractor(database='/home/bokhimi/.cod/cod-db.pkl', sites=-1, elements = -1,
maxatoms= -1,
                dictionary='diccionario', features='datosrahm.csv',
                site_normalization=False):

    """
    Parameters:
        database: A pickle file which contains information about the spacegroups and
                  the occupied symmetry sites. This must be specified with extension.
        dictionary: A txt - file which contains the symmetry site occupation for each
                  spacegroup where a given structure crystallizes. This must not be
                  specified with extension
        sites: Constriction to choose structures with a maximum amount of sites.
              By default, there is not a constriction
        features: A csv - file which contains the features to be use for each present
              element in the sites of the structure.
        include: if this is equal to 'atoms', info about atoms per occupied sites is
              included in the features. If this is 'mult', multiplicity of occupied
              sites is included in the features. If empty, i.e. '', nothing of above
              mentioned is included
    Returns:
        X: A matrix of samples x sites x features.
        y: A True - False vector
        s: A matrix of samples x multiplicity for each site.
        df: A pandas DataFrame with True/False values.
    """

    df=create_collection(database=database,sites=sites, elements=elements, maxatoms=maxatoms,
                    dictionary=dictionary)

    start=time.time()

    datos=pd.read_csv(features)
    datos=datos.fillna(-1)

    dicc=dict(datos[['Symbol','Z']].values)

    dicc['D']=1
    dicc['Bk']=97
    dicc['Cf']=98
    dicc['Es']=99
    dicc['Fm']=100
    dicc['Md']=101
    dicc['No']=102
    dicc['Lr']=103

    max_sitios = max(df['sitios'].values)

    df=df[df['sitios'] <= max_sitios].reset_index(drop=True)

    X=np.zeros((len(df),max_sitios,104))
    y=np.zeros((len(df),1))
```

```
    mult=np.zeros((len(df),max_sitios))
    wyckmul=np.load('WyckoffSG_dict.npy').item()['wyckmul']

    for row in range(len(df)):

        item=df['WyckOcc'][row]
        sitios=list(item.values()) #Diccionario de elementos con fracciones de ocupación en ese sitio
        sitocc=np.zeros((len(sitios),104)) #Vector para 104 elementos de la tabla periódica
        spacegroup = str(df['sgnum'][row]).zfill(3)

        try:

            s=[int(wyckmul[spacegroup][i]) for j in [list(item.keys()) for item in \
              sitios] for i in j]

        except:
            print('There exists an error concerning with the space group of CIF ', df['cif'][row],'\n')
            print('Please check in www.crystallography.net to provide the correct space group number of
that CIF',
                  '\n','\n')
            spacegroup=input('Give me the correct spacegroup:'+'\n'+'\n')
            s=[int(wyckmul[spacegroup][i]) for j in [list(item.keys()) for item in \
              list(df['WyckOcc'][row].values())] for i in j]

        occs=[]
        for i in range(len(sitios)):

            for j in list(sitios[i].values()):

                ocupacion=np.array(list(j.values()))
                llaves=[llave.replace('+','').replace('-','').replace('1',
                       '').replace('2','').replace('3','').replace('4',
                              '') for llave in np.array(list(j.keys()))]
                llaves=[llave.replace('.','') for llave in llaves]
                llaves=[llave.replace('5','').replace('6','').replace('7',
                       '').replace('8','').replace('9','').replace('0',
                              '') for llave in llaves]
                vector=np.zeros((1,104))
                occs=[sum(ocupacion)]+occs

                try:

                    idx=[dicc[k] for k in llaves]

                except:

                    print(' ELEMENTO NO IDENTIFICADO EN LA LISTA ',llaves,'\n',
                         'REVISA EL SIGUIENTE CIF PARA HACER LA CORRECCION:','\t',df['cif'][row])

                    former = input('Elemento Incorrecto: ')
                    current = input('Elemento Correcto: ')

                    llaves=[current if x == former else x for x in llaves]
```

```
                idx=[dicc[k] for k in llaves]


            for k in idx:
                vector[0][k-1] = ocupacion[idx.index(k)]


        sitocc[i]=vector
    '''
    if include == 'atoms':
        s=list(np.multiply(np.round(occs,4),s))
    '''
    while sitocc.shape[0] != max_sitios:
        sitocc=np.concatenate((np.zeros((1,104)),sitocc))
        s=[0]+s

    X[row,:,:]=sitocc
    y[row]=df['target'][row]
    mult[row]=s

S = np.expand_dims(mult,axis=2)
features=datos.iloc[:,2:].values
x=X[:,:,:96]

fracsum = np.expand_dims(np.sum(x,axis=2), axis=2)

if site_normalization == True:
    x = np.nan_to_num(x/fracsum)

x=np.dot(x,features)
'''
if include == 'atoms' or include == 'mult':
    suma = np.stack((np.sum(S, axis=1),)*max_sitios,axis=1)
    x = np.concatenate((x,S/suma),axis=2)
'''
print('Atomic radii and electronegativities for each Wyckoff site extracted in',
      round(time.time()-start,2),' s')

np.save('raw_features', x)
np.save('output_values', y)
np.save('multiplicities', S)
np.save('occupation_fractions', S)

return x, y, S, fracsum, df

def compute_quotients(X = np.zeros((1,1,2))):
    """
    Returns the atomic radii pair quotients and the atomic radii
    pair sum - quotients as a numpy array. Thjs is the first part of
    all the features used to train the ANNs
    Parameters:
        X: A numpy array, which is created with the function raw_features_extractor
    Returns:
```

```
        X: A numpy array of dimension [samples,1,features]
    """

    start=time.time()
    rad = X[:,:,1]

    X = np.reshape(X,(X.shape[0],1,X.shape[1]*X.shape[2]))

    drad = np.asarray([[item[0]/item[1] if item[1] != 0 else 0 for item in
list(itertools.combinations(rad[sample],2))] \
                       for sample in range(X.shape[0])])

    dradsum = np.asarray([[item[0]/item[1] if item[1] != 0 else 0 for item in itertools.combinations([ \
                       item[0]+item[1] for item in list(itertools.combinations(rad[sample],2))], 2)] \
                       for sample in range(drad.shape[0])])

    drad = np.reshape(drad,(drad.shape[0],1,drad.shape[-1]))
    drads = np.reshape(dradsum,(dradsum.shape[0],1,dradsum.shape[-1]))

    X = np.concatenate((drad,drads), axis=2)
    print('Geometric and packing factors computed in', round(time.time()-start,2),' s')
    np.save('X', X)

    return X

def append_local_functions(X = np.zeros((1,1,1)), df = pd.DataFrame(),
local_function='fij_2.0_25_diccio'):
    """
    Returns the features with the local functions. In case the local function
    does not exist for a sample in the collection, this is deleted and the
    collection is updated.
    Parameters:
        X: The numpy array created with compute_quotients
        df: The pandas DataFrame created with raw_features_extractor
        local_function: The numpy dictionary having the local function to use.
    Returns:
        X: The numpy array with all neccesary features for the ANNs.
        df: The pandas DataFrame updated.
    """
    start = time.time()
    print('The dictionary ' + local_function + ' will be used for local functions')
    fij = np.load(local_function + '.npy').item()

    delrow = list()
    n = np.max(df['sitios'])

    f = np.zeros((df.shape[0],n,n))

    for row in range(df.shape[0]):
        if df['cif'][row] not in fij.keys():
            delrow += [row]
        else:
            loc = fij[df['cif'][row]]
```

```
            s = loc.shape[1]
            f[row,-s:,-s:] = loc

    if len(delrow) != 0:

        print('The compounds with the next cifs will be deleted since ',
              'their local functions are not currently available')
        print([df['cif'][i] for i in delrow])
        print('The compound collection will be updated')

        totake = [i for i in range(df.shape[0]) if i not in delrow]
        df = df.take(totake).reset_index(drop=True)
        X = X[totake]
        f = f[totake]
        df.to_csv('compounds_collection.csv', index=None)

    fn = np.zeros((f.shape[0], f.shape[1], f.shape[2] - 1))
    for item in range(f.shape[0]):
        delec = f[item]
        delec = delec[~np.eye(delec.shape[0], dtype=bool)].reshape(delec.shape[0],-1)
        fn[item] = delec

    f = fn
    f = f.reshape((f.shape[0], 1, f.shape[1]*f.shape[2]))

    X = np.concatenate((X,f), axis = 2)
    print('Local functions appended to features in ', round(time.time()-start,2),' s')
    return X, df

def split_collection(X = np.zeros((1)), df = pd.DataFrame(), frac = 0.15):
    """
    Splits the complete compounds collection in two sets:
        one for training and cross - validation and another for testing.
    Parameters:
        X: A numpy array with the features of all compounds in the collection to split.
        df: A pandas DataFrame with all the compounds.
        frac: The fraction reserved to create the test - set. If frac equals zero,
        arguments are passed to returns without modifications.
    Returns:
        Xtraval: A numpy array with the features of the compounds in the
        training and cross - validation sets. This is saved as Xtraval.
        Xtest: A numpy array with the features of the compounds in the
        test set. This is saved as Xtest.
        dftraval: A panda DataFrame with the compounds in the training and
        cross validation sets. This is saved as dftraval.
        dftest: A panda DataFrame with the compounds in the test set. This is
        saved as dftest.
    """
    if frac != 0:

        tidx = df[df['target'] == True].index
        fidx = df[df['target'] == False].index
```

```
        ttest = np.random.choice(tidx, size = int(frac*len(tidx)), replace = False)
        ftest = np.random.choice(fidx, size = int(frac*len(fidx)), replace = False)

        ttest = [i for i in ttest]
        ftest = [i for i in ftest]

        ttraval = [i for i in tidx if i not in ttest]
        ftraval = [i for i in fidx if i not in ftest]

        traval = ttraval + ftraval
        test = ttest + ftest

        Xtraval = X[traval]
        Xtest = X[test]

        dftraval = df.take(traval).reset_index(drop=True)
        dftest = df.take(test).reset_index(drop=True)

        np.save('Xtraval', Xtraval)
        np.save('Xtest', Xtest)

        dftraval.to_csv('dbtraval.csv', index=None)
        dftest.to_csv('dbtest.csv', index=None)

    else:
        Xtraval = X
        dftraval = df
        Xtest = None
        dftest = None

    return Xtraval, Xtest, dftraval, dftest

def feat_stand(X = np.zeros((1,1,1))):
    """
    Scales de features with standarisation.
    During the function execution, a dictionary is saved in formats
    *.npy and *.txt. This dictionary contains both mean and standard
    deviation values of each feature
    Parameters:
    X: an array containing all the samples with their input data
    Returns:
    X: The same array but standarised
    """
    average = np.mean(X, axis=0)
    stdev = np.std(X, axis=0)

    X = (X - average)/stdev

    dicfeatstand = {'mean':average,'std':stdev}
    np.save('feature_standarisation',dicfeatstand)

    with open('feature_standarisation.txt','w') as f:
        f.write('X matrix has dimensions '+str(X.shape[0])+' samples x ' + \
```

```
            str(X.shape[1]) + ' sites x ' + str(X.shape[2]) + \
            ' features'+'\n'+'\n')
    f.write('Features - mean:'+'\n'+'\n')
    f.write(str(average)+'\n'+'\n')
    f.write('Features - std:'+'\n'+'\n')
    f.write(str(stdev))
    f.close()

return X
```

A.5 neighdist.py

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Tue Oct  2 10:29:44 2018
@author: iG
"""

import pandas as pd
import numpy as np
import itertools as it



def positions(pos = dict(),angles=[],abc=[], dist=100):

    mult = [np.asarray(list(pos[i].values())).shape[1] for i in range(len(pos))]

    pos = np.concatenate([np.asarray(list(pos[item].values())) \
                      for item in range(len(pos))],axis=1)

    mot = pos.reshape((pos.shape[1],pos.shape[2]))

    volumen=abc[0]*abc[1]*abc[2]*np.sqrt(1-(np.cos(np.deg2rad(angles[0])))**2 - \
            (np.cos(np.deg2rad(angles[1])))**2 -\
            (np.cos(np.deg2rad(angles[2])))**2 + \

2*np.cos(np.deg2rad(angles[0]))*np.cos(np.deg2rad(angles[1]))*np.cos(np.deg2rad(angles[2])))


matrix=np.array([[abc[0],abc[1]*np.cos(np.deg2rad(angles[2])),abc[2]*np.cos(np.deg2rad(angles[1]))],
                 [0,abc[1]*np.sin(np.deg2rad(angles[2])),abc[2]*(np.cos(np.deg2rad(angles[0]))-
np.cos(np.deg2rad(angles[1]))*np.cos(np.deg2rad(angles[2])))/np.sin(np.deg2rad(angles[2]))],
                 [0,0,volumen/(abc[0]*abc[1]*np.sin(np.deg2rad(angles[2])))]])

    mt = np.round(matrix,5)

    n = int(np.ceil((dist+10)/np.min(mt[mt > 0])))

    if n > 30:
       print('Number of unit cell for each half - dimension is ',n,'\n')
       n = 30

    tras = list(it.product(np.arange(-n,n+1),repeat=3))

    zero = tras.index((0,0,0))
    tras = np.asarray(tras)

    h,w = mot.shape
    d = tras.shape[0]

    tras = tras.T
```

```
    mot = mot[:,:,np.newaxis]
    tras = tras[np.newaxis, :, :]


    mot = np.repeat(mot, d, axis=2)
    tras = np.repeat(tras, h, axis=0)

    mot = tras + mot
    mot = np.swapaxes(mot,1,2)
    mot = mot.astype(float)
    mot = np.matmul(mot,matrix)

    return mot, zero, n, mult

def exponential(x,n = 1, coef = 1):
    return np.exp(-coef*np.power(x,n))

def potential(x, n = 1, coef = 1):
    return np.power(coef*x,-n)

def angcos(x, dist = 5):
    return np.multiply(0.5*(np.cos(np.pi*x/dist) + 1), x <= dist)

def angtanh(x, dist = 5):
    return np.multiply(np.power(np.tahn(1-x/dist),3), x <= dist)

def rij(mult=[1,1,3], p = np.zeros((1,1,1)), zero = 1, dist=100,
        sites = 4, radii = [1,1,1]):

    radii = [item for item in radii if item != 0]
    l = [sum(mult[:i]) for i in range(len(mult)+1)]
    rij = list()

    for i,atrad_i in zip(range(1,len(l)),radii):
        r = p - p[l[i-1],zero,:]
        r = np.linalg.norm(r,axis=2)

        for j,atrad_j in zip(range(1,len(l)), radii):
            coef = (atrad_i + atrad_j)**(-2)
            init = l[j-1]
            fin = l[j]
            rj = r[init:fin,:]
            rj = rj[rj <= dist]
            rj = rj[rj != 0]
            rj = np.sum(exponential(x = rj, n =2, coef=coef)*angcos(x=rj,dist=dist))
            rij += [rj]

    lon = int((len(rij))**(1/2))
    rij = np.asarray(rij).reshape((lon,lon))

    s = sites
```

```
if lon != s:
    rij = np.concatenate((np.zeros((rij.shape[0],s-lon)),rij),axis=1)
    rij = np.concatenate((np.zeros((s-lon,s)),rij), axis=0)

return rij
```

A.6 simulate_compounds.py

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Mon Sep 24 16:53:35 2018
@author: iG
"""

import pandas as pd
import numpy as np
import keras.models as models
import matplotlib.pyplot as plt
import matplotlib.colors
import neighdist
import itertools

plt.rcParams['figure.figsize']=(12,9)
plt.rcParams['font.size'] = 16.0
plt.rcParams['font.family'] = 'sans-serif'

cmap = plt.cm.get_cmap('RdYlBu')
norm = matplotlib.colors.BoundaryNorm(np.arange(0,1.1,0.1), cmap.N)

"""An auxiliary file used to compute the features needed by the ANN"""
datos = pd.read_csv('datosrahm.csv')

maindict = {}
for row in range(datos.shape[0]):
    maindict[datos['Symbol'][row]] = \
    datos.iloc[row,:][['elecPau','atradrahm']].values

"""The Artificial Neural Network is loaded as well as the parameters to do the feature standarization"""
model_dict = np.load('dictionary_upto_4Wyckoffsites.npy').item()
model = models.load_model('patolli_upto_4Wyckoffsites.h5')

wyckcub = {0:{'A' : np.asarray([[0.0,0.0,0.0]])},
           1:{'B' : np.asarray([[0.5,0.5,0.5]])},
           2:{'X' : np.asarray([[0.5,0.5,0.0],[0.5,0.0,0.5],[0.0,0.5,0.5]])}}

def compute_quotients(X = np.zeros((1,1,2))):
    """
    Returns the atomic radii pair quotients and the atomic radii
    pair sum - quotients as a numpy array. Thjs is the first part of
    all the features used to train the ANNs
    Parameters:
        X: A numpy array, which is created with the function raw_features_extractor
    Returns:
        X: A numpy array of dimension [samples,1,features]
    """

    rad = X[:,:,1]
```

```
    X = np.reshape(X,(X.shape[0],1,X.shape[1]*X.shape[2]))

    drad = np.asarray([[item[0]/item[1] if item[1] != 0 else 0 for item in
list(itertools.combinations(rad[sample],2))] \
                       for sample in range(X.shape[0])])

    dradsum = np.asarray([[item[0]/item[1] if item[1] != 0 else 0 for item in itertools.combinations([ \
                    item[0]+item[1] for item in list(itertools.combinations(rad[sample],2))], 2)] \
                    for sample in range(drad.shape[0])])

    drad = np.reshape(drad,(drad.shape[0],1,drad.shape[-1]))
    drads = np.reshape(dradsum,(dradsum.shape[0],1,dradsum.shape[-1]))

    X = np.concatenate((drad,drads), axis=2)
    print('Geometric and packing factors computed')

    return X

def scan_nnoutputs(elements = [['Na'],['V'], ['O']],
                   compositions = [[1.0], [1.0],[1.0]],
                   maxdev = 0.2, stepsize = 0.01,
                   name = ''):
    """
    This function assesses the probability to crystallize as a perovskite structure
    among different lattice parameter. The perovskite structures are considered in the
    aristotype form (space group No. 221).
    Besides, csv- and png- files are saved containing the probabilities against the
    explored lattice parameters.
    Parameters:
        elements: a list of list, which specifies the atoms located in the cuboctahedral,
                        octahedral and vertex sites.
        compositions: a list of list, which indicates the ocupation fraction of each atom
                        in the sites.
        maxdev: float, maximum deviation from the calculated lattice parameter. The lattice
                        parameter is proposed to be the sum of the atomic radii of the atoms in the
                        octahedral and vertex sites. Default, 0.2.
        stepsize: float, stepsize used to scan the perovskite probability among the range
                        defined by the maxdev. Default,  0.01.
        name: string, the name for the saved files. If it is not given, the name will be
                        the formula of the simulated compound.
    Returns:
        y: Numpy array, containing the probabilies to crystallize as a perovskite structure.
        scanned_latpar: Numpy array, containing the lattice parameter used to perform the simulation.
    """

    scann = np.arange(1 - maxdev,
                      1 + maxdev + stepsize,
                      stepsize)

    multiplicities = [[1],[1],[3]]
    sidx = [np.asarray(i)*np.asarray(j) for i,j in zip(compositions, multiplicities)]
    sidx = [k for h in sidx for k in h]
```

```
elements2 = [k for j in elements for k in j]
formula = [i + '$_{' + "%.2f" % j + '}$' \
          for i,j in zip(elements2, sidx)]
formula = ''.join([i for i in formula])

primfeat = [np.dot(np.asarray(subind), np.asarray([maindict.get(item,None) \
         for item in site])) for site, subind in zip (elements, compositions)]

primfeat = np.asarray(primfeat)
primfeat = np.concatenate((np.zeros((1,primfeat.shape[1])),primfeat), axis = 0)

latpar = primfeat[2][1] + primfeat[3][1]

multiplicities = np.asarray(multiplicities)

primfeat = primfeat.reshape((1, primfeat.shape[0],primfeat.shape[1]))
x = compute_quotients(X=primfeat)

scann = scann**3

scanned_latpar = latpar*(scann**(1/3))

dist = 25

radii = primfeat[:,:,1]

elec = primfeat[:,:,0]
elec = elec.reshape((4,1))

delec = np.repeat(elec[:,np.newaxis],4,axis=2) - \
        np.repeat(elec[:,np.newaxis],4,axis=2).T
delec = delec.reshape((delec.shape[0],delec.shape[2]))

fr = np.zeros((len(scann),4,4-1))
for item in range(len(scanned_latpar)):

    p, z, n, m = neighdist.positions(pos = wyckcub,
                                angles = [90,90,90],
                                abc = [scanned_latpar[item],]*3,
                                dist = dist)
    r = neighdist.rij(mult=m,p=p,zero=z,
                    dist=dist, radii = np.ravel(radii))

    temp = np.multiply(r,delec)
    temp = temp[~np.eye(temp.shape[0],
                dtype=bool)].reshape(temp.shape[0],-1)
    fr[item] = temp

fr = fr.reshape((fr.shape[0],1,fr.shape[1]*fr.shape[2]))

x = np.asarray((x,)*len(scann))
x = x.reshape((x.shape[0],1,x.shape[3]))
x = np.concatenate((x,fr), axis=2)
```

```
    X = (x - model_dict['mean'])/model_dict['std']

    y = model.predict(X.reshape((X.shape[0],X.shape[2])))

    if not name:
        name = formula.replace('$_{','').replace('}$','') + '_ann_prob'

    plt.figure()
    plt.title(r'Perovskite probability for ' + formula)
    plt.scatter(scanned_latpar, np.round(100*y[:,0],2),
                marker='o', s=75,  color = '#a60b20')

    plt.xlabel('lattice parameter')
    plt.ylabel('ANN output probability')
    plt.savefig(name + '.png')

    with open(name + '.csv', 'w') as f:

        f.write('lattice parameter (angstrom),perovskite probability (%)' + '\n' )

        for item in range(scanned_latpar.shape[0]):

            probability = 100*y[item,0]

            f.write("%.4f" % scanned_latpar[item] + ',')
            f.write("%.2f" % probability)
            f.write('\n')
        f.close()

    print('file saved for the compound', formula.replace('$_{','').replace('}$',''),
          'with the name', name)
    return y, scanned_latpar


def ctrl_dictionary(archivo='compounds2simulate'):
    """
    Parameters: A txt - file which has the compounds to simulate with the ANN.
    Return: A dictionary used afterwards with the function scan_nnoutputs.
    """

    f=list(filter(None,open(str(archivo)+'.txt','r').read().split('\n')))

    sg_ikeys=[f.index(sg) for sg in f if 'COMPOUND' in sg]+[len(f)]

    diccio={}
    for item in range(len(sg_ikeys)-1):
        text = f[sg_ikeys[item]:sg_ikeys[item+1]]
        key = [entry.split(':')[0] for entry in text]
        value = [entry.split(':')[1] for entry in text]
        diccio[item] = {k:v for k,v in zip(key,value)}

    dicciovar = {}
```

```
    for key in range(len(diccio.keys())):
        dicciovar[key] = dict()
        dicciovar[key]['name'] = diccio[key]['COMPOUND'].lstrip()
        dicciovar[key]['elements'] = list()
        dicciovar[key]['compositions'] = list()
        dicciovar[key]['maxdev'] = float(diccio[key]['maxdev'])
        dicciovar[key]['stepsize'] = float(diccio[key]['stepsize'])

        dicciovar[key]['elements'] += [[i.lstrip() for i in diccio[key]['cuboctahedron_atom'].split(',')]]
        dicciovar[key]['elements'] += [[i.lstrip() for i in diccio[key]['octahedron_atom'].split(',')]]
        dicciovar[key]['elements'] += [[i.lstrip() for i in diccio[key]['vertex_atom'].split(',')]]

        dicciovar[key]['compositions'] += [[float(i) for i in diccio[key]['cuboctahedron_frac'].split(',')]]
        dicciovar[key]['compositions'] += [[float(i) for i in diccio[key]['octahedron_frac'].split(',')]]
        dicciovar[key]['compositions'] += [[float(i) for i in diccio[key]['vertex_frac'].split(',')]]


    return dicciovar


archivo = input('Please provide the name of the text file containing ' +
                'the compounds to simulate [Default compounds2simulate]:' +
                '\n')
if not archivo:
    archivo = 'compounds2simulate'

filediccio = ctrl_dictionary(archivo=archivo)

for key in filediccio:
    scan_nnoutputs(elements = filediccio[key]['elements'],
                   compositions = filediccio[key]['compositions'],
                   maxdev = filediccio[key]['maxdev'],
                   stepsize = filediccio[key]['stepsize'],
                   name = filediccio[key]['name'])
```

APÉNDICE B: ARCHIVOS DE TEXTO

B.1 *structure_dictionary.txt*

spacegroup: 221

option: 0
A: b
B: a
X: d

spacegroup: 127

option: 0
A: c
B: a
X: b, g

spacegroup: 139

option: 0
A: a, b, c
B: f
X: h, n

spacegroup: 204

option: 0
A: a, b
B: c
X: g

spacegroup: 71

option: 0
A: a,b,c,d
B: k
X: l, m, n

spacegroup: 140

option: 0
A: b
B: c
X: a, h

spacegroup: 74

option: 0
A: e
B: b
X: e, g

spacegroup: 167

option: 0
A: a
B: b
X: e

spacegroup: 12

option: 0
A: i
B: e
X: i, g, h

option: 1
A: i
B1: a
B2: d
X: i, j

spacegroup: 15

option: 0
A: e
B: b
X: e, f

option: 1
A: e, e
B1: c
B2: d
X: f, f, f

spacegroup: 2

option: 0
A: i
B: a, b
X: i, i, i

option: 1
A: i, i, i, i
B1: a, e, f, g
B2: b, c, d, h
X: i, i, i, i, i, i

spacegroup: 63

option: 0
A: c, c
B: d
X: e, f, g

spacegroup: 62

option: 0
A: c
B: b
X: c, d

spacegroup: 11

option: 0
A: e, e
B: b, c
X: e, e, f, f

spacegroup: 137

option: 0
A: a, b, d
B: e
X: g, g, f

spacegroup: 59

option: 0
A: a, a, b, b
B: c, d
X: e, e, f, f, g

spacegroup: 48

option: 0
A: a, b, c, d
B1: e
B2: f
X: m, m, m

spacegroup: 201

option: 0
A: a, d
B1: b
B2: c
X: h

spacegroup: 13

option: 0
A: e, e, f, f
B1: a, b
B2: c, d
X: g, g, g, g

spacegroup: 86

option: 0
A : a, b, e
B1: c
B2: d
X: g, g, g

spacegroup: 14

option: 0
A : e
B1: c
B2: d
X: e, e, e

spacegroup: 148

option: 0
A: c
B1: a
B2: b
X: f

spacegroup: 134

option: 0
A: a, b, c
B1: e
B2: f
X: m, n

spacegroup: 128

option: 0
A: c
B1: a
B2: b
X: e, h

spacegroup: 87

option: 0
A: b
B1: a
B2: b

X: e, h

spacegroup: 225

option: 0

A: c
B1: a
B2: b
X: e

B.2 datosrahm.csv

```
Symbol,Z,elecPau,atradrahm
H,1,2.2,1.54
He,2,,1.34
Li,3,0.98,2.2
Be,4,1.57,2.19
B,5,2.04,2.05
C,6,2.55,1.9
N,7,3.04,1.79
O,8,3.44,1.71
F,9,3.98,1.63
Ne,10,,1.56
Na,11,0.93,2.25
Mg,12,1.31,2.4
Al,13,1.61,2.39
Si,14,1.9,2.32
P,15,2.19,2.23
S,16,2.58,2.14
Cl,17,3.16,2.06
Ar,18,,1.97
K,19,0.82,2.34
Ca,20,1,2.7
Sc,21,1.36,2.63
Ti,22,1.54,2.57
V,23,1.63,2.52
Cr,24,1.66,2.33
Mn,25,1.55,2.42
Fe,26,1.83,2.26
Co,27,1.88,2.22
Ni,28,1.91,2.19
Cu,29,1.9,2.17
Zn,30,1.65,2.22
Ga,31,1.81,2.33
Ge,32,2.01,2.34
As,33,2.18,2.31
Se,34,2.55,2.24
Br,35,2.96,2.19
Kr,36,,2.12
Rb,37,0.82,2.4
Sr,38,0.95,2.79
Y,39,1.22,2.74
Zr,40,1.33,2.68
Nb,41,1.6,2.51
Mo,42,2.16,2.44
Tc,43,2.1,2.41
Ru,44,2.2,2.37
Rh,45,2.28,2.33
Pd,46,2.2,2.15
Symbol,Z,elecPau,atradrahm
Ag,47,1.93,2.25
Cd,48,1.69,2.38
In,49,1.78,2.46
Sn,50,1.96,2.48
Sb,51,2.05,2.46
Te,52,2.1,2.42
I,53,2.66,2.38
Xe,54,2.6,2.32
Cs,55,0.79,2.49
Ba,56,0.89,2.93
La,57,1.1,2.84
Ce,58,1.12,2.82
Pr,59,1.13,2.86
Nd,60,1.14,2.84
Pm,61,1.13,2.83
Sm,62,1.17,2.8
Eu,63,1.2,2.8
Gd,64,1.2,2.77
Tb,65,1.1,2.76
Dy,66,1.22,2.75
Ho,67,1.23,2.73
Er,68,1.24,2.72
Tm,69,1.25,2.71
Yb,70,1.1,2.77
Lu,71,1.27,2.7
Hf,72,1.3,2.64
Ta,73,1.5,2.58
W,74,1.7,2.53
Re,75,1.9,2.49
Os,76,2.2,2.44
Ir,77,2.2,2.33
Pt,78,2.2,2.3
Au,79,2.4,2.26
Hg,80,1.9,2.29
Tl,81,1.8,2.42
Pb,82,1.8,2.49
Bi,83,1.9,2.5
Po,84,2,2.5
At,85,2.2,2.47
Rn,86,,2.43
Fr,87,0.7,2.58
Ra,88,0.9,2.92
Ac,89,1.1,2.93
Th,90,1.3,2.89
Pa,91,1.5,2.85
U,92,1.7,2.83
```

Symbol,Z,elecPau,atradrahm
Np,93,1.3,2.8
Pu,94,1.3,2.78
Am,95,,2.76
Cm,96,,2.76

APÉNDICE C: COMPUESTOS UTILIZADOS PARA EL DESARROLLO DE LAS REDES NEURONALES ARTIFICIALES

Consúltese la liga siguiente:
https://github.com/gomezperalta/phd_thesis/tree/master/TodosLosCompuestosUsados

APÉNDICE D: HISTOGRAMAS DE LOS DATOS DE ENTRADA

Consúltese la liga siguiente:
https://github.com/gomezperalta/phd_thesis/tree/master/HistogramasDatosDeEntrada

APÉNDICE E: MATRICES DE CORRELACIÓN E INFLUENCIA DE REMOVER DATOS DE ENTRADA DE LAS REDES NEURONALES DE SEIS Y OCHO SITIOS

Consúltese la liga siguiente:
https://github.com/gomezperalta/phd_thesis/tree/master/MatricesDeCorrelacion_InfluenciaDeLosRasgos

APÉNDICE F: COMPUESTOS PREDICHOS POR LA RED Y QUE CONVERGIERON EN CÁLCULOS DE ENERGÍA DE UN SOLO PUNTO

Consúltese la liga siguiente:
https://github.com/gomezperalta/phd_thesis/tree/master/CalculosEnergiaDe1Punto

En caso de tener algún problema con las ligas anteriores, favor de mandar un correo a gomezperalta.ai@gmail.com

Journal of Solid State Chemistry 285 (2020) 121253

Contents lists available at ScienceDirect

Journal of Solid State Chemistry

journal homepage: www.elsevier.com/locate/jssc

Discovering new perovskites with artificial intelligence

Juan I. Gómez – Peralta, Xim Bokhimi *

Laboratorio de Inteligencia Artificial del Instituto de Física, Universidad Nacional Autónoma de México, A. P. 20 – 364, 01000, Mexico

A B S T R A C T

An Artificial Neural Network (ANN) was developed to discover new inorganic perovskite – structures. The ANN assessed the probability to crystallize as a perovskite structure for compounds described with up to four Wyckoff sites. The ANN was also able to address the compounds independently of their crystal system. The input data needed by the ANN, also known as features, were based on the treatment of the atomic radii, electronegativity, and atom positions of the crystal compound. In this manner, the ANN was fed with information concerning the geometric and packing factors as well as the chemical environment of the atoms in the material. Quantum mechanical calculations were not required to obtain a feature for the ANN, but they were used to validate the predictions done by the ANN, such as $CsBeCl_3$.

1. Introduction

Perovskites are paradigmatic crystal structures [1] that have found application in areas such as piezoelectrics [2], high – temperature superconductivity [3], and recently in solar cells [4,5], to mention some. The perovskites structures are normally introduced in their most symmetric form, known as *aristotype*, which corresponds to a cubic cell (space group $Pm\bar{3}m$, No. 221). In the aristotype, there are three different ions according to the geometry of the site where they locate. The cations occupy dodecahedral and octahedral sites. The anions are in the vertex of octahedra. Octahedra are connected by only sharing the vertices. The dodecahedral sites locate in the voids left by the octahedral connection. However, there are also perovskites in other crystal systems due to a reduction of the crystal symmetry. The reduction is ascribed mainly to the rotation between the octahedra framework, but there are other reasons like the displacement of the cation within the octahedral cage or distortions of the octahedra. The latter fact is occasioned Jahn – Teller effect [6–8].

Regarding to the crystal nature of the perovskite compounds, it must be mentioned that the atomic distribution in any crystal compound is described by the translational repetition, in space, of a unit cell. Additionally, the distribution of the atoms within the unit cell can be represented by point symmetry groups derived from the space group. The point symmetry groups generate sites within the unit cell that are known as the Wyckoff sites. The Wyckoff sites define the positions within the unit cell where the atoms can occupy. Therefore, Wyckoff sites can be used to describe the atomic environments around each of the atoms in the compound. For example, the Wyckoff sites help to describe the distribution of the atomic neighbors around a specific atom, which determines the behavior of the surrounded atom.

In recent years, initiatives have emerged to accelerate the discovery of new materials [9,10]. The traditional way to discover a material was by trial and error, by proposing models to explain the crystalline structure of known materials. After guessing a specific compound using a model, some groups started to try the synthesis of the proposed material. Other groups chose to test the stability of the atomic distribution of the proposed material. The test is performed using quantum mechanical calculations ab initio. This kind of calculations, however, have the disadvantage that they take long times and high costs. An alternative to the above methods is the use of the techniques of artificial intelligence. These techniques take advantage of the available information in databases to learn and predict new materials.

Among the most popular and effective Artificial Intelligence (AI) algorithms are Artificial Neural Networks (ANN). They have been proven to be successful even in those tasks commonly thought as human – skilled. The most expensive cost with the ANN is during the training stage. In contrast, the time and the cost required to evaluate a sample with the trained ANN (learned) is sharply reduced compared to quantum mechanical calculations.

ANNs and other AI algorithms are also applied to explore the potential energy surface of materials [11,12]. In this case, the ANNs are trained using the information of molecular dynamics of the material [13,14], and of their thermodynamic and mechanical properties [15–18]. The ANN algorithm has also been used to interpret spectra of vibrational spectroscopy [19,20], and to predict the crystalline structure of new materials [21–23]. ANNs have also been applied in problems related to materials having a perovskite type crystalline structure. For example, for materials showing ferroelectricity, and for discovering new materials

* Corresponding author.
E-mail address: bokhimi@fisica.unam.mx (X. Bokhimi).

https://doi.org/10.1016/j.jssc.2020.121253
Received 4 August 2019; Received in revised form 9 January 2020; Accepted 7 February 2020
Available online 8 February 2020
0022-4596/

that have the perovskite structure [24–29].

In this work, we take advantage of the crystallographic information of the compounds to train the ANN in order to classify them with a crystalline structure related or not with the perovskite. The crystallographic information, including the Wyckoff sites, of the compound, was used to construct the feature space that will provide the input data for the ANN. As far as we know, this method to construct the features has not been reported up to now. The methodology developed in the present work permits to deal with the analysis of different crystalline systems.

2. Materials and methods

2.1. Creation of compound collection

A collection gathering perovskite (also known as *true* samples) and non – perovskite structures (or *false* samples) was constructed to train the ANNs. There were 2862 samples in the created collection, which had a 1:1 proportion of perovskite and non – perovskite samples. The samples were taken from Crystallography Open Database (COD) [30]. The information about the atoms present in the sample and their arrangement within a unit cell was taken from COD without any further modification. Information about the CIF number of the samples in the collection can be found in the Supplementary Material – A.

The papers of P. Woodward [6,7], on simple and double perovskites, were the guide to choose the compounds regarded as perovskites. The present work considers only the perovskite structures with three or four Wyckoff sites, since most of the perovskites found in the COD can have that number of point – symmetry sites. The compounds that have a perovskite structure described with three Wyckoff sites were cubic with the space group $Pm\bar{3}m$ or trigonal with the space group $R\bar{3}c$. Compounds described with a perovskite structure with four Wyckoff sites can have different symmetries. It can be cubic with the space groups $Fm\bar{3}m$ and $Im\bar{3}$, or tetragonal with the space groups $I4/mcm$ and $P4/\ mbm$; or orthorhombic with the space groups Pnma and Imma. It can also be trigonal with the space group $R\bar{3}$, or monoclinic with the space group $C2/c$.

All compounds that had the perovskite structure contained at most 40 atoms in the unit cell. These compounds were stochiometric or solid solutions. They were oxides, nitrides, halides, chalcogenides, anti–perovskites, or bronzes. The crystalline structure of the compounds that did not have the perovskite structure contained three or four Wyckoff sites. In these compounds, there were no restrictions on the number of elements in their formula or on the number of atoms in their unit cell.

2.2. ANNs and input data

Fig. 1 shows a sketch of the ANN. From left to right, the first is the input layer; the last one is the output layer. The layers in between are the hidden layers. The number of nodes in the input layer corresponds to the number of features that describe a compound. In the present work, this number was 33 (Table 1). The ANN is full feed-forward connected. Each node of a layer connects to all the nodes of the next layer. Each connection is named weight and its value changes during the training of the ANN.

All nodes in the input layer connect to each node of the first hidden layer. Each connection is named as weight because it gives the weighted contribution of the associated input node to the node in the hidden layer. For each node in the hidden layer, the received weights form a sum that is transformed with an activation function to generate the node value.

Then, the nodes of the first hidden layer connect to the nodes of the second hidden layer. This connection follows the same rules described for the connection between the input layer and the first hidden layer. The final connection is between the last hidden layer and the output layer. With the output value of the only node in the hidden layer, the classification as perovskite (1) or non-perovskite (0) is done after thresholding.

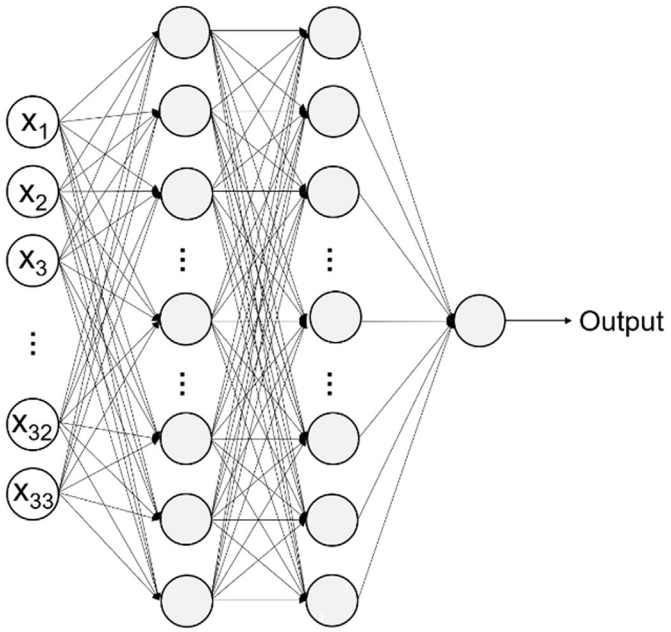


Fig. 1. Sketch of the best ANN. Our best ANN had two hidden layers both with 132 nodes and a single node in the output layer. The ANN was full – connected and feed – forward.

In this work, the threshold was set to 0.5. If the output value surpasses that threshold, the compound is predicted to have a perovskite structure; otherwise, it has a non - perovskite structure.

Since ANNs need to have the same input data dimension for all compounds, that is, the same number of features, an empty site was added to those described by three Wyckoff sites. The order of the sites increased with multiplicity (Supplementary Material – B).

2.3. Construction of the features

To each atom was assigned an average atomic radius [31] (r_i), as well as an average electronegativity [32], (χ_i). This assignment required to know the occupation fraction of the different atoms in the same Wyckoff site. When the site had vacancies, the occupation fraction was not normalized.

Table 1 gives the features provided to the ANN for classifying the compounds having a crystalline structure related or not to the perovskite. The ratios of atomic radii (features 1–6) describe the geometric factors of the crystal. The quotients of the sum of the pair of atomic radii (features 7–21) are related to the packing factors. They substituted the Goldschmidt tolerance factor. Local environment functions (features 22–33) describe the chemical surroundings of a Wyckoff site. The subscripts refer to either one or a pair of sites.

The number of combinations of pairs of the four average atomic radii determined the six ratios of atomic radii. Similarly, the six possible sums of two atomic radii determined the fifteen quotients of these sums. When the denominator of the quotient was zero, it was set to zero.

The local environment function of an atom in an i – site surrounded by atoms in j – sites is defined as follows:

$$f_{ij} = \begin{cases} d_{ij[n]} \le R_c,\ (\chi_i - \chi_j) \sum_{n=1} \left[\frac{1}{2} \left(\cos \frac{\pi d_{ij_{[n]}}}{R_c} + 1 \right) \right] exp \left[- \left(\frac{d_{ij_{[n]}}}{r_i^{norm} + r_j^{norm}} \right)^2 \right] \\ d_{ij[n]} > R_c,\ 0 \end{cases} \quad \text{(eq. 1)}$$

In the successive, that function is just named as local function. Equation (1) defines the local function of an atom at an i – site surrounded by atoms at j – sites, for $i \neq j$. When one of the sites is

Table 1
List of all the 33 features used to train the ANNs. These features are enlisted as x_n. Features related to geometric factors are x_1 to x_6, which refer to atomic radii pair quotients, and x_7 to x_{21}, which refer to atomic radii pair sum – quotients. Features x_{22} to x_{33} correspond to a local function, f_{ij}, which described the chemical environment of atoms in j – site around an atom in a i – site. The average atomic radii in an i – site is represented as r_i. Local function is described in equation (1).

Atomic radii pair quotients					
$x_1 : \frac{r_1}{r_4} x_2 : \frac{r_2}{r_4}$	$x_3 : \frac{r_3}{r_4} x_4 : \frac{r_2}{r_3}$			$x_5 : \frac{r_2}{r_4} x_6 : \frac{r_3}{r_4}$	
Atomic radii pair sum – quotients					
$x_7 : \frac{r_1+r_2}{r_1+r_3} x_8 : \frac{r_1+r_2}{r_1+r_4} x_9 : \frac{r_1+r_2}{r_2+r_3}$	$x_{10} : \frac{r_1+r_2}{r_2+r_4} x_{11} : \frac{r_1+r_2}{r_3+r_4} x_{12} : \frac{r_1+r_3}{r_1+r_4}$	$x_{13} : \frac{r_1+r_3}{r_2+r_3} x_{14} : \frac{r_1+r_3}{r_2+r_4} x_{15} : \frac{r_1+r_3}{r_3+r_4}$	$x_{16} : \frac{r_1+r_4}{r_2+r_3} x_{17} : \frac{r_1+r_4}{r_2+r_4} x_{18} : \frac{r_1+r_4}{r_3+r_4}$	$x_{19} : \frac{r_2+r_3}{r_2+r_4} x_{20} : \frac{r_2+r_3}{r_3+r_4} x_{21} : \frac{r_2+r_4}{r_3+r_4}$
Local functions				
$x_{22} : f_{12} x_{23} : f_{13} x_{24} : f_{14}$	$x_{25} : f_{21} x_{26} : f_{23} x_{27} : f_{24}$		$x_{28} : f_{31} x_{29} : f_{32} x_{30} : f_{34}$	$x_{31} : f_{41} x_{32} : f_{42} x_{33} : f_{43}$

unoccupied, the f_{ij} value is zero. This function is not zero when the atoms at *i* – and *j* – sites are different. Therefore, the features f_{11}, f_{22}, f_{33} and f_{44} were not computed.

In the local function equation, χ and r^{norm} refer to the electronegativity of the atoms in a site and to the atomic radii present in the involved Wyckoff sites, respectively. All distances between the central atom in the *i* – site and each neighbour in the *j* – site, $d_{ij[n]}$, were considered within a cut – off radius, R_c, which was equalled to 2.5 nm. This local function is based on the published work of J. Behler and M. Parrinello [11].

2.4. ANN – training

The 2862 compounds were split in a 70:15:15 proportion to create the training, cross-validation and test sets, respectively. The parameters of the ANN were optimized by using batches of 16 training compounds. Once all compounds of the training set were used, it is said that an epoch is completed. After that, the compounds of the cross-validation set were used to try the model without updating parameters. The process was repeated with up to 800 epochs.

Different ANNs were analyzed, which had different nodes in the hidden layers. All of these ANNs were feed-forward and full-connected. The initialization of the ANN parameters was performed with random values between [-1,1] for weights and zeros for biases. The technique of Dropout regularization [33] was implemented along with Adam optimization [34,35] to avoid overfitting. All features were scaled by standardization to avoid the dominance of one or some of them. Furthermore, feature scaling accelerates parameter optimization. All the activation functions were of type S, where hyperbolic tangents were used for the nodes in hidden layers and the sigmoid function for the output layer.

After training and cross-validation of the ANNs, they were tested with the compounds from the test set. Besides, there were 14339 compounds of the non-perovskite type remaining from COD. They were used for a second test. This manuscript only reports the best ANN, which obtained the highest scores and had the lowest value in the error function. The best ANN had two hidden layers, each with 132 nodes (Fig. 1).

The computed scores used to determine the best ANN correspond to the metric of Precision. Precision measures the ratio of successful prediction for a compound type, say perovskite or non - perovskite structure. These scores were determined with all the compounds of the training and cross - validation sets, by one side, and the test set, by the other side.

ANNs were built using the Python libraries Theano [36] and Keras [37]. The extraction of crystallographic information for each compound of COD was performed with Pymatgen [38]. All additional calculations were performed with the Python numpy, pandas and scikit-learn libraries [39].

2.5. Validation of the predicted new perovskites with quantum chemical calculations

The use of the ANN demands to have all the features of a compound to assess its probability to crystallize as perovskite structure. The computation of the features depends not only on the composition but also on the distribution of the atoms within the unit cell. Then, it becomes necessary to propose a space group with particular lattice parameters and the right occupation of symmetry sites. Fig. 2 summarizes these ideas to use the ANN to evaluate a new compound.

Ternary halides of transition and main - group metals were evaluated with the ANN to predict new perovskite materials. The halides were considered to crystallize in the aristotype form. The lattice parameter for the halides was proposed as follows. First, the sum of the atomic radii of the atom in the octahedral site and the halide was calculated. After that, the lattice parameter was considered to be that sum modified in −10, −5, 0, 5 and 10%. The reason behind this methodology to propose a lattice parameter is explained in the next section.

The halides proposals were evaluated with the ANN. Those proposals predicted as perovskite structures were later validated with quantum

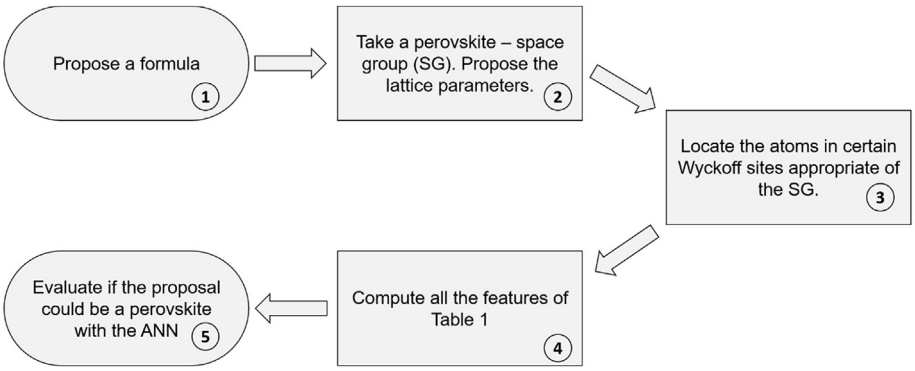


Fig. 2. Guideline to use the ANN. A set of atoms (elements) must be defined (step 1). After selecting a perovskite space – group [6,7], the lattice parameters have to be proposed (step 2) with the location of those atoms (step 3) in the corresponding Wyckoff sites. This location admits to share a Wyckoff site between different atom type or vacancies. Afterwards, the features have to be computed (step 4) to finally feed the ANN (step 5).

mechanical calculations in two steps.

First, a cluster of the predicted compound was built for a single-point molecular calculation. The cluster corresponded to a 2 × 2 × 2 super-structure of the unit-cell. There were 71 atoms in the cluster (Fig. 3, left): eight were in octahedral sites, 27 were in dodecahedral sites, and the remaining corresponded to the halide. The calculations were implemented with the functional PBE0 and the basis-set lanl2dz with effective-core potential (ECP). The single-point calculations were performed using TeraChem [40] software package.

The predicted compounds converged in the single-point calculation were afterwards validated with Periodic Boundary Conditions (PBC) calculations. Among the predictions with the same composition, it was chosen the one with the lowest energy in the single-point calculation. For the second calculations, the aforementioned cluster without the atoms of three faces was used due to PBC. Right-side of Fig. 3 sketches the modeled structure. The calculations were implemented with the basis-set DZVP – MOLOPT – GTH and the functional PBE, with the cp2k [41] software package. The Supplementary Material – C details the element pseudopotential for the PBC calculations.

This article shows the top - five prediction of each halide-type that scored higher with the ANN. These predictions converged in both quantum chemical calculations.

2.6. Lattice parameters of the predicted compounds

There is a well-known relationship between the lattice parameter of an aristotype perovskite and the sum of the atomic radii. That sum involves the radii of octahedral cation and the anion. Since atomic radii were used instead of the ionic ones in this work, that relationship was revisited with the here worked radii.

290 and 225 perovskite compounds of the cubic space groups 221 and 225, respectively, were considered. They had all their Wyckoff sites occupied with only one element and without vacancies. The fit between the experimental lattice parameter with the mentioned sum was $a_{exp} = 1.0378a_{pred} - 0.1680$, with $R^2 = 0.9401$. The RMSE (Root Mean Square Error) and MAE (Mean Absolute Error) values were 0.5244 and 0.3896, respectively, whereas mean absolute percent error to the lattice parameter was 6.3182 ± 4.8292%. The graph of the fit is in the Supplementary Material - D. 86.38% of the 492 compounds used in the fit fell within a range between 0 and 10% of deviation in the atomic radii sum, and therefore, the lattice parameters were varied within that range.

3. Results

Table 2 shows the precisions obtained with the best ANN using all the compounds regardless of the number of original sites. With all the

Table 2
Precision in the prediction as perovskite and non – perovskite structures. The scores were calculated with all the compound in the training/CV and test sets. In the former set there were 2434 compounds, whereas in the last one there were 428 samples.

Set	Precision with perovskite structures (%)	Precision with non-perovskite structures (%)
Training/CV	98.18	97.52
Test	96.67	94.95

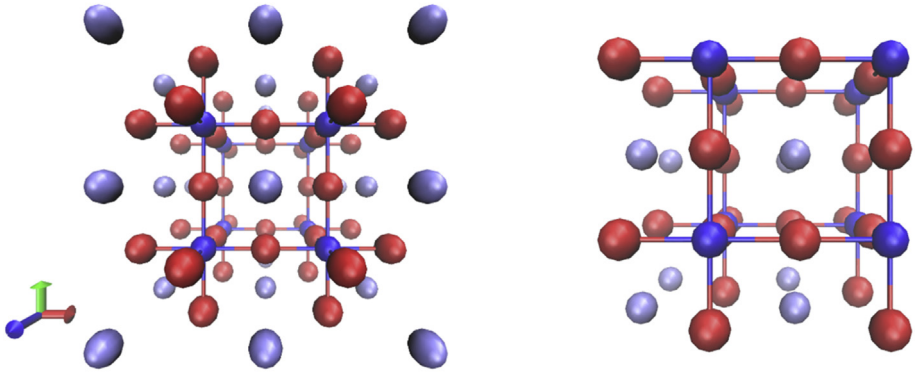


Fig. 3. Examples of the structures used for single – point (left) and the cell optimization (right) calculations. The first structure was run in TeraChem, whereas the second one was calculated in cp2k with PBC. Images displayed with VMD [42].

samples of the training and cross-validation sets, it was obtained an average of 97.85%. The average obtained with the compounds of the test set was 95.80%. This difference between the mean scores is an indicator of little overfitting, i.e., the model was able to infer the crystal structure of a new compound.

The aforementioned results were also analyzed based on the number of initially Wyckoff sites in the compounds. Tables 3 and 4 contain the precisions for the compounds with three and four Wyckoff sites. The determined values for the compounds in Table 3 were close to the ones condensed in Table 2. Furthermore, the overfitting is essentially negligible between the sets of training and cross-validation and the test set. That did not happen in Table 4. The ANN classified better the perovskites structures than the non – perovskites structures in the test set than in the training and cross – validation sets. This last result could be ascribed to a difference in the distribution between the compounds in the sets of training and cross-validation and the test set. Furthermore, slight overfitting in non – perovskite structures was observed. A way to homogenize these results and diminish overfitting is to increase the number of compounds of the collection.

In the second and larger test, all non – perovskites structures (14339 compounds) remaining from COD were used to evaluate the performance of the trained ANNs. The results with this second test are shown in Table 5. These scores correspond essentially to the metric Recall. In this second test, we included compounds originally described with one and two Wyckoff sites even though we did not use this type of compounds during the training of the ANNs. Evaluation of this type of compound is straightforward for a skilled experimentalist since perovskite structures need at least three Wyckoff sites to be described. It is expected that an ANN should be able to answer those trivial questions. This purpose was only achieved with the non – perovskite structures described with one Wyckoff site. The ANN erred in 1.09% of all 4307 crystal compounds with initially two Wyckoff sites. Finally, the results with the compounds of three and four originally Wyckoff sites were similar to the precision values shown in Tables 3 and 4

Table 3
Precisions in the prediction as perovskite and non – perovskite structures for the compounds with initially three Wyckoff sites. In the whole training/CV sets there were 929 compounds, whereas in the test set there were 176 compounds.

Set	Precision with perovskite structures (%)	Precision with non-perovskite structures (%)
Training/CV	97.82	98.09
Test	97.92	98.75

Table 4
Precision in the prediction as perovskite and non – perovskite structures for the compounds with initially four Wyckoff sites. In the whole training/CV sets there were 1505 compounds, whereas in the test set there were 252 compounds. Metrics are differentiated according to the structure type.

Set	Precision with perovskite structures (%)	Precision with non-perovskite structures (%)
Training/CV	98.40	97.22
Test	95.61	92.75

Table 5
Results after testing the best ANN with all remaining non – perovskites in COD. The trained model allowed us to manage compounds with up to four Wyckoff sites. The accuracy of the classification according to the number of Wyckoff sites in the non – perovskite compounds are shown by columns.

Compounds with 1 occupied site (1189)	Compounds with 2 occupied sites (4307)	Compounds with 3 occupied sites (5442)	Compounds with 4 occupied sites (3401)
100.00%	98.91%	96.31%	96.09%

The probability predicted by the ANN for all the compounds used among this work can be consulted in the Supplementary Material – A.

The list of the predicted halide perovskite is shown in Table 6. All these compounds have not been synthesized yet, but $SrLiF_3$, $TlSrF_3$, $ZnScF_3$, $RbPdCl_3$, $ZnLiF_3$, $TlBeF_3$ and $CsBeCl_3$ were already reported in previous theoretical works [27,43–45] with other approaches.

The first column of Table 6 shows the formula of the predicted. The first two elements of each formula were proposed to occupy the dodecahedral and octahedral sites, respectively. In most PBC calculations, cell optimization only changed the value of the lattice parameter. The lattice parameters are referred to one unit – cell for most cases. The exceptions were with $KNiI_3$, $KIrI_3$ and $RbIrBr_3$. In those structures, the atoms moved due to distortion within the octahedral framework, and therefore, the lattice parameter doubled. For those cases, the unit cell contained eight unit-formulas. Despite the distortion, the cubic symmetry was preserved in the supercell.

The optimized positions of the distorted perovskites can be found in the Supplementary Material – E. For the non – distorted structures of Table 6, the atom positions corresponded to the concerning ones of the aristotype.

4. Discussion

All constructed features to train the ANN were based on already reported data about the atomic radii and the electronegativity. Furthermore, the feature construction did not require any quantum chemical computation to obtain a feature. Therefore, we expect that users of the present methodology do not find a black box behind the computation of any needed feature for the ANN. The features had a base on empirical concepts such as geometric factors, packing factors, and chemical environment around the atoms, which is defined by one Wyckoff symmetry site. These concepts are well known for a long time for skilled experimentalists and have been useful to estimate the feasibility of a set of elements to crystallize as a perovskite.

Since perovskites are typically considered as ionic compounds, it is expected the use of ionic radii instead of the atomic ones. It is well known

Table 6
Possible new cubic halide perovskites predicted with the ANN and validated with PBC calculations. For each formula in the first column, the dodecahedral and octahedral sites are occupied for the first and the second element, respectively. PBC calculations were done by using a $2 \times 2 \times 2$ supercell (40 atoms). The ANN – output value for each new structure is in column 2. The initial and optimized lattice parameters (in Å) are in columns 3 and 4. Those structures marked with * had distortions in their framework due to rotations between octahedra and their lattice parameter are referred to the $2 \times 2 \times 2$ supercell (eight formula-units within instead of one).

Predicted compound	Probability predicted	Proposed lattice parameter (Å)	Optimized lattice parameter (Å)
$SrLiF_3$ [43]	0.9992	3.830	3.803
$TlSrF_3$ [27]	0.9962	4.641	4.724
$ZnScF_3$ [27]	0.9920	4.047	4.159
$CaCuI_3$	0.9903	5.005	5.395
$RbPdBr_3$ [42]	0.9897	4.774	5.167
$ZnLiF_3$ [27]	0.9881	3.447	3.571
$CaAgI_3$	0.9863	5.093	5.619
$BaAgBr_3$	0.9858	4.884	5.379
$CaCuBr_3$	0.9848	4.796	5.034
$BaCuCl_3$	0.9841	4.653	4.861
$KNiI_3$ *	0.9839	10.054	10.600
$KIrI_3$ *	0.9834	10.362	10.678
$RbRhBr_3$	0.9829	4.972	5.162
$CsTcI_3$	0.9823	5.264	5.616
$BaAgCl_3$	0.9815	4.741	5.122
$RbIrBr_3$ *	0.9807	9.948	10.138
$CsRhCl_3$	0.9765	4.829	4.905
$CsRuCl_3$	0.9751	4.873	4.883
$TlBeF_3$ [27]	0.9693	3.820	3.867
$CsBeCl_3$ [43]	0.9661	4.675	4.795

that ionic radii depend on the oxidation state as well as on the coordination number. The determination of oxidation states can be as straightforward as knowing all charges of the elements in the formula except one or as laborious as using codes based on the valence – bond method [44] or even quantum chemical calculations. By the other side, the oxidation state assessment may even become tricky when there are two elements with several possible charges in the formula, like with transition metals. Additionally, the oxidation states also depend on the coordination number, which is easily determined for the most symmetrical structures. Due to the mentioned reasons, the use of atomic radii was for simplicity sake.

Even though ionic radii were not considered, the performance of the ANN was above 95%. We may suggest that this high score is mainly due to local functions, which modeled the surrounding chemical environment of the atoms. The local functions took into account the electronegativity difference between the atoms of the two concerning sites and also were sensible to the distance between the involved pair of atoms. Exploration of other functional forms of the local function is left as material for other studies, as well as the use of ionic radii instead of those here worked.

The ANN was trained even to manage distorted structures, though the focus was in modeling new compounds with the aristotype. That was possible because, in some sense, the ANN was not only fed with information about the composition of the compound, but it also obtained an idea about the distribution of atoms in space. The atom distribution was implicitly present due to the feature construction based on Wyckoff sites. We consider that the inclusion of the atom arrangement helps to narrow the efforts for finding new perovskite structures.

5. Conclusions

The ANN trained with features solely based on empirical data led to scores above 96% on the classification of crystal compounds as perovskite or non – perovskite structures. The ANN allowed not only to deal with the high symmetry, cubic perovskite structures but also with those where there is rotation between octahedra. The compounds $SrLiF_3$, $TlSrF_3$, $ZnScF_3$, $CaCuI_3$, $RbPdBr_3$, $ZnLiF_3$, $CaAgI_3$, $BaAgBr_3$, $CaCuBr_3$, $BaCuCl_3$, $KNiI_3$, $KIrI_3$, $RbRhBr_3$, $CsTcI_3$, $BaAgCl_3$, $RbIrBr_3$, $CsRhCl_3$, $CsRuCl_3$, $TlBeF_3$ and $CsBeCl_3$ were predicted by the ANN as new materials with perovskite structure. Calculations with Periodic Boundary Conditions gave support to the predictions.

The scopes of this investigation suggest that the present methodology can be applied to other crystal systems with more than four sites. This approach may serve to develop further binary classification models for other structure types such as garnets, spinels or wutzites. The success of the present job also set a precedent to train ANNs to classify a compound into a manifold of crystal - structures.

CRediT authorship contribution statement

Juan I. Gómez – Peralta: Conceptualization, Methodology, Software, Validation, Writing - original draft, Visualization, Formal analysis. **Xim Bokhimi:** Conceptualization, Resources, Writing - review & editing, Project administration, Formal analysis, Funding acquisition.

Acknowledgements

Juan I. Gómez – Peralta thanks CONACyT, for the doctoral scholarship No. 336003 (CVU: 620161), Artificial Intelligence Lab of the Institute of Physics (LIA – IFUNAM), for the given support.

Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.jssc.2020.121253.

References

[1] A.R. Chakmouradian, P.M. Woodward, Phys. Chem. Miner. 41 (2014) 387–391.
[2] K. Uchino, Sci. Technol. Adv. Mater. 16 (2015) 4, 046001.
[3] C.A. Hancock, J.M. Porras – Vazquez, P.J. Keenan, P.R. Slater, Dalton Trans. 44 (2015) 10559–10569.
[4] T.M. Koh, K. Thirumal, H.S. Soo, N. Mathews, Chem. Sus. Chem. 9 (2016) 2541–2558.
[5] J. Liang, J. Liu, Z. Jin, Sol. RRL 1 (2017) 1700086.
[6] P.M. Woodward, Acta Crystallogr. B53 (1997) 32–43.
[7] P.M. Woodward, Acta Crystallogr. B53 (1997) 44–46.
[8] C.J. Howard, H.T. Stokes, Acta Crystallogr. B54 (1998) 782–789.
[9] Materials Genome Initiative, Strategic Plan, National Science Technology Council, USA, 2014. https://obamawhitehouse.archives.gov/mgi (accessed January, 2019).
[10] Report of the Clean Energy Materials Innovation Challenge Expert Workshop, Mission Innovation, 2018. bit.ly/2rn09Vj (accessed January, 2019).
[11] J. Behler, M. Parrinello, PRL 98 (2007) 146401.
[12] J.S. Smith, O. Isayev, A.E. Roitberg, Chem. Sci. 8 (2017) 3192–3203.
[13] F. Häse, S. Valleau, E. Pyzer – Knapp, A. Aspuru – Guzik, Chem. Sci. 7 (2016) 5139–5147.
[14] F. Häse, C. Kreisbeck, A. Aspuru – Guzik, Chem. Sci. 8 (2017) 8419–8426.
[15] A.V. Fedorov, I.V. Shamanaev, Mol. Inf. 36 (2017) 1600162.
[16] T. Xie, J.C. Grossman, PRL 120 (2018) 145301.
[17] K. Hansen, G. Montavon, F. Biegler, S. Fazli, M. Rupp, M. Scheffler, O.A. von Lilienfield, A. Tkatchenko, K.R. Müller, J. Chem. Theor. Comput. 9 (2013) 3404–3419.
[18] F. Faber, A. Lindmaa, O.A. von Lilienfeld, R. Armiento, Int. J. Quant. Chem. 115 (2015) 1094–1101.
[19] J. Liu, M. Osadchuy, L. Ashton, M. Foster, C.J. Solomon, S.J. Gibson, Analyst 142 (2017) 4067–4074.
[20] M. Gastegger, J. Behler, P. Marquetand, Chem. Sci. 8 (2017) 6924–6935.
[21] A. Ziletti, D. Kumar, M. Scheffler, L.M. Ghiringhelli, Nat. Commun. 9 (2018) 2775.
[22] W.B. Park, J. Chung, J. Jung, K. Sohn, S.P. Singh, M. Pyo, N. Shin, K.S. Sohn, IUCrJ 4 (2017) 486–494.
[23] W.F. Reinhart, A.W. Long, M.P. Howard, A.L. Ferguson, A.Z. Panagiotopoulos, Soft Matter 13 (2017) 4733–4735.
[24] O. Allam, C. Holmes, Z. Greenberg, K.C. Kim, S.S. Jang, ChemPhysChem 19 (2018) 2559–2565.
[25] P.V. Balachandran, B. Kowalski, A. Sehirlioglu, T. Lookman, Nat. Commun. 9 (2018) 1668.
[26] G. Pilania, P.V. Balachandran, C. Kim, T. Lookman, Front. Mater. 3 (2016) 19.
[27] A. van Roekenghem, J. Carrete, C. Oses, S. Curtarolo, N. Mingo, Phys. Rev. X 6 (2016), 041061.
[28] Q. Xu, Z. Li, M. Liu, W.J. Yin, J. Phys. Chem. Lett. 9 (2018) 6948–6954.
[29] J. Im, S. Lee, T.W. Ko, H.W. Kim, Y. Hyon, H. Chang, npj Computational Materials 5 (2019) 37.
[30] a) S. Gražulis, D. Chateigner, R.T. Downs, A.T. Yokochi, M. Quiros, L. Lutterotti, E. Manakova, J. Butkus, P. Moeck, A. Le Bail, J. Appl. Crystallogr. 42 (2009) 726–729;
b) S. Gražulis, A. Daškevič, A. Merkys, D. Chateigner, L. Lutterotti, M. Quirós, N.R. Serebryanaya, P. Moeck, R.T. Downs, A. LeBail, Nucleic Acids Res. 40 (2012) D420–D427.
[31] M. Rahm, R. Hoffmann, N.W. Ashcroft, Chem. Eur J. 22 (2016) 14625–14632.
[32] W.M. Haynes, CRC Handbook of Chemistry and Physics. 100 Key Points, 95th edition, CRC Press, London, 2014.
[33] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, J. Mach. Learn. Res. 15 (2014) 1929–1958.
[34] S. Ruder, arXiV: 1609.04747
[35] D.P. Kingma, J.L. Ba, in: Proceedings of the 3rd International Conference on Learning Representations (ICLR) 2014, 2014 arXiV: 1412.6980.
[36] The Theano Development Team, arXiV: 1605.02688, 2016
[37] François Chollet. https://keras.io (accessed January, 2019).
[38] S.P. Ong, W.D. Richards, A. Jain, G. Hautier, M. Kocher, S. Cholia, D. Gunter, V.L. Chevrier, K.A. Persson, G. Ceder, Comp. Mat. Science 68 (2013) 314–319.
[39] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, J. Mach. Learn. Res. 12 (2011) 2825–2830.
[40] a) I.S. Ufimtsev, T.J. Martínez, J. Chem. Theor. Comput. 5 (2009) 2619;
b) A.V. Titov, I.S. Ufimtsev, N. Luehr, T.J. Martínez, J. Chem. Theor. Comput. 99 (2013) 213;
c) C. Song, L.–P. Wang, T.J. Martínez, J. Chem. Theor. Comput. 12 (2016) 92;
d) J. Kästner, J.M. Carr, T.W. Keal, W. Thiel, A. Wander, P. Sherwood, J. Phys. Chem. 113 (2009) 11856.
[41] a) The CP2K Developers Group. https://www.cp2k.org/ (Accessed: May 24th, 2019);
b) J. Hutter, M. Ianuzzi, F. Schiffmann, J. VandeVondele, WIRE – Computational Molecular Science 4 (2014) 15–25.
[42] W. Humphrey, A. Dalke, K. Schulten, J. Mol. Graph. 14 (1996) 33–38.
[43] S. Körbel, M.A.L. Marques, S. Botti, J. Mater. Chem. C. 4 (2016) 3157–3167.
[44] M.A. Costales, Castro Topología de la densidad electrónica en cristales. Una teoría cuántica del enlace cristalino, PhD. Thesis, Universidad de Oviedo, 1998.
[45] I.D. Brown, K.R. Poeppelmeier, Bond Valences. *Structure and Bonding* 158, Springer – Verlag Berlin Heidelberg, 2014.