



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

ESCUELA NACIONAL DE ESTUDIOS SUPERIORES
UNIDAD MORELIA

DETECCIÓN DE SISMOS UTILIZANDO
APRENDIZAJE PROFUNDO

T E S I S

QUE PARA OBTENER EL TÍTULO DE

LICENCIADO EN TECNOLOGÍAS PARA
LA INFORMACIÓN EN CIENCIAS

P R E S E N T A

FERNANDO RODRIGO AGUILAR JAVIER

TUTOR

DR. SERGIO ROGELIO TINOCO MARTÍNEZ

CO-TUTOR

DR. LUIS ANTONIO DOMÍNGUEZ RAMÍREZ



ESCUELA
NACIONAL
de ESTUDIOS
SUPERIORES
mm
UNIDAD MORELIA

MORELIA, MICHOACÁN SEPTIEMBRE 2021



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
ESCUELA NACIONAL DE ESTUDIOS SUPERIORES, UNIDAD MORELIA
SECRETARÍA GENERAL
SERVICIOS ESCOLARES

MTRA. IVONNE RAMÍREZ WENCE
 DIRECTORA
 DIRECCIÓN GENERAL DE ADMINISTRACIÓN ESCOLAR
P R E S E N T E

Por medio de la presente me permito informar a usted que en la **sesión ordinaria 06** del **H. Consejo Técnico** de la Escuela Nacional de Estudios Superiores (ENES) Unidad Morelia celebrada el día **22 de junio del 2021**, acordó poner a su consideración el siguiente jurado para la presentación del Trabajo Profesional del alumno (a) **Fernando Rodrigo Aguilar Javier** adscrito a la Licenciatura en Tecnologías para la Información en Ciencias con número de cuenta **416064890**, quien presenta la tesis titulada: "Detección de sismos utilizando aprendizaje profundo" bajo la dirección como **tutor** del Dr. Sergio Rogelio Tinoco Martínez y como **co-tutor** el Dr. Luis Antonio Domínguez Ramírez.

El jurado queda integrado de la siguiente manera:

Presidente: Dr. Víctor Hugo de la Luz Rodríguez
Vocal: Dr. Luis Miguel García Velázquez
Secretario: Dr. Sergio Rogelio Tinoco Martínez
Suplente 1: Dr. Teodoro Carlón Allende
Suplente 2: Dra. Adriana Menchaca Méndez

Sin otro particular, quedo de usted.

A t e n t a m e n t e
 "POR MI RAZA HABLARA EL ESPIRITU"
 Morelia, Michoacán a 29 de junio del 2021.

DRA. YESENIA ARREDONDO LEÓN
SECRETARIA GENERAL

CAMPUS MORELIA

Antigua Carretera a Pátzcuaro N° 8701, Col. Ex Hacienda de San José de la Huerta
 58190, Morelia, Michoacán, México. Tel: (443)689.3500 y (55)56.23.73.00, Extensión Red UNAM: 80614
www.enesmorelia.unam.mx

Agradecimientos institucionales

La agradezco a la Escuela Nacional de Estudios Superiores Unidad Morelia, a toda la UNAM campus Morelia y a la Universidad Nacional Autónoma de México, por haberme aceptado para formar parte de esta increíble institución, por haberme dado los conocimientos y habilidades necesarias en el ámbito no solo profesional, sino también ético, académico y laboral.

A todo el cuerpo docente de la licenciatura en Tecnologías para la Información en Ciencias, mi más sincero agradecimiento, en especial a los siguientes académicos: Dr. Víctor Hugo de la Luz Rodríguez, Dra. Adriana Menchaca Méndez, Dr. Luis Miguel García Velázquez y, de la licenciatura en Geociencias, al Dr. Luis Antonio Domínguez Ramírez; por haber influido de manera positiva para no decaer en el proceso y por toda la ayuda durante mi formación y paso por la universidad.

Le agradezco también a todos los académicos que tan amables aceptaron participar en la mesa sinodal.

Agradezco a los siguientes proyectos que me ayudaron económicamente durante el desarrollo de esta tesis: al proyecto **PAPIIT TA101920** y al proyecto **PAPIME PE103018**, ya que sin este apoyo no hubiera podido concluir este trabajo.

Agradezco al Dr. Sergio Tinoco por su amistad antes, durante y después de este proyecto, por haber fungido como mi asesor y por sus observaciones.

Al Dr. Luis Antonio por haber fungido como co-asesor, por su tiempo y empeño en este trabajo.

Esta tesis está dedicada a dos personas que amo con todo el corazón. Una de ellas es mi Madre, que siempre me dió todo para salir adelante y nunca se rindió por darme la oportunidad de continuar estudiando y cumplir mis sueños. La segunda persona a la que esta dedicada esta tesis es Pamela Ortiz (Sarita) quien es el amor de mi vida, siempre tan encantadora y hermosa, con quien compartí los mejores momentos y a quien siempre llevare en mí corazón ♡.

Agradecimientos personales

Le agradezco a mis padres, mi familia y amigos que estuvieron siempre a mi lado, me dieron ánimos y apoyo incondicional. En especial a Sofía Hortensia Valencia Torres con quien compartí momentos maravillosos durante este proceso y con quien contaré y llevaré siempre con cariño. Le agradezco a mi prima, Juanita Hernández quien siempre me acompañó en mi camino académico y me animó en los momentos más difíciles. También un agradecimiento a mis mejores amigos: Javier Alan Salas Madrigal y a Ricardo Vieyra Magaña.

Va un agradecimiento especial a mi tutor de tesis, el Dr. Sergio Tinoco, por su amistad. Le agradezco a todos y cada uno de los académicos que influyeron en mí de manera positiva para no decaer en el proceso y que me ayudaron durante mi formación y paso por la universidad, en especial a las siguientes personas: Dr. Víctor Hugo de la Luz Rodríguez, Dra. Adriana Menchaca Méndez, Dr. Luis Miguel García Velázquez y al Dr. Luis Antonio Domínguez Ramírez.

Resumen

En los últimos años, la capacidad de monitoreo del Servicio Sismológico Nacional (SSN), así como la de diversas redes en el mundo, ha crecido de forma exponencial debido al aumento en el número de estaciones y a la facilidad de almacenar datos de forma digital. Esto ha permitido, por una parte, tener un registro de sismos cada vez más pequeños que, por ser más frecuentes, son la base del crecimiento acelerado del registro sísmico; aunque, por otra parte, ha hecho que los métodos manuales de análisis de sismos sean cada vez más difíciles de llevar a cabo.

Ante esta problemática, se sugiere un algoritmo de detección de sismos utilizando espectrogramas¹, debido a que permiten diferenciar las distintas señales que registra un sismómetro, por ejemplo, ruido producido por el paso de vehículos, señales provenientes de la atmósfera (huracanes, tormentas, oleaje, viento, etc.), entre otros. En particular el conjunto de datos utilizado contiene en su mayoría réplicas del sismo de Ometepec, Guerrero, con magnitud $M_w = 7.4$ registrado el 20 de marzo de 2012; por lo que el trabajo se centra en detectar sismos de baja tasa señal a ruido², que por su naturaleza, en su mayoría no son reportados en el catálogo del SSN.

El modelo implementado es un algoritmo de dos fases que permitió resolver la tarea de detección de sismos. Cada una de estas fases se describe a continuación:

1. Primera fase (clasificación): consistió en la preparación de los conjuntos de datos de entrenamiento, prueba y validación a partir de los eventos etiquetados por

¹Los espectrogramas son representaciones bidimensionales de la señal, tanto en el dominio del tiempo como en el de la frecuencia. En un espectrograma se puede observar el tiempo de arribo de las ondas sísmicas, así como su contenido espectral.

²Ocurre cuando la señal capturada por el instrumento fue generada por un sismo, pero la señal presenta una amplitud de onda tan baja (pequeña) que vuelve complicado discriminar entre una señal sísmica y ruido.

un analista, con ventanas de tiempo de 30 segundos. En cada una de estas ventanas se tienen las ondas que componen a un sismo (ondas de cuerpo y/o superficiales). La mayoría de estos eventos presentan solo ondas superficiales debido a que son réplicas del sismo de Ometepe.

2. Segunda fase (detección): Consistió en aplicar el modelo entrenado a una traza sísmica de 6 días reservada para tal efecto, utilizando ventanas deslizantes de 30 segundos, a la cual se le tuvo que aplicar un postproceso previo a la detección final.

El modelo propuesto en este trabajo cumple satisfactoriamente con el proceso de detección. Los resultados, tanto en la parte de clasificación como en la parte de detección, son muy competitivos dado que se encuentran en niveles de desempeño semejantes a los resultados de los modelos en el estado del arte.

En la parte de la clasificación se muestra como resultado una exactitud³ de 99.37 % y un Recall⁴ del 100 %. En cuanto a la detección, los resultados son un mAP = 98.74 %⁵. Estos resultados son confiables estadísticamente, por lo tanto, el algoritmo de detección de sismos desarrollado se puede implementar en otras estaciones sísmicas como apoyo en la determinación de la ocurrencia de estos eventos.

³La exactitud es la relación del número de predicciones correctas entre el total de predicciones.

⁴El recall o sensibilidad es la relación que existe entre el número de ejemplos positivos correctamente clasificados como positivos y el total de muestras positivas.

⁵El mAP se calcula promediando las **Average Precision (AP)** de todas las clases. Para comprender el AP es necesario entender varios conceptos como el IoU, la curva PR y el AUC. Dado que estos conceptos son bastante amplios, se invita al lector a revisar la Sección 4.14, en la cual se analizan con detalle.

Abstract

In recent years, the monitoring capacity of the Mexican National Seismological Service (SSN), as well as various networks around the world has grown exponentially. This is the result of the increasing number of seismic stations and the ease to store data in a digital way. This has allowed to have a better record of increasingly smaller earthquakes. Nonetheless, it has made manual methods of earthquake analysis increasingly difficult to carry out.

To deal with this issue, we propose an algorithm for earthquake detection based on both image processing and artificial intelligence algorithms using spectrograms⁶. We employed this type of representation since it allows classification between different signals recorded by the seismometers. For example, noise produced by passing vehicles, signals coming from the atmosphere (hurricanes, storms, waves, wind, etc.), among others. In particular, the dataset used in this work contains mostly aftershocks from the Ometepec, Guerrero earthquake, a magnitude $M_w = 7.4$ earthquake occurred on March 20, 2012. On this work, we focused on detecting low signal-to-noise ratio earthquakes that, due to their nature, are not reported in the SSN catalog most of the time.

The implementation of this model was made through a two-phase algorithm. Each of these phases is described as follows:

1. First phase (classification): This phase consisted in preparing the data for training, test and validation datasets from events labeled by an analyst with 30-second time windows. In each of this time windows, waves that make up an

⁶Spectrograms are 2D representations of a signal, in time and frequency domains. In a spectrogram, the arrival time of seismic waves can be observed, as well as their spectral content.

earthquake (body and surface waves) are present. It should be noted that most of the positive events do not present surface waves because they are aftershocks of the Ometepec earthquake.

2. Second phase (detection): This phase consisted in applying the trained model to a 6-day seismic trace reserved for this purpose, using 30-second sliding windows, to which a post-processing had to be applied prior to the final detection process.

The model proposed on this work satisfactorily accomplishes the earthquake detection process. The results, both in the classification part and in the detection part, are very competitive given that they are at levels of performance similar to the results of the state-of-the-art models.

Results in the classification stage have an accuracy⁷ of 99.37% and recall⁸ of 100%. In terms of detection, results have a $mAP = 98.74\%$ ⁹. These results are statistically reliable, therefore, the developed earthquake detection algorithm can be implemented in other seismic stations to support the determination of the occurrence of these events.

⁷Accuracy is the proportion of correct predictions against the total number of predictions.

⁸Recall is the relationship that exists between the positive samples correctly classify as positive, and the total number of positive samples.

⁹The mAP metric is calculated by averaging the **Average Precision (AP)** of each class. In order to understand the AP evaluation metric other concepts must be understood first as IoU, PR-curve and AUC. If you want to read more about it go to Section 4.14.

Índice general

Agradecimientos institucionales	II
Agradecimientos personales	IV
Resumen	V
Abstract	VII
1. Introducción	1
1.1. Antecedentes	2
1.1.1. Estado del arte	7
1.2. Objetivo	9
1.2.1. Objetivos particulares	9
1.3. Descripción general	9
2. Sismología y tectónica de placas en México	11
2.1. Sismología	11
2.1.1. Tipos de ondas sísmicas	11
2.1.2. Características de un sismo	13
2.1.3. Tectónica en México	14
3. Procesamiento digital de señales	17
3.1. Señales digitales	17
3.2. sProcesamiento Digital de Señales	18
3.3. Transformada de Fourier	18

3.4.	Transformada Discreta de Fourier	19
3.4.1.	Propiedades de la Transformada Discreta de Fourier	19
3.4.2.	La Transformada Rápida de Fourier	20
3.5.	Filtrado digital de señales	21
3.5.1.	Espectrograma	23
4.	Aprendizaje profundo	24
4.1.	Aprendizaje automático	24
4.1.1.	Tipos de aprendizaje	25
4.1.2.	Aprendizaje supervisado	26
4.2.	Redes Neuronales Artificiales (RNAs)	28
4.3.	Aprendizaje profundo	31
4.4.	Arquitectura de la corteza visual	32
4.5.	Capas convolucionales	34
4.5.1.	Filtros	36
4.6.	Capas de <i>Pooling</i>	38
4.7.	Funciones de error	39
4.8.	Optimizadores y tasa de aprendizaje	40
4.8.1.	Descenso de gradiente	41
4.8.2.	Aprendizaje adaptativo	43
4.9.	Parámetros y metaparámetros	44
4.9.1.	Metaparámetros de RNAs	44
4.10.	Tareas de visión por computadora	46
4.10.1.	Clasificación	46
4.10.2.	Localización	46
4.10.3.	Detección de objetos	47
4.11.	Detección de objetos utilizando una CNN	48
4.12.	Supresión de no máximos	49
4.13.	Modelos de aprendizaje profundo	50
4.14.	Métricas de evaluación	51

<i>ÍNDICE GENERAL</i>	XI
5. Metodología	56
5.1. Obtención y análisis de datos	57
5.2. Preprocesamiento	58
5.2.1. Conjuntos de datos para entrenamiento, validación y prueba	61
5.2.2. Generación de espectrogramas usando Obspy	64
5.3. Primera fase: entrenamiento de la CNN	65
5.3.1. Curva de aprendizaje y función de pérdida	69
5.3.2. Evaluación de la primera fase del modelo	72
6. Experimentos y resultados	74
6.1. Postproceso: Supresión de no máximos	75
7. Conclusiones y trabajo futuro	84
Appendices	86
A. Teorema del muestreo de Nyquist-Shannon	87
B. Conceptos básicos sobre redes neuronales	89
B.1. Función de pérdida o de error	89
B.2. Optimizador	89
B.3. Tasa de aprendizaje	90
B.4. Tamaño de lote	90
B.5. Conjunto de validación	90
B.6. Número de épocas	90
B.7. Dropout	91

Índice de figuras

1.1. Primer sismo registrado en el mundo, ocurrido el 17 de abril de 1889 en Japón y registrado en la ciudad de Potsdam, en lo que es ahora el país de Alemania, tomado de [Seismology Lab, 2010].	2
1.2. Volumen acumulado de datos sísmicos almacenado por el DMC de IRIS, al 1° de enero de 2020 [IRIS, 2020].	3
1.3. Mapa de las redes sismológicas permanentes del SSN, tomado de [SSN, 2020].	4
1.4. Ejemplo de un registro manual reportado por el Servicio Sismológico Nacional para el sismo de Huatulco de magnitud 7.4 del 23 de agosto de 1965. Archivo histórico del SSN.	5
1.5. Catálogo de sismos reportados por el SSN de 1990 al 22 de mayo de 2021. El aumento del número de eventos sísmicos reportados (localizaciones) en los últimos años responde principalmente a la instalación de nuevas estaciones sismológicas [SSN, 2018].	7
2.1. Deformaciones y movimiento de partículas asociadas al paso de las ondas (a) P y (b) S, tomado de [Kearey et al., 2013].	12
2.2. Deformaciones y movimiento de partículas asociadas al paso de las ondas (a) Rayleigh y (b) Love, tomado de [Kearey et al., 2013]. . . .	13
2.3. En la imagen se muestra el epicentro e hipocentro de un sismo generado a partir de una falla y la señal captada por una estación de monitoreo.	14
2.4. Distintos tipos de fallas asociados a los límites de placas a) Falla normal, b) Falla inversa y c) Falla transformante.	15

3.1.	Respuesta de frecuencia de filtro pasa bajas.	21
3.2.	Respuesta de frecuencia de filtro pasa altas.	22
3.3.	Ejemplos de espectrogramas generados a partir de la señal sísmica obtenida de la estación PNIG en Pinotepa Nacional, Oaxaca. a) Representación bidimensional de las ondas P y S de un evento sísmico y b) Señal proveniente de diversas fuentes (ruido).	23
4.1.	Aproximación del proceso de aprendizaje automático, tomado de [Géron, 2019].	26
4.2.	Ejemplo de un conjunto de datos de entrenamiento etiquetados, utilizados para clasificación de spam, tomado de [Géron, 2019].	27
4.3.	Diagrama de una neurona biológica, disponible en ww.wikipedia.org/neuron	28
4.4.	Neurona artificial tomada del modelo de Rumelhart y McClelland, tomado de [McClelland et al., 1986].	29
4.5.	Funciones de activación más comunes.	30
4.6.	Diferencia entre inteligencia artificial, aprendizaje automático y aprendizaje profundo.	32
4.7.	A medida que la señal visual se abre paso a través de las capas de neuronas cerebrales consecutivas, las neuronas responden a patrones más complejos en campos receptivos más grandes, tomado de [Géron, 2019].	33
4.8.	Capas de una CNN con sus correspondientes campos receptivos locales, tomado de [Géron, 2019].	34
4.9.	Capas de una CNN con sus campos receptivos locales, tomado de [Géron, 2019].	35
4.10.	Reducción de dimensión de una imagen (capa de entrada) de 5×7 con padding de ceros (7×9), conectada a una capa más pequeña de 3×4 , usando un tamaño del campo receptivo local de $f_h = 3 \times f_w = 3$ y un <i>stride</i> de 2, [Géron, 2019]. Las medidas se consideran en píxeles. . . .	36
4.11.	Mapas de <i>características</i> de una CNN, tomado de [Géron, 2019] . . .	37

4.12. Capa de Pooling que reduce el mapa de características (imagen) de 8×6 a un nuevo mapa de tamaño 4×3 tomando el valor máximo (con $s_h, s_w = 2$, salto de 2 y sin padding), modificado de [Géron, 2019]. Los tamaños se consideran en píxeles. 39

4.13. Ejemplo de pendientes obtenidos a partir del cálculo de la derivada en los puntos de la función. 41

4.14. Descenso de gradiente para una gráfica de dos dimensiones (2D) . . . 42

4.15. En la figura se muestran dos casos que pueden ocurrir cuando se intenta optimizar una función: a) tamaño de la tasa de aprendizaje y b) estancamiento en mínimos locales. 42

4.16. Ejemplo de descenso de gradiente adaptativo. 44

4.17. Ejemplo de la tarea de clasificación, donde al lado izquierdo se tiene la clase sismo y de lado derecho la clase ruido. 47

4.18. Ejemplos de la tarea de a) localización y b) detección de sismos en una imagen. 47

4.19. Ejemplo con múltiples detecciones. Obtenido a partir de deslizar una ventana (color negro) a través de la imagen, las ventanas en color rojo son los rectángulos que delimitan al objeto, tomado de [Géron, 2019] 48

4.20. Ejemplo de detección de sismos deslizando ventanas de 30s a través del espectrograma (imagen). 49

4.21. Ejemplo del antes (arriba) y después (abajo) de aplicar la snm para el caso de detección de sismos. 50

4.22. Ejemplo de IoU como métrica para evaluar un rectángulo delimitador predicho (rectángulo rojo) del correcto (rectángulo verde), tomado de [Géron, 2019]. 51

4.23. Ejemplo de uso de la IoU , donde se considera que una $IoU \geq 0.8$ es un buen resultado. 52

4.24. Curva Recall-Precisión, tomada de [Géron, 2019] 54

5.1. Ubicación de la estación del SSN con clave PNIG y del lugar donde ocurrió el sismo de Ometepec el 20 de marzo de 2012.	57
5.2. Histograma de duraciones de sismos que conforman el conjunto positivo, usando la componente Z y filtrando 425 eventos (ver texto).	59
5.3. Ocurrencia sísmica durante el periodo de enero-abril de 2012 registrada por la estación PNIG en Pinotepa Nacional, Oaxaca.	59
5.4. Ocurrencia de sismos por intervalos. El conjunto positivo se divide en dos subconjuntos: el primero se usa para la primera fase de entrenamiento del modelo, dentro del lapso del 1 de enero al 14 de abril de 2012 y, el segundo, para la fase de detección en una traza de 6 días, del 15-20 de abril de 2012	60
5.5. A la izquierda se muestran 4 espectrogramas positivos (sismos, donde se observan las ondas P y S que componen el sismo). A la derecha se muestran 4 ejemplos de espectrogramas negativos (ruido) generados para el conjunto de entrenamiento. Se añaden los identificadores alfanuméricos de los sismos (id), así como un id para los eventos negativos, que se basa en las marcas de tiempo.	63
5.6. Ejemplo de un sismograma (arriba) y su respectivo espectrograma (abajo) registrado por la estación sismológica PNIG en Pinotepa Nacional, Oaxaca.	64
5.7. Ejemplos de espectrogramas generados para el caso a) positivo y b) negativo (arriba) junto con su respectiva imagen redimensionada (abajo) de tamaño 60 × 60 px.	66
5.8. Desempeño del modelo de CNN durante el proceso de determinación de metaparámetros para el proceso de clasificación de imágenes (sismos).	70
5.9. Arquitectura de la CNN usada para resolver la tarea de clasificación de imágenes (sismos).	71
5.10. Matriz de confusión	72

6.1. Ejemplo de múltiples rectángulos delimitadores detectados para cada sismo en la imagen.	74
6.2. Ejemplo del antes y después de la supresión de no máximos. Después de la SNM sólo el rectángulo delimitador que cuente con la probabilidad más alta de ser sismo se mantendrá y el resto se eliminará.	75
6.3. Probabilidades del conjunto de rectángulos delimitadores para cada sismo, donde se observa una distribución normal de la distribución numérica de los datos.	76
6.4. Ejemplo de aplicar la métrica <i>IoU</i> para cada ventana donde el modelo detecta un sismo.	78
6.5. Resultados de la segunda fase (la detección de sismos) del modelo propuesto, variando el umbral de probabilidad de detección u	79
6.6. Matriz de confusión para los resultados obtenidos en la segunda fase del algoritmo de detección de sismos.	81
6.7. Curva PR correspondiente a la segunda fase del modelo propuesto. . .	81
A.1. Medidas estadísticas descriptivas de la señal sísmica obtenida por la estación sismológica de Pinotepa Nacional, Oaxaca; entre el 1° de enero y el 20 de abril de 2012, a 100 mps. a) enero, b) febrero, c) marzo y d) abril.	88

Capítulo 1

Introducción

La propuesta que se plantea en esta tesis consiste en entrenar un algoritmo de Inteligencia Artificial (IA). Este algoritmo será capaz de reconocer el aspecto de un sismo usando el espectro de la señal, es decir, mediante una representación en forma bidimensional semejante a una imagen, denominada *espectrograma*¹.

Adicionalmente al objetivo antes mencionado, se enfoca en generar un algoritmo para la detección de sismos con baja relación señal a ruido. Este tipo de sismos frecuentemente no es reportado en el catálogo del Servicio Sismológico Nacional (SSN) dadas sus características específicas y, debido a que ocurren con mucha más frecuencia que los sismos de mayor magnitud, de acuerdo con la ley de Gutenberg-Richter [Gutenberg, 1956], su análisis se vuelve cada vez más difícil de llevar a cabo con métodos manuales.

Por otro lado, el número de registros sísmicos que se obtienen cada año aumenta de manera exponencial a medida que se instalan nuevas estaciones, lo que genera la necesidad de automatizar los procesos de detección que permitan tener una mejor detección de los sismos de pequeñas magnitudes que ocurren en México.

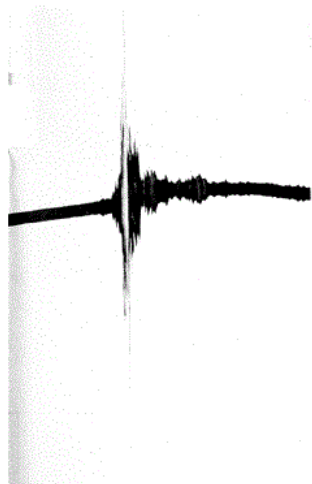


Figura 1.1: Primer sismo registrado en el mundo, ocurrido el 17 de abril de 1889 en Japón y registrado en la ciudad de Potsdam, en lo que es ahora el país de Alemania, tomado de [Seismology Lab, 2010].

1.1. Antecedentes

El primer registro sísmico (Figura 1.1) se obtuvo en la ciudad de Potsdam (actual Alemania) en 1889, utilizando un péndulo que en un principio estaba destinado a medir cambios en la gravedad terrestre provocada por el movimiento de otros planetas. Sin embargo, el 17 de abril de ese año, una señal inesperada apareció en los registros del péndulo, la cual sería correlacionada meses después con un sismo ocurrido en Japón [Seismology Lab, 2010]. Pocos años después, en 1892, el ingeniero Británico John Milne desarrolló en Japón los primeros instrumentos sísmicos mecánicos que permitieron la instalación de 40 observatorios alrededor del mundo para principios del siglo XX [Lay and Wallace, 1995].

Actualmente, existen miles de estaciones sísmicas distribuidas alrededor de todo el mundo que registran una enorme cantidad de datos de forma digital todos los días, lo que implica un aumento significativo en el registro sísmico a nivel mundial. Tan solo en el año 2008 el Centro de Manejo de Datos (*DMC*, por sus siglas en inglés) del Centro de Instituciones de Investigación Incorporadas para la Sismología (*IRIS*, por sus siglas en inglés) había superado los 81 TB de datos; y, el 1º de enero de 2020,

¹Un espectrograma es una representación bidimensional de una señal en la que se puede visualizar el contenido de frecuencias de una señal en función del tiempo.

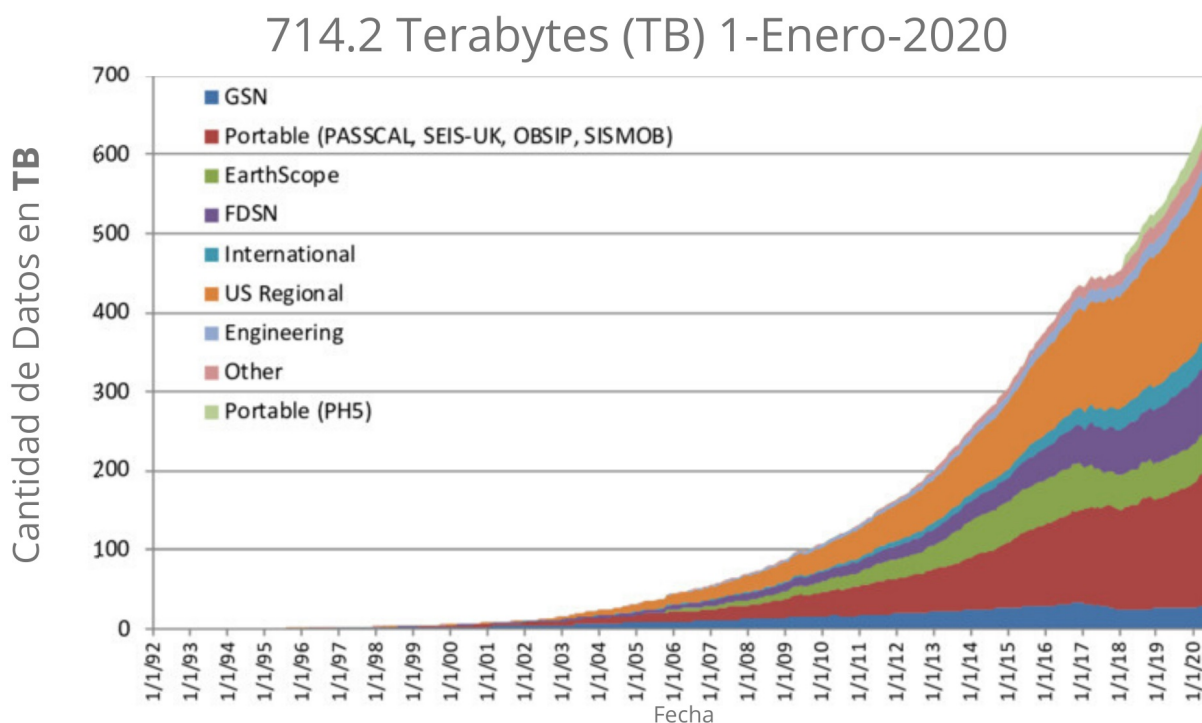


Figura 1.2: Volumen acumulado de datos sísmicos almacenado por el DMC de IRIS, al 1° de enero de 2020 [IRIS, 2020].

cuenta con **714.2 TB** [IRIS, 2020], Figura 1.2.

En México, la institución encargada de registrar, almacenar y distribuir los datos de las estaciones sismológicas es el Servicio Sismológico Nacional (SSN), el cual tiene registros sísmicos desde 1905. Actualmente, el SSN cuenta con cuatro redes sismológicas permanentes (Figura 1.3): la Red de Banda Ancha es la más grande de todas y está distribuida alrededor de todo el país, compuesta por 63 estaciones (53 en tiempo real); la Red del Valle de México (RSVM), integrada por 30 estaciones de monitoreo; la red Táchana integrada por 3 estaciones ubicada en la frontera sur con Guatemala y la red *-Comprehensive Nuclear-Test-Ban Treaty Organization (CTBTO)-*, ubicada en la isla Socorro e integrada por 3 estaciones [Pérez-Campos et al., 2018]. Además de las anteriores, el SSN usa numerosas redes temporales dirigidas al estudio de la sismicidad de zonas en específico, por ejemplo las redes MASE (100 estaciones) [Lukac et al., 2009, Husker et al., 2008], VEOX (50 estaciones) [Melgar and Pérez-Campos, 2011], MARS (50 estaciones) y CODEX (23 estaciones) [Watkins et al., 2018].

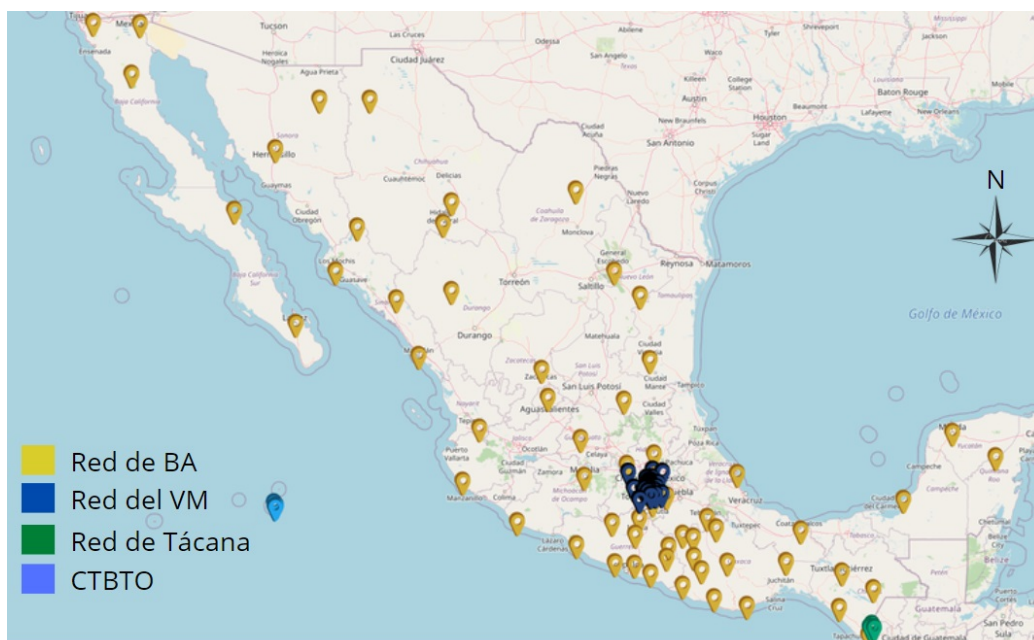


Figura 1.3: Mapa de las redes sismológicas permanentes del SSN, tomado de [SSN, 2020].

En sus inicios, la detección de sismos se realizaba de forma manual a partir de la señal que se registraba en las estaciones sísmicas de forma analógica (Figura 1.4). El procedimiento consistía en colocar a un analista a revisar las amplitudes y los cambios de onda de la señal, traza por traza. Este proceso representaba varias desventajas, además de ser bastante tardado, generaba detecciones sesgadas dependiendo del operador en turno. Con el desarrollo de nuevas tecnologías y el uso de software específico para la detección de sismos de manera semiautomática [Perol et al., 2018], se ha facilitado este proceso, pero en general, aún es tardado y continúa presentando un factor de sesgo, ya que requiere de la validación humana. Además, requiere ser ajustado para las condiciones locales de cada región o zona donde se encuentran las estaciones sismológicas [Sabbione and Velis, 2010].

Durante las últimas décadas se han desarrollado numerosas técnicas para detectar o determinar los posibles sismos de forma automática o semiautomática. En la actualidad, la estrategia más común es utilizar un algoritmo que identifica posibles sismos a partir de una métrica que mide cambios abruptos, para después ser validados de manera manual por un analista. Entre los métodos comunes, uno de los más

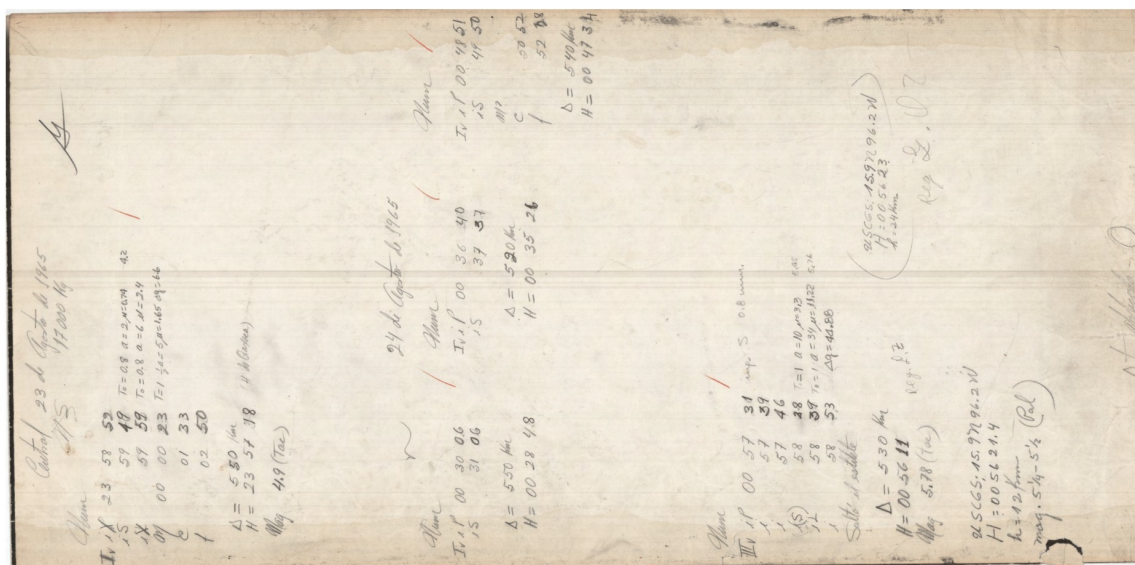


Figura 1.4: Ejemplo de un registro manual reportado por el Servicio Sismológico Nacional para el sismo de Huatulco de magnitud 7.4 del 23 de agosto de 1965. Archivo histórico del SSN.

utilizados es el método *-Short Time Average over Long Time Average (STA/LTA)-* [Allen, 1978] y el *-Cross-Correlation (CC)-* [Yoo and Han, 2009]. Este tipo de modelos presentan una eficiencia mayor, pero son propensos a la detección de falsos positivos y, ante la aparición de ruido en la señal, no logran detectar los sismos debido principalmente a una mala relación de la tasa señal a ruido que existe en el instrumento [Mousa et al., 2011].

México es un país sísmicamente activo donde han ocurrido grandes terremotos que han causado pérdidas tanto materiales como humanas. La región que presenta la mayor actividad sísmica se encuentra en el suroeste del país, específicamente en la zona más meridional en el borde entre la placa de Cocos y la placa Norteamericana [Nunez-Cornu, 1996], donde ocurren numerosos terremotos tanto de magnitud pequeña que provocan ligeras deformaciones con intervalos de ocurrencia cortos; como sismos más grandes, que causan incontables daños, como los ocurridos en los sismos de 1957 ($M_w = 7.8$), el de Michoacán en 1985 ($M_w = 8.1$) y los de Puebla-Morelos ($M_w = 7.1$) y Oaxaca ($M_w = 8.2$) en 2017.

El enfoque de este trabajo es la detección de sismos de baja magnitud (< 3.5) derivada su importancia de dos factores determinantes: por un lado está el creciente interés

en este tipo de sismos debido principalmente a estudios recientes [Meier et al., 2019, Tibi et al., 2019] que han demostrado que el hecho de registrarlos permite obtener una imagen más clara sobre los movimientos y la secuencia de eventos sísmicos (visibiliza los sismos previos, los principales y las réplicas); con ello se puede obtener más información de sismos que aparentemente son aislados y repentinos [Brodsky, 2019]. Por otro lado, los sismómetros apenas logran detectar una minoría de sismos de pequeñas magnitudes, debido a su dificultad de discriminación (baja tasa señal a ruido), lo que implica una gran pérdida de información dado que son sismos mucho más frecuentes que los sismos de escala mayor [Gutenberg, 1956] (el orden de frecuencia es de 10 veces por grado de magnitud, es decir, un sismo de magnitud 3 ocurre 1×10^4 veces más que un sismo de magnitud 7).

Ante esta dificultad de discriminación, los sismos de pequeñas magnitudes frecuentemente no son reportados en el catálogo del SSN, lo que sugiere la necesidad del uso de herramientas computacionales cada vez más robustas para evitar la pérdida de información y poder realizar estudios más detallados de sismos.

Con relación al ruido en la señal, las técnicas de aprendizaje profundo han mostrado ser suficientemente robustas y con un alto rendimiento en la automatización de procesos [Jordan and Mitchell, 2015] lo cual, aunado a los avances en poder de cómputo a través del uso de diversas unidades de procesamiento (CPU, GPU y TPU) cada vez de mayor capacidad, así como del uso de la enorme cantidad de información disponible (ver Figura 1.5) y que se genera actualmente, se han logrado obtener resultados sin precedentes en diferentes áreas de investigación, por ejemplo: en sistemas de reconocimiento facial [Sipe et al., 2013], en tareas de visión por computadora [Krizhevsky et al., 2012] y en el procesamiento de lenguaje natural [Brown et al., 2020, Roller et al., 2020], entre otras.

Al mismo tiempo que el registro sísmico a nivel mundial se ha venido multiplicando significativamente de manera constante, los sismos detectados por el SSN reportados en su catálogo presentan un incremento similar (como se muestra en la Figura 1.5), debido al aumento en el número de estaciones sismológicas. Esto genera una cantidad cada vez mayor de datos y, ante esto, la necesidad de automatizar su análisis.

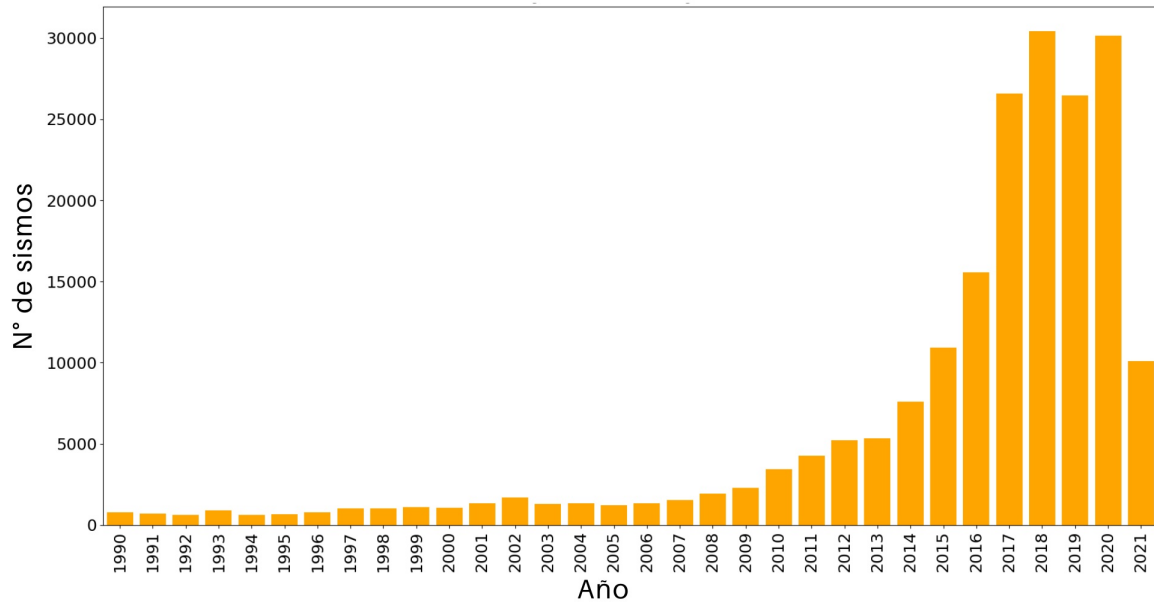


Figura 1.5: Catálogo de sismos reportados por el SSN de 1990 al 22 de mayo de 2021. El aumento del número de eventos sísmicos reportados (localizaciones) en los últimos años responde principalmente a la instalación de nuevas estaciones sismológicas [SSN, 2018].

En la siguiente sección, se describe el estado actual de los trabajos más relevantes en cuanto al uso de aprendizaje profundo para resolver tareas de detección de señales sísmicas.

1.1.1. Estado del arte

Los trabajos en la literatura más actualizada con relación al área de estudio de esta investigación presentan distintas combinaciones de parámetros de entrada para la red neuronal.

Por ejemplo, algunos utilizan las tres componentes o únicamente la componente Z, que por lo general es la que presenta una mejor relación tasa señal a ruido. Además al obtener solo una imagen de esa componente, se puede representar con tres canales de entrada (RGB) que simulan las tres componentes (N, E, Z), por lo tanto, el uso de una o las tres componentes, depende de la arquitectura de red que arroje los resultados deseados.

A continuación se describe el estado del arte.

[Ross et al., 2018] entrenaron un modelo de aprendizaje profundo para clasificar las fases de llegada² que componen un sismo. El entrenamiento del modelo se llevó a cabo usando datos etiquetados por analistas de la “Red sísmica del sureste de California” -*Southern California Seismic Network (SCSN)*- (273,882 sismogramas) a través de 692 estaciones de banda ancha durante 2000-2017.

[Perol et al., 2018] proponen la detección de señales sísmicas así como su ubicación, empleando un modelo de aprendizaje profundo y comparan su propuesta con técnicas convencionales de detección de sismos, como el método de *autocorrelación* [Pearl, 1936] y el -*Fingerprint and Similarity Association Trace (FAST)*- [Jung, 1910]. Los datos consisten en 2,021 eventos sísmicos, desde el 15 de febrero de 2014 hasta el 16 de noviembre de 2016, con un rango de magnitud de [-0.2, 5.8], obtenidos de las estaciones GS.OK027 y GS.OK029 cercanas a la ciudad de Guthrie, Oklahoma, EE. UU.

[Zhu et al., 2018] también aplican un modelo de aprendizaje profundo para detectar eventos y fases de llegada de un sismo. Los datos son registros sísmicos de la región de Sichuan, China detectados durante el año 2008. Se trata de 9,648 eventos etiquetados como sismos, obtenidos de 14 estaciones distintas establecidas en la zona.

Por otra parte, [Zhou et al., 2019] implementan un algoritmo híbrido basado en redes neuronales convolucionales [LeCun et al., 1989] -*Convolutional Neural Networks (CNNs)*- y redes neuronales recurrentes [Hochreiter and Schmidhuber, 1997] -*Recurrent Neural Networks (RNNs)*- para la detección del evento sísmico y, posteriormente, para la detección de las fases que lo componen. Los datos son eventos sísmicos con ventanas de tiempo de 30 segundos de duración con las tres componentes (N, E y Z) obtenidos de 16 estaciones sismológicas de la región de Wenchuan, China; siendo 135,966 eventos etiquetados como sismos y con sus fases P y S.

Finalmente, [Meier et al., 2019] entrenaron modelos basados en CNNs y redes generativas adversarias [Goodfellow et al., 2014] -*Generative Adversarial Networks (GANs)*- combinadas en un modelo híbrido con bosques aleatorios [Ho, 1995] -*Random*

²Son los tiempos de llegada (marcas de tiempo) de cada una de las ondas que componen un sismo (cuerpo y superficiales) registradas por un estación sismológica

Forests (RF)-, una red neuronal completamente conectada [Schmidhuber, 2015] -*Feed-forward Neural Networks (FNN)*- y una RNN. Los autores trabajaron con 374,000 eventos etiquetados como sismos, con magnitud de [3, 9.1], y 946,000 eventos etiquetados como ruido. Los datos positivos (sismos) tienen una frecuencia de 0.075Hz, con las tres componentes (N, E y Z) como entrada para todos los algoritmos usados.

1.2. Objetivo

Detectar sismos con una baja tasa señal a ruido a partir de las señales sísmicas representadas como espectrogramas, mediante la aplicación de técnicas de aprendizaje profundo.

1.2.1. Objetivos particulares

1. Crear los conjuntos de datos para entrenar, validar y probar el modelo de aprendizaje profundo a desarrollar.
2. Desarrollar un modelo de aprendizaje profundo para detectar sismos, aproximando un modelo de redes de convolución.
3. Evaluar y discutir los resultados del modelo desarrollado.

1.3. Descripción general

El presente trabajo se divide en 7 capítulos. A continuación, se da una breve descripción de cada uno de ellos.

- Capítulo 1: se presenta una breve introducción del trabajo realizado, los antecedentes, la justificación y los objetivos generales y particulares que se plantearon para llevar a cabo esta tesis.
- Capítulo 2: se presentan los conceptos básicos en el área de sismología y la tectónica de placas en México, desde cómo se genera un sismo, los tipos de ondas que genera, hasta los distintos tipos de fallas que existen.

- Capítulo 3: se señalan los principales métodos y algoritmos utilizados para el procesamiento de señales digitales, así como los tipos de filtrado de frecuencias más comunes. En particular, el algoritmo de la Transformada Rápida de Fourier fue utilizado en este trabajo para obtener una imagen a partir de las frecuencias de una señal en función del tiempo, conocido como espectrograma.
- Capítulo 4: se introduce al campo del aprendizaje profundo, hablando primero sobre el aprendizaje automático, la Inteligencia Artificial (IA) y los principales paradigmas de aprendizaje de máquina, las tareas de visión por computadora y cuáles de estas se resolvieron.
- En los Capítulos 5 y 6 se detallan todos los pasos que se llevaron a cabo para resolver la tarea de detección de sismos con un algoritmo de dos fases. La primera fase (Capítulo 5) consistió en el preprocesamiento, la creación de los conjuntos de datos positivo y negativo, los conjuntos de entrenamiento, validación y prueba, hasta la evaluación del modelo de redes neuronales convolucionales. Mientras tanto, en la segunda fase (Capítulo 6), se realiza la detección aproximando la red neuronal y aplicando un postproceso para obtener la detección final, cuyo resultado se evalúa utilizando las métricas de evaluación más comunes en el área de aprendizaje automático.
- En el capítulo 7 se discuten los resultados logrados y los caminos a futuro sobre los cuales se puede continuar y profundizar lo realizado.

A continuación, en los siguientes tres capítulos (Capítulo 2, 3 y 4) se detalla el marco teórico-conceptual básico para la comprensión de los sismos así como del aprendizaje profundo.

Capítulo 2

Sismología y tectónica de placas en México

En este capítulo se detalla una breve terminología de cada uno de los conceptos en el área de sismología necesarios para el desarrollo de la presente investigación.

2.1. Sismología

La sismología es la ciencia encargada de caracterizar, estudiar y registrar los sismos. Gracias a esta ciencia fue posible confirmar la teoría de tectónica de placas, además ha permitido conocer cuál es la estructura interna de la Tierra. Las herramientas creadas en esta disciplina han sido fundamentales para localización de yacimientos petroleros y para la caracterización de suelos en la ingeniería civil, entre otras aplicaciones.

En las secciones siguientes se describen varios de los conceptos principales de la sismología: las ondas sísmicas y los sismos; y se comenta sobre la tectónica en México.

2.1.1. Tipos de ondas sísmicas

Las ondas sísmicas se clasifican de acuerdo con el tipo de movimiento que generan en el medio. En primera instancia las ondas sísmicas se agrupan en ondas superficiales

y de cuerpo. Estas últimas se caracterizan por tener una trayectoria de propagación principalmente en el interior de la Tierra y se clasifican en ondas P (primarias) y S (secundarias). Las ondas P son conocidas como ondas de compresión o longitudinales y se caracterizan por viajar más rápido y con menor amplitud con respecto a las ondas S. Las ondas P provocan un tipo de oscilación de tal manera que empujan (comprimen) y jalan (expanden) el suelo en la dirección de su propagación (Figura 2.1 inciso (a)). Posterior al arribo de estas ondas se registran las ondas S, que se caracterizan por tener una amplitud mucho mayor que las ondas P. Las ondas S u ondas de corte mueven el material por el que viajan en dirección perpendicular a la dirección de propagación (Figura 2.1 inciso (b)) [Kearey et al., 2013].

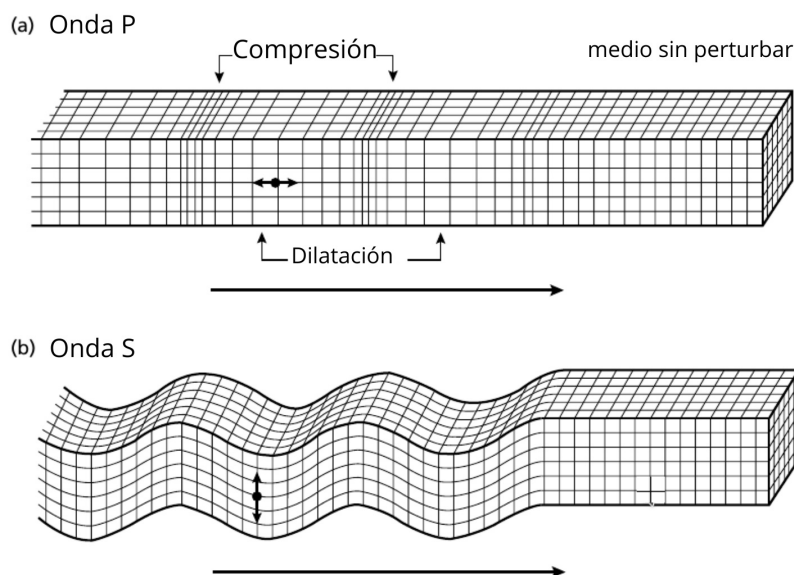


Figura 2.1: Deformaciones y movimiento de partículas asociadas al paso de las ondas (a) P y (b) S, tomado de [Kearey et al., 2013].

Las ondas superficiales, en cambio, viajan cercanas a la superficie. Se clasifican por sus características en: ondas Rayleigh (R) y ondas Love (L). Las ondas Rayleigh generan ciclos en dirección contraria con respecto a su dirección de viaje mientras que, las ondas Love, son más parecidas a las ondas S, pero con la diferencia de que éstas viajan en forma horizontal perpendicular a su dirección de propagación (Figura 2.2).

Los datos con los que se cuenta para el trabajo de esta tesis son en su mayoría

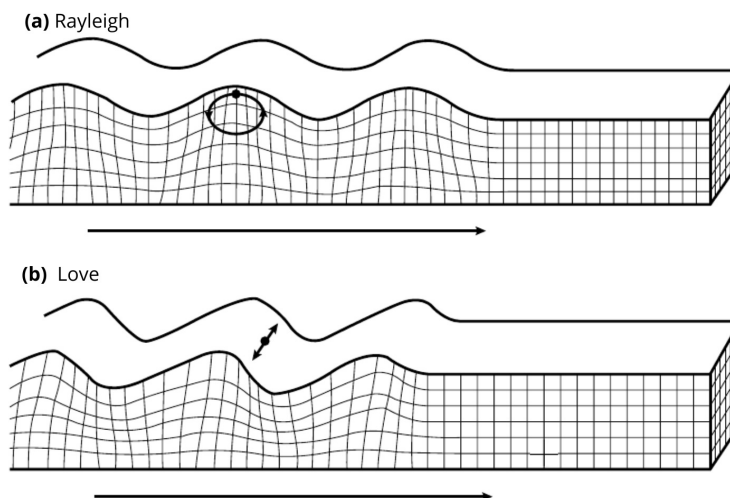


Figura 2.2: Deformaciones y movimiento de partículas asociadas al paso de las ondas (a) Rayleigh y (b) Love, tomado de [Kearey et al., 2013].

réplicas de un sismo con magnitud de 7.4, registrado el 20 de marzo de 2012 con epicentro en Ometepec, Guerrero. Estas réplicas son de mucho menor magnitud que el evento del 20 de marzo y, debido a su cercanía con la estación sísmica, no generan ondas superficiales.

Los sismos tienen ciertas propiedades o características que ayudan para su identificación, mismas que se detallan en el apartado siguiente.

2.1.2. Características de un sismo

Los sismos se componen de diversas propiedades tales como ubicación, magnitud y área de ruptura. La ubicación de un sismo está dada por la latitud y longitud del epicentro. Al sitio donde ocurrió la ruptura por debajo de la superficie se le conoce como hipocentro. En la Figura 2.3 se ilustra la diferencia entre epicentro e hipocentro.

Debido a que el momento sísmico M_0 es un número que puede abarcar varios órdenes de magnitud, suele expresarse como una cantidad adimensional conocida como magnitud de momento, que es la que reporta la mayoría de los servicios sismológicos alrededor del mundo [Hanks and Kanamori, 1979].

La magnitud de momento M_w sigue la escala logarítmica y se expresa de la si-

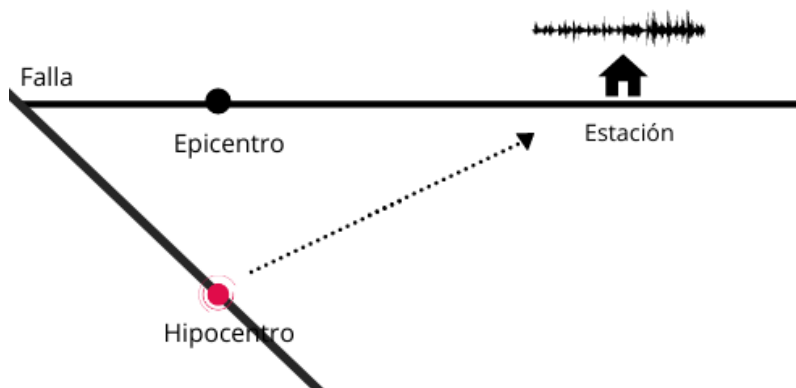


Figura 2.3: En la imagen se muestra el epicentro e hipocentro de un sismo generado a partir de una falla y la señal captada por una estación de monitoreo.

guiente forma:

$$M_w = \frac{3}{2}(\log_{10} M_0 - 16) \quad (2.1)$$

Se estima a partir del momento sísmico M_0 , calculado del coeficiente de fricción μ , el área de ruptura A y el desplazamiento promedio d , definido como sigue:

$$M_0 = \mu A d \quad (2.2)$$

2.1.3. Tectónica en México

La tectónica en México es bastante compleja debido a que se compone de 5 placas tectónicas, estas son:

- **La placa Norteamericana** que abarca casi en su totalidad toda América del norte, parte del occidente del Atlántico Norte y sur del océano Glacial Ártico.
- **La placa del Pacífico** cubre casi en su totalidad el océano Pacífico.
- **La placa del Caribe** cubre gran parte de América central y el mar Caribe.
- **La placa de Cocos** causa la mayor parte de la sismicidad en México y se localiza en el límite costero de los estados de Jalisco, Michoacán, Oaxaca, Guerrero y Chiapas.

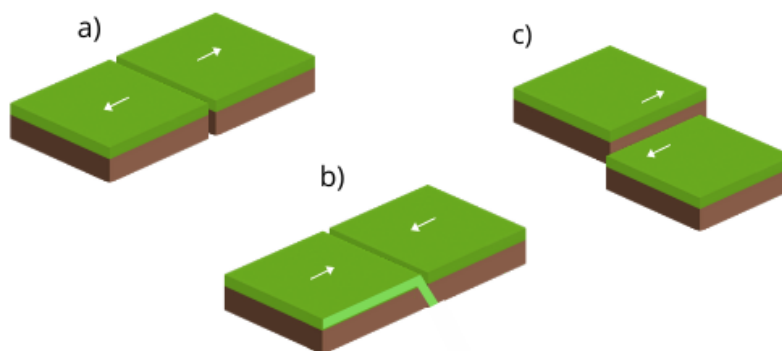


Figura 2.4: Distintos tipos de fallas asociados a los límites de placas a) Falla normal, b) Falla inversa y c) Falla transformante.

- **La placa de Rivera** es la placa más pequeña y se ubica al norte de la placa de Cocos.

Las placas tectónicas presentan diversos movimientos entre ellas, por ejemplo la placa Norteamericana limita al suroeste con la placa del Caribe, donde se presenta un movimiento de tipo transformante (Figura 2.4 inciso (c)) creando la falla conocida como Montagua-Polochic. Otro par de placas que presentan este tipo de movimiento son la placa del Pacífico y la Norteamericana en su parte central, mientras que en la parte más al norte se presenta un movimiento normal (Figura 2.4 inciso (a)), donde se forma el mar de Cortés.

Las placas de Cocos, Rivera y Pacífico generan un movimiento de tipo normal, creando la cadena volcánica submarina que delimita con el borde de cada una de estas placas. Por último, la placa de Cocos y la de Rivera presentan un movimiento inverso (Figura 2.4 inciso (b)) con respecto a la placa Norteamericana. Esta zona en particular es importante para el trabajo de esta tesis debido a que es aquí donde se ubica la estación sísmica de Pinotepa Nacional, Oaxaca; de donde se obtuvieron los registros del 1 de enero al 20 de abril de 2012 y en donde ocurrió el sismo del 20 de marzo de 2012 con $M_w = 7.4$, con epicentro en Ometepec, Guerrero. Además, esta zona se caracteriza por tener la mayor actividad sísmica de todo el país.

De manera general el choque de las placas tectónicas produce fricción que va acumulando energía durante el movimiento, hasta que esta energía llega a ser mayor

que la fricción, lo que provoca una liberación espontánea de toda la energía acumulada. La liberación de energía se realiza de maneras distintas, como calor, deformación de rocas y energía sísmica que se propaga a través del medio hasta llegar a la superficie [Helfrich and Wood, 2001].

Las placas tectónicas son las principales responsables de la mayoría de los eventos sísmicos que ocurren alrededor del mundo. Para monitorear este tipo de eventos se crearon diversos instrumentos que registran en forma de señal las ondas generadas por los sismos.

En el capítulo siguiente, se detallan los distintos tipos y métodos para procesar señales digitales, en particular la Transformada Rápida de Fourier debido a que permite obtener una imagen a partir del espectro de la señal.

Capítulo 3

Procesamiento digital de señales

Una señal es una función que recoge información acerca de un fenómeno físico [Priemer, 1991]. Nuestro entorno está lleno de señales que van desde voltajes generados por nuestro cerebro, los radares, sonares, ondas sísmicas, señales de satélites, entre muchos otros [Rao and Swamy, 2018]. En este capítulo se detallan los métodos más usados para generar una imagen a partir de la manipulación de una señal. En particular el uso de la Transformada de Fourier facilita la obtención del contenido espectral de tales señales. Por lo que, en la presente investigación, la señal captada por el sismómetro se convertirá a un espectrograma utilizando una convención de este método.

3.1. Señales digitales

Las señales digitales son un paquete de información que se puede expresar como una función del tiempo. Esta información se puede transmitir de varias maneras, en el caso de las señales que son captadas por un sismómetro se presenta una secuencia de amplitudes en función del tiempo [Stanley et al., 1988]. La señal proveniente de distintas fuentes como las vibraciones sísmicas, reflectividad de los objetos u ondas de sonido son captadas por un sensor de datos que las transforma a través de un *-Digital Analog Converter (DAC)-* [Rao and Swamy, 2018], mejor conocido como digitalizador. Un digitalizador convierte la señal analógica a digital aplicando técnicas

y métodos matemáticos, como la Transformada de Fourier -*Fourier Transform (FT)*-. Cabe mencionar que en este trabajo no se realizará la derivación matemática de la FT, para tal efecto se sugiere revisar la siguiente bibliografía [Nussbaumer, 1981].

El hecho de hacer la conversión de señal analógica a digital se debe a múltiples ventajas que ofrecen las señales digitales:

- **Precisión:** Las señales digitales proveen una precisión deseable a manera en que se aumenta el número de bits.
- **Fácil almacenamiento:** Las señales digitales se pueden almacenar en diversos formatos sin pérdida de información.
- **Procesamiento:** Permiten la aplicación de procesos y filtrados sofisticados como la FT que, en contraparte, las señales analógicas no.
- **Disponibilidad:** El hecho de poderse almacenar en dispositivos de alta fidelidad, permite su acceso a través de Internet en todo momento.

3.2. sProcesamiento Digital de Señales

El objetivo del Procesamiento Digital de Señales (PDS) es extraer y/o modificar la información de una señal a través de algoritmos, técnicas y métodos matemáticos. En este trabajo de tesis en particular el análisis de Fourier será útil debido a que permite generar una representación bidimensional, semejante a una imagen, de una señal sísmica a partir del cálculo de sus componentes espectrales.

3.3. Transformada de Fourier

La Serie de Fourier constituye un análisis práctico para trabajar con señales digitales por sus nobles propiedades, que se detallan más adelante en la Sección 3.5. Desde su aparición en el Siglo XVIII en el estudio de las vibraciones de las cuerdas [Barón de Fourier, 1822], las series de Fourier se han convertido en un instrumento

indispensable en el procesamiento digital de señales. La idea fundamental se basa en aproximar la función de Fourier a través de series de funciones periódicas (senos y cosenos) [Zhang, 2019].

La FT busca mapear una función en el dominio del tiempo a otra función en el dominio de frecuencias. En otras palabras, obtiene el espectro de la señal a partir del cálculo de la potencia de las amplitudes de frecuencia de la señal.

3.4. Transformada Discreta de Fourier

Las componentes espectrales de una señal se calculan a partir de los datos analógicos transformados a valores discretos por un digitalizador. Para calcular estas componentes se debe discretizar la Serie de Fourier correspondiente, dando como resultado una formulación discretizada de la FT conocida como *-Discrete Fourier Transform (DFT)-*.

El cálculo directo de la DFT a partir del planteamiento matemático resulta computacionalmente ineficiente, no obstante, al aplicar las propiedades de la FT sobre su formulación discretizada, se puede aumentar la eficiencia en el cálculo [Brigham, 1988].

3.4.1. Propiedades de la Transformada Discreta de Fourier

La DFT tiene ciertas propiedades que la convierten en uno de los métodos más sofisticados para el procesamiento de señales. Las más importantes para esta investigación son:

1. **Linealidad:** esta propiedad indica que la DFT de la suma de dos funciones con dominio en el tiempo, será igual a la suma de las DFTs de cada una de las funciones.
2. **Traslación:** esta propiedad señala que trasladar una función en el dominio del tiempo (o el de la frecuencia) corresponde a una multiplicación por una función exponencial compleja en el dominio de la frecuencia (o del tiempo, respectivamente).

3. **Simetría:** esta propiedad indica que la magnitud de la DFT de una función real (como por ejemplo, una imagen digital) $f(x, y) = F(u, v)$ es una función simétrica conjugada, donde $F(u, v)$ es la DFT de $f(x, y)$. Por tal motivo:

$$|F(u, v)| = |F(-u, v)| = |F(u, -v)| = |F(-u, -v)| \quad (3.1)$$

donde $|F(u, v)|$ es la magnitud de $F(u, v)$ y la propiedad no se restringe solamente a funciones de dos variables.

3.4.2. La Transformada Rápida de Fourier

En términos prácticos la discretización de la FT es necesaria para trabajar con números reales debido a que en una computadora no se pueden almacenar valores continuos. Sin embargo, el cálculo de la DFT resulta muy alto en términos computacionales, por ello es que existen algoritmos que logran optimizar este proceso. La Transformada Rápida de Fourier -*Fast Fourier Transform (FFT)*- es un algoritmo para calcular la DFT usando sus propiedades y, con ello, reducir drásticamente el número de operaciones y el tiempo invertidos en su cálculo [Cooley and Tukey, 1965].

El cálculo de la DFT tiene una complejidad algorítmica $\mathcal{O}(N^2)$, pero la FFT la reduce a $\mathcal{O}(N \log_2 N)$. Para esta investigación, N representa el número de mediciones en cada registro sísmico con el que se cuenta. Por tanto, la FFT se emplea para obtener una representación espectral de dicha señal en forma semejante a una imagen, la cual se introducirá al modelo de aprendizaje profundo. Esta imagen será generada usando la biblioteca *Obspy* de Python, misma que utiliza la FFT para calcular el espectrograma de la señal.

Antes de crear los espectrogramas es necesario aplicar un filtrado a la señal. Este filtrado y los distintos tipos que existen se describen en la siguiente sección.

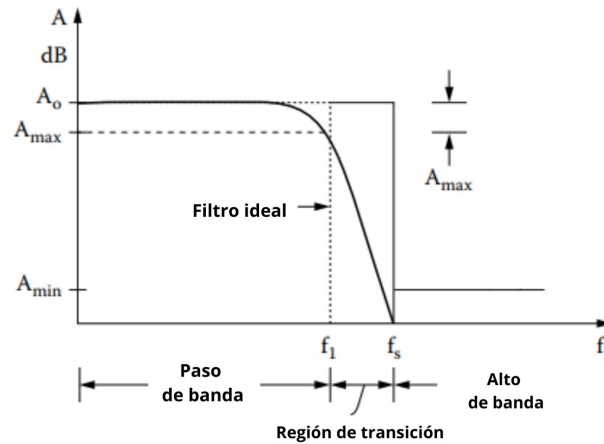


Figura 3.1: Respuesta de frecuencia de filtro pasa bajas.

3.5. Filtrado digital de señales

Un filtro es un circuito diseñado para dejar pasar una banda de frecuencia en específico. Definiremos a la señal como la parte de los datos en la que se está interesado, en el caso de esta tesis son los tipos de ondas sísmicas que se generan cuando ocurre un sismo, mientras que el resto se considerará como ruido. Los filtrados más comunes son:

- **Filtrado pasa bajas:** el primer tipo es el filtro pasa bajas, que como su nombre lo indica deja pasar las señales de baja frecuencia y elimina las frecuencias arriba de una frecuencia de corte. El filtro ideal tiene una forma rectangular, que indica cuando el límite entre el paso de banda y el alto de banda es abrupto y la pendiente es bastante pronunciada (Figura 3.1).

En la Figura 3.1 el filtro pasa bajas permite el paso de banda de la potencia de la señal A_0 [dB] hasta una marca de corte f_1 [Hz]. Este filtro tendrá un alto de banda definido por un valor mínimo A_{min} y un valor máximo A_{max} al que se sacará una frecuencia de corte dentro del periodo f_1 a f_s (región de transición) que es cuando la señal cambia la frecuencia de A_0 a A_{max} y, a partir de ahí, comienza a decrecer.

- **Filtro pasa altas:** el filtro inverso al pasa bajas es el filtro pasa altas, el cual regula las componentes fuera de la banda de frecuencias (Figura 3.2).

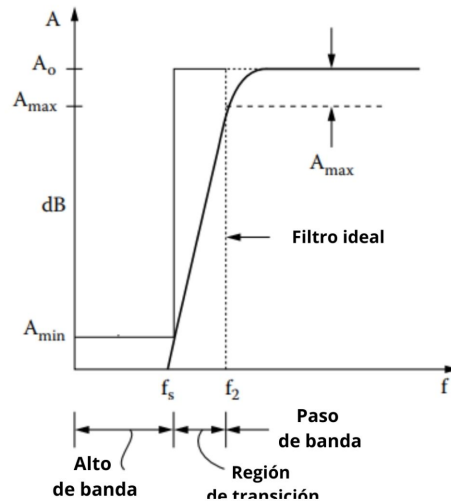


Figura 3.2: Respuesta de frecuencia de filtro pasa altas.

En la Figura 3.2 el filtro pasa altas tiene un alto de banda definido por un valor mínimo A_{min} y un valor máximo A_{max} , el cual recibe una señal A_0 [dB] al que se sacará una frecuencia de corte dentro del periodo f_s a f_2 (región de transición). Dentro de este periodo el valor de la señal $A_0 \geq A_{min}$ se irá incrementando (en el caso del filtrado ideal). El paso de banda dejará pasar las frecuencias cuando en f_2 ocurra que $A_0 \geq A_{max}$.

En este trabajo de tesis se utiliza un filtrado pasa altas a $2Hz$ de frecuencia, debido a que en esa frecuencia se concentra la mayor parte de las componentes de frecuencia de los sismos de interés. Por otra parte, también reduce el efecto producido por las ondas de superficie provenientes de eventos lejanos de mayor magnitud. Las frecuencias mayores a los $8Hz$ están asociadas principalmente al ruido urbano proveniente del paso de vehículos y personas, por lo que también se descartan.

Las señales que se registran a través de los diferentes tipos de sismómetros se pueden representar de distintas formas, tales como los llamados **sismogramas**¹ o **espectrogramas**, entre otros. Para fines de este trabajo solamente se detallan los espectrogramas en la siguiente sección.

¹Un sismograma es una representación bidimensional del tiempo (eje horizontal) con respecto a la amplitud de la onda (eje vertical)

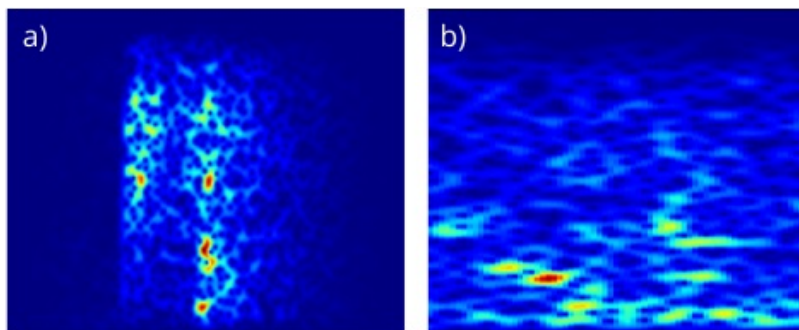


Figura 3.3: Ejemplos de **espectrogramas** generados a partir de la señal sísmica obtenida de la estación PNIG en Pinotepa Nacional, Oaxaca. a) Representación bidimensional de las ondas P y S de un evento sísmico y b) Señal proveniente de diversas fuentes (ruido).

3.5.1. Espectrograma

El espectrograma consiste en el cálculo de las componentes espectrales de una señal en función del tiempo, representadas como una gráfica tridimensional, donde en la primera dimensión se tiene el dominio del tiempo, en la segunda la frecuencia en Hz. y, en la tercera (el color), la potencia (Figura 3.3). Los espectrogramas se pueden representar como una imagen en RGB de 2 dimensiones manteniendo la tercera dimensión en las distintas tonalidades que se presentan en los píxeles que la componen.

En un espectrograma se puede observar el tiempo de arribo de las ondas sísmicas así como su contenido espectral.

Los espectrogramas permiten diferenciar las distintas señales que registra un sísmómetro, tales como:

- Ruido producido por el paso de vehículos
- Señales provenientes de la atmósfera (huracanes, tormentas, oleaje, vientos, etc.)

A partir de la generación de los espectrogramas de la señal, se pueden aplicar diversos métodos de aprendizaje automático para realizar la detección de sismos, pero antes de aplicar estas técnicas es preciso definir algunos de sus conceptos básicos, lo cual se llevará a cabo en el siguiente capítulo.

Capítulo 4

Aprendizaje profundo

En este capítulo se mencionan los distintos conceptos que existen en el campo del aprendizaje automático. Desde los conceptos más simples como lo es el aprendizaje profundo y los paradigmas de aprendizaje hasta los algoritmos más utilizados. En particular los algoritmos o modelos de redes neuronales artificiales permiten resolver la tarea de detección de objetos con una gran fiabilidad, por lo que se utilizará un algoritmo que hace uso de estos modelos.

4.1. Aprendizaje automático

Se han desarrollado muchos trabajos utilizando aprendizaje automático, pero no fue sino hasta el año de 1990, con el filtrado de *spam*, que comenzó a ser **ampliamente** conocido y estudiado este campo, ya que ayudó a miles de usuarios a no recibir más basura en su bandeja de entrada [Géron, 2019]. Con el aprendizaje automático se han obtenido grandes avances en todo tipo de aplicaciones que han logrado conquistar la industria tecnológica, como el reconocimiento facial en los teléfonos inteligentes [Sipe et al., 2013], la recomendación de contenido (videos, música, etc.) [Bennett et al., 2007] e, inclusive, venciendo al campeón mundial de Go [Silver et al., 2016].

El término aprendizaje automático -*Machine Learning (ML)*- es una rama de la Inteligencia Artificial (IA) que busca, a través de un conjunto de algoritmos y técnicas

basadas en modelos matemáticos, aprender a partir de datos para resolver una tarea en específico [Gulli and Pal, 2017].

Otras definiciones más generales son:

1. El ML es una ciencia de programar computadoras que puedan aprender a partir de datos [Géron, 2019].
2. El ML es el campo de estudio que da a las computadoras la capacidad de aprender sin ser programadas explícitamente [Samuel, 1959].

Se entiende como “aprendizaje” la generalización del conocimiento a partir de un conjunto de experiencias. El aprendizaje se puede aplicar a diversos algoritmos donde a partir de ejemplos o experiencias (datos) se logra generalizar el conocimiento adquirido para un ejemplo (dato) nunca antes visto. El aprendizaje automático se ejemplifica en la Figura 4.1.

De manera general, en el aprendizaje automático se seleccionan uno o varios modelos candidatos, los cuales se deben adaptar a la naturaleza de los datos, para luego ser entrenados. Una vez entrenados los modelos, se evalúan con otros datos que no se hayan visto con anterioridad, de esta forma se asegura que el modelo sea capaz de **generalizar** y no solamente aprender ejemplos, reglas o patrones.

Existen varios paradigmas de aprendizaje automático que se detallan en la siguiente sección.

4.1.1. Tipos de aprendizaje

El aprendizaje automático presenta diferentes tipos, mismos que se clasifican en categorías las cuales, a su vez, se dividen en diversos paradigmas de aprendizaje siguiendo los criterios siguientes:

1. Si pueden o no ser entrenados con supervisión humana:
 - a) **Aprendizaje supervisado**
 - b) Aprendizaje no supervisado

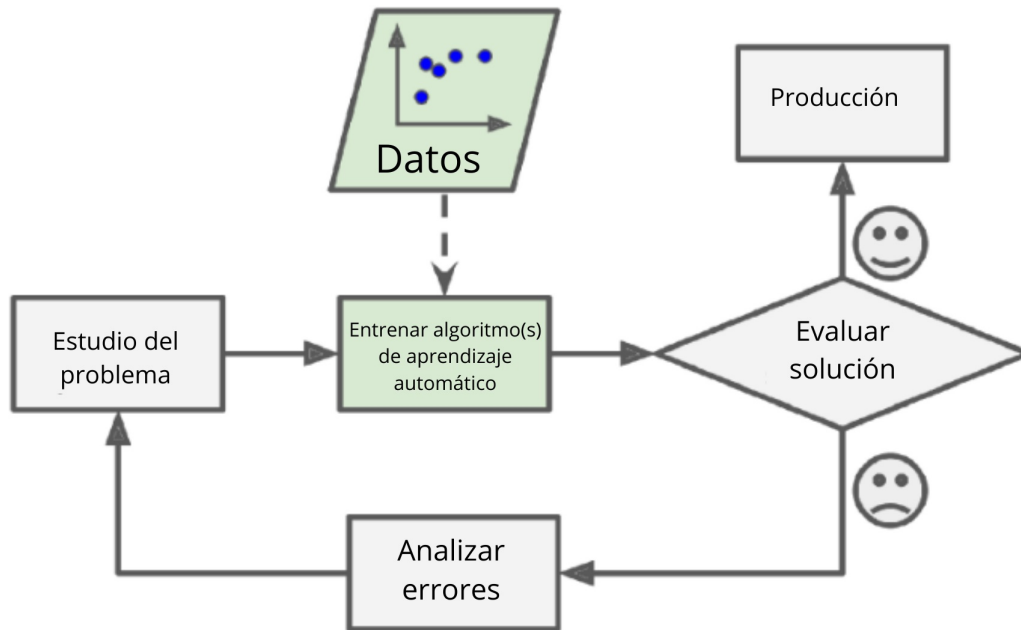


Figura 4.1: Aproximación del proceso de aprendizaje automático, tomado de [Géron, 2019].

- c) Aprendizaje semi supervisado
- d) Aprendizaje por refuerzo

2. Si pueden o no aprender de forma incremental sobre la marcha:

- a) Aprendizaje por lotes
- b) Aprendizaje en línea

Para los fines de este trabajo, solamente se detallará a fondo el aprendizaje supervisado, el cual se describe a continuación.

4.1.2. Aprendizaje supervisado

El aprendizaje supervisado, al igual que los demás tipos de aprendizajes, tiene un conjunto de datos con instancias (elementos o entes), pero con una diferencia importante: **la etiqueta** (la clase a la que pertenece la instancia). A esto se le conoce como datos etiquetados, proceso generalmente realizado por una persona, los cuales se dividen en dos conjuntos principales: el conjunto de entrenamiento y el conjunto

Conjunto de entrenamiento

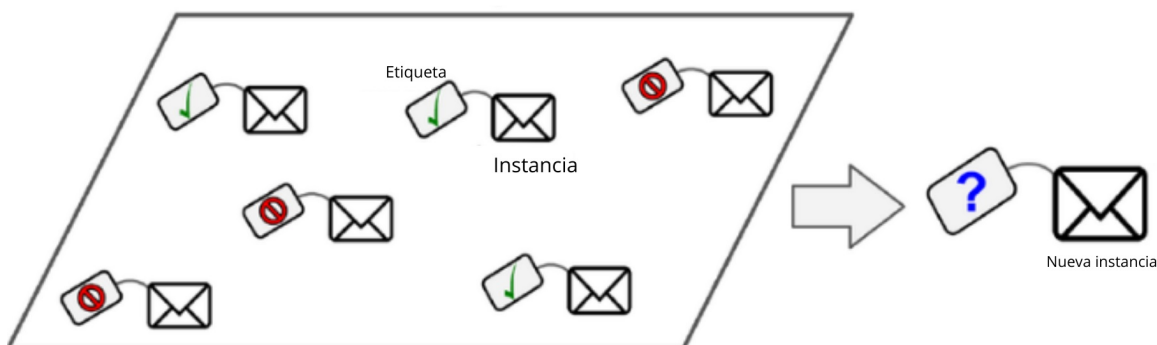


Figura 4.2: Ejemplo de un conjunto de datos de entrenamiento etiquetados, utilizados para clasificación de spam, tomado de [Géron, 2019].

de prueba. De esta manera, al aplicar el modelo a los datos de entrenamiento, se espera que este logre aprender y cuando se aplique para clasificar una instancia del conjunto de prueba (al cual nunca tuvo acceso) se observe si el modelo realmente está generalizando el conocimiento adquirido (aprendiendo) o solamente memorizó los ejemplos (Figura 4.2).

El aprendizaje supervisado es de los más utilizados en el campo del aprendizaje automático debido a sus múltiples aplicaciones que van desde la creación de un *chatbot* [Roller et al., 2020], la clasificación de imágenes [Krizhevsky et al., 2012], el procesamiento del lenguaje natural [Brown et al., 2020] y hasta la detección de objetos en tiempo real [Redmon and Farhadi, 2017]. Las aplicaciones mencionadas han ayudado a obtener avances en campos como la robótica, la medicina y la semántica, entre muchos otros.

Entre las tareas más importantes del aprendizaje supervisado están:

- Tareas de regresión
- Tareas de clasificación y
- **Tareas de visión por computadora**

Los algoritmos que se utilizan para resolver problemas específicos relacionados con las tareas citadas son muchos y con resultados que varían, pero las "Redes Neuronales

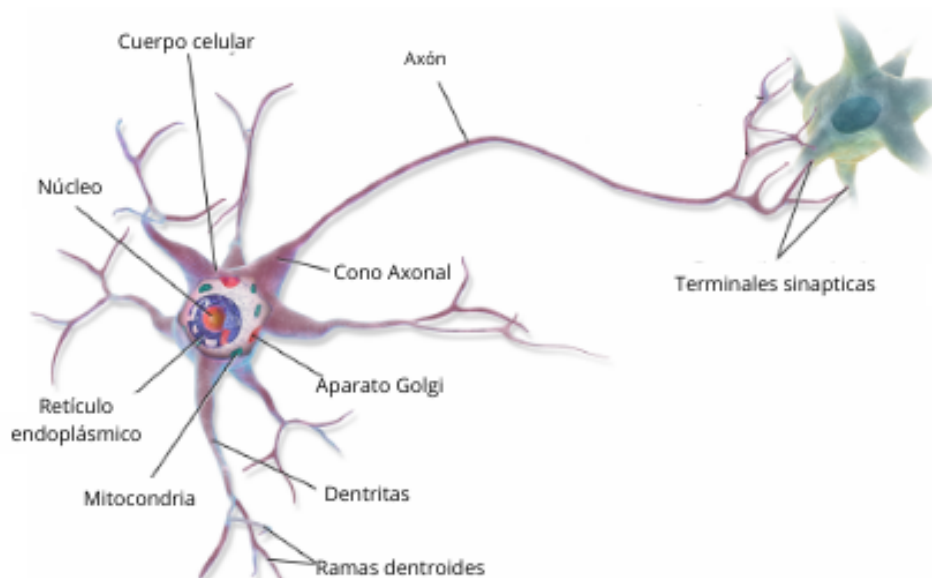


Figura 4.3: Diagrama de una neurona biológica, disponible en www.wikipedia.org/neuron

Artificiales” (RNAs) son un conjunto de algoritmos que en particular ha destacado por los avances obtenidos en los últimos años y por su aplicación no solamente en tareas de clasificación o regresión. Este tema se tratará en el apartado siguiente.

4.2. Redes Neuronales Artificiales (RNAs)

Las RNAs conforman un conjunto de algoritmos de ML inspirados en las redes de neuronas biológicas del cerebro humano. El modelo biológico de las redes neuronales está compuesto por un tejido nervioso formado principalmente por las neuronas biológicas (Figura 4.3) y por células de la glía (neuroglías), que se distribuyen como redes por todo el organismo. Las neuronas tienen la misión de transmitir los impulsos nerviosos a todas partes del cuerpo, en cambio, las células de la glía cumplen funciones de sostén, de nutrición y de defensa de las células nerviosas.

El primer modelo matemático de una RNA era bastante sencillo y resolvía solamente problemas de lógica proposicional¹. Fue propuesto en 1943 por el neurofisiólogo *Warren McCulloch* y el matemático *Walter Pitts* en [McCulloch and Pitts, 1943].

¹Sistema basado en proposiciones (afirmaciones) y operaciones usando conectivos lógicos [Gurovich, 2008]

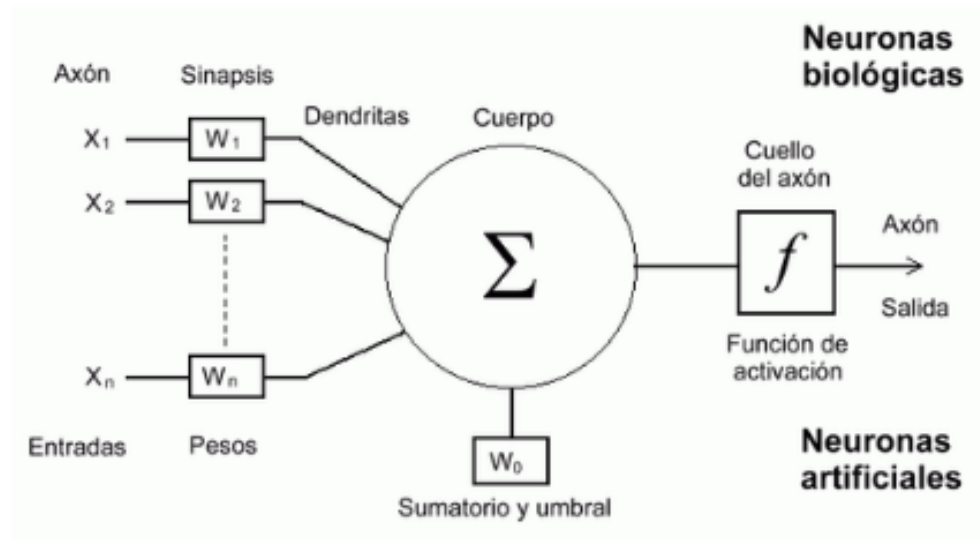


Figura 4.4: Neurona artificial tomada del modelo de Rumelhart y McClelland, tomado de [McClelland et al., 1986].

Años después, *Frank Rosenblatt* y otros desarrollaron una clase diferente de RNA denominada “Perceptrón” [Rosenblatt, 1958], el cual resolvía algunos problemas de reconocimiento de patrones.

A partir de este modelo se han desarrollado otros que impulsaron esta área dentro del aprendizaje automático y que continúan haciéndolo.

El modelo de una RNA se puede generalizar a los elementos siguientes (Figura 4.4):

- Un vector X con n entradas.
- Una matriz de pesos sinápticos w_{ij} , que representa la influencia de la neurona presináptica² i con respecto a la neurona postsináptica³ j .
- Una función de propagación $h_j = d(w_{ij}, x_i(t))$, la cual proporciona el potencial postsináptico.
- Una función de activación $a_j(t) = f(a_j(t-1), h_j(t))$ (Figura 4.5), que permite estimular a la neurona para emitir o no un valor de salida.

²Neurona anterior a la que esta conectada la neurona sinaptica

³Neurona posterior a la sisnapsis

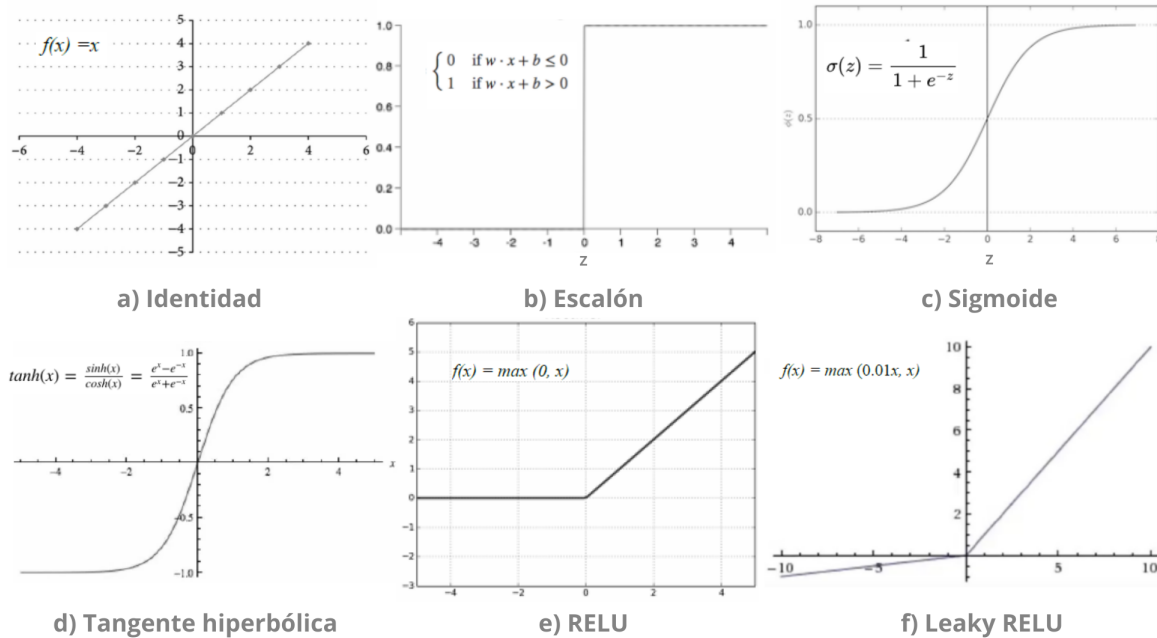


Figura 4.5: Funciones de activación más comunes.

Las funciones de activación permiten añadir la propiedad de no linealidad a las neuronas, lo que las hace más robustas al resolver problemas no lineales. Si no se le aplicara esta función, la neurona emitiría un valor dado por combinaciones lineales (suma ponderada) a las neuronas postsinápticas con las que esté conectada y estará limitada a resolver únicamente problemas lineales.

- Una función de salida $F_j = y_j(a_j(t))$, que proporciona la salida en función del estado de activación.

Aunado a lo anterior, la arquitectura de una red de neuronas es la estructura o patrón de conexiones que existen en la red. Esta arquitectura se compone de capas en las que se agrupan las neuronas. Las capas se denominan de acuerdo con la función que realizan:

- **Capa de entrada:** recibe datos o señales provenientes del entorno.
- **Capa oculta:** se emplea para el procesamiento de la red y generalmente se compone de más de una capa.

- **Capa de salida:** es la respuesta de la red a los estímulos (operaciones) de la entrada.

Cuando se tiene un gran número de capas se le denomina red neuronal *profunda*, lo cual origina el término *aprendizaje profundo*. El concepto se retomará en la siguiente sección.

En los últimos años las RNAs han tenido un avance muy importante debido principalmente a tres factores:

1. La gran cantidad de datos disponibles para su entrenamiento.
2. El poder de cómputo que se ha incrementado desde 1990 sin parar y que ahora hace posible entrenar redes neuronales cada vez más profundas en un tiempo razonable. Las unidades de procesamiento (CPU y GPU) tienen mayor capacidad de cálculo, e inclusive, se han desarrollado algunas unidades especializadas en las operaciones requeridas para entrenar redes neuronales, las denominadas *-Tensor Processing Units (TPU)-*.
3. Los algoritmos para entrenar redes neuronales han tenido una continuidad de mejoras.

El aprendizaje profundo ha tenido un gran impacto en muchas de las áreas de la investigación, entre ellas la sismología, la cual se aborda en este trabajo de tesis y por lo cual se detalla el aprendizaje profundo en el apartado siguiente.

4.3. Aprendizaje profundo

En el campo de la IA los términos “aprendizaje profundo” y “aprendizaje automático” se llegan a confundir continuamente, pero no son sinónimos. Para ilustrar la diferencia, en la Figura 4.6 se presenta un diagrama de Venn sobre estos conceptos.

El aprendizaje profundo es un subconjunto del aprendizaje automático. El término **profundo** hace referencia a un conjunto de modelos de RNAs que se caracterizan por tener un número significativo de capas de neuronas. Hasta hace varios años el término

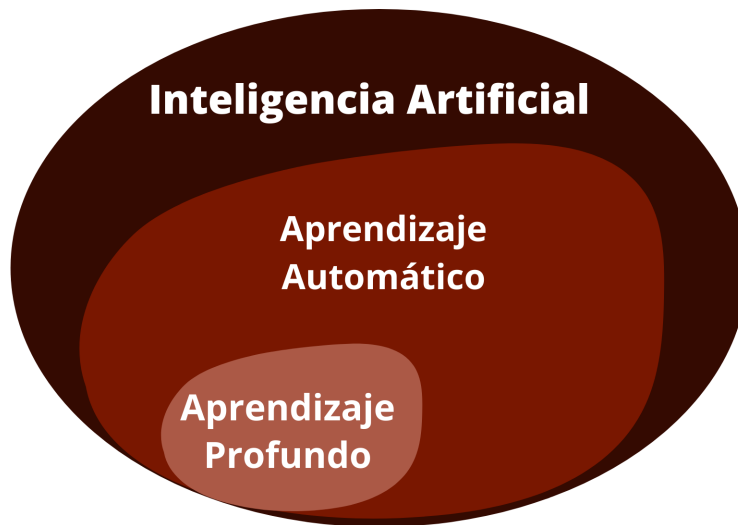


Figura 4.6: Diferencia entre inteligencia artificial, aprendizaje automático y aprendizaje profundo.

hacía referencia a una arquitectura de una red neuronal de 3 a 5 capas. Actualmente se habla de que una red es profunda cuando presenta una arquitectura con cientos de capas [Gulli and Pal, 2017].

El aprendizaje profundo es sin duda la técnica más popular dentro del campo del aprendizaje automático y es el estado del arte en muchas áreas, como el procesamiento del lenguaje natural -*Natural Language Processing (NLP)*- [Brown et al., 2020], en tareas de visión por computadora como reconocimiento facial [Sipe et al., 2013], etc.

En el presente trabajo el aprendizaje profundo aplicado a tareas de visión por computadora es especialmente importante, debido a que se utilizarán modelos como las redes convolucionales para resolver un problema de detección de sismos. Para ello es conveniente retomar la inspiración biológica que originó el desarrollo de estos modelos de RNAs: la visión humana.

4.4. Arquitectura de la corteza visual

David H. Hubel y Torsten Wiesel mostraron que algunas neuronas en la corteza visual tienen un campo receptivo local que sólo reacciona a estímulos visuales localizados y limitados por regiones, de todo el campo visual [Hubel, 1959, Hubel and Wiesel, 1968].

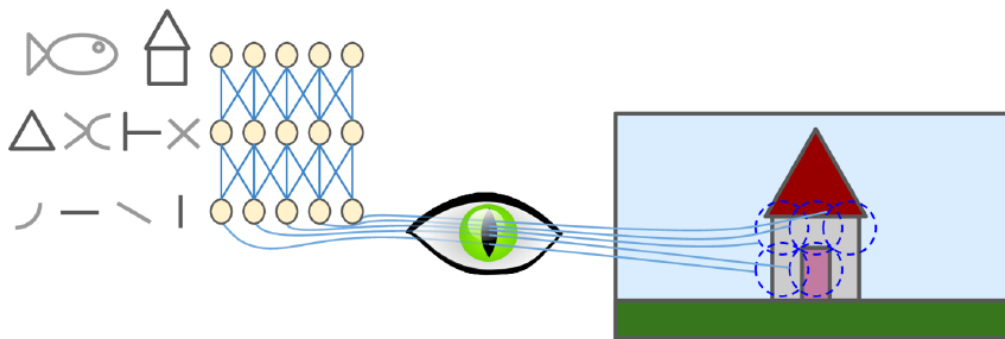


Figura 4.7: A medida que la señal visual se abre paso a través de las capas de neuronas cerebrales consecutivas, las neuronas responden a patrones más complejos en campos receptivos más grandes, tomado de [Géron, 2019].

Llevaron a cabo sus experimentos en gatos y luego en monos, dando información crucial sobre la estructura de la corteza visual humana y siendo galardonados con el premio Nobel de medicina en 1981.

Además de que *Hubel y Wiesel* mostraron que algunas neuronas reaccionan únicamente a imágenes con líneas horizontales, mientras que otras, reaccionan a líneas con diferentes orientaciones. Lo que ayudó a observar que algunas neuronas tenían campos receptivos más grandes y reaccionaban a patrones cada vez más complejos, que son combinaciones de patrones obtenidos de las neuronas vecinas de niveles más bajos (Figura 4.7). Con base en lo anterior, los autores mostraron que esta arquitectura es capaz de detectar todo tipo de patrones complejos en cualquier área del campo visual [Géron, 2019].

Los trabajos de *Hubel y Wiesel* inspiraron el *Neocognitrón*⁴, el cual es una herramienta automática para reconocimiento óptico de caracteres. Esta herramienta introdujo y es la base de las técnicas más actuales de las redes de convolución. Sin embargo, no fue sino hasta 1988 que Yann LeCun y otros autores consolidaron el término “Redes neuronales convolucionales” -*Convolutional Neural Networks (CNNs)*-, presentando los bloques de capas de convolución y las capas de *pooling* en la arquitectura LeNet-5 [LeCun et al., 1989], que son las que caracterizan a este tipo de redes.

⁴Kunihiko Fukushima, “Neocognitron: A Self-Organizing Neural Network Model for a Mechanism of Pattern Recognition Unaffected by Shift in Position,” *Biological Cybernetics* 36 (1980): 193–202.

4.5. Capas convolucionales

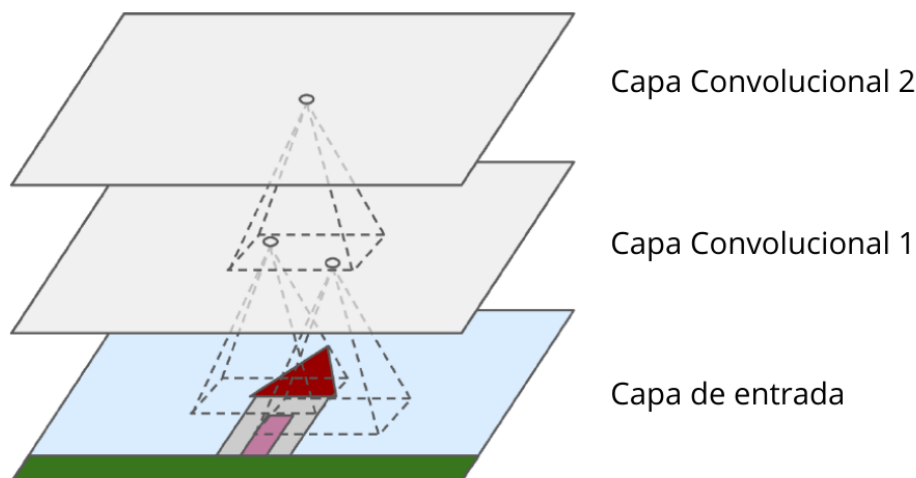


Figura 4.8: Capas de una CNN con sus correspondientes campos receptivos locales, tomado de [Géron, 2019].

Una convolución es una operación matemática que consiste en deslizar una función sobre otra midiendo la integral de su producto punto. La convolución está muy relacionada con la Transformada de Fourier y la Transformada de Laplace y es muy utilizada en el procesamiento de señales [Géron, 2019]. Las capas convolucionales de las que se habla en este trabajo usan correlaciones cruzadas, que son muy similares a la convolución, lo que da su nombre.

Las neuronas de la primera capa de convolución no están conectadas a cada píxel de la imagen de entrada, sino solamente a los píxeles de sus campos receptivos locales, Figura 4.8. Las neuronas de una capa convolutiva aplican una función de convolución sobre los datos que reciben y el resultado lo pasan a una neurona vecina de la capa convolutiva siguiente. Esta ensambla los datos como parte de su propio campo receptivo local y realiza la misma operación.

La característica de las CNNs que les da su gran poder de discriminación proviene del hecho de que cada capa realiza una abstracción del conocimiento que las neuronas de su campo receptivo aprenden, llegando a obtener un conocimiento generalizado sobre los datos con los cuales son entrenadas. Por ejemplo, tratándose de

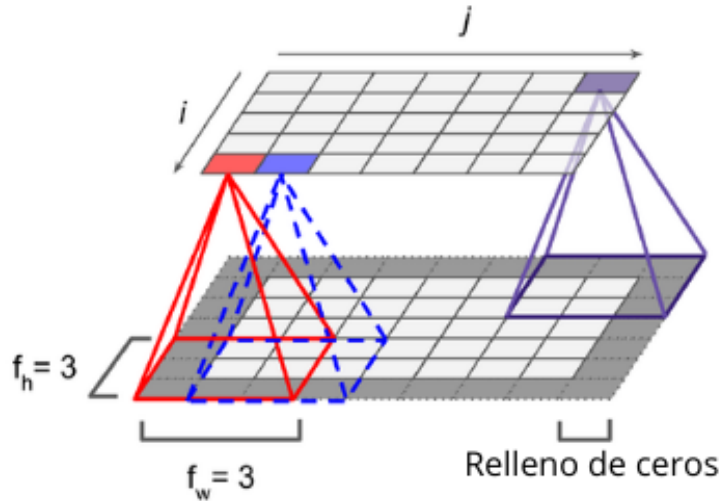


Figura 4.9: Capas de una CNN con sus campos receptivos locales, tomado de [Géron, 2019].

clasificar objetos en imágenes, la primera capa convolucional recibe píxeles, de los cuales aprende rasgos (*características*) simples (líneas, puntos, colores, etc.); de estos, las siguientes capas aprenden *características* más complejas (esquinas, curvas, espirales, composición de colores, etc.); y así en adelante, hasta que las últimas capas convolucionales aprenden *características* específicas que permiten la clasificación (rostros, brazos, alas, ruedas, picos, etc.) [Géron, 2019].

Cada neurona en una capa de convolución se conecta a $f_h \times f_w$ neuronas de la capa anterior (alto y ancho del campo receptivo) y se lleva a cabo un *padding* o relleno alrededor de la entrada para mantener una matriz cuadrada de $n \times n$ (imagen). En la Figura 4.9 se observa que la neurona del renglón i , columna j de la capa superior está conectada a las salidas de las neuronas en la capa anterior, localizadas de i a $i + f_h - 1$ renglones y de j a $j + f_w - 1$ columnas.

Se agrega a la entrada un *padding* de ceros cuando se requiere mantener las mismas dimensiones de la entrada en la salida, dado que la operación de convolución las reduce. El relleno no se realiza si las dimensiones entre la entrada y la salida pueden variar.

Para reducir la complejidad computacional del modelo es posible conectar una capa de entrada con muchos píxeles (para el caso de una imagen) a una capa convo-

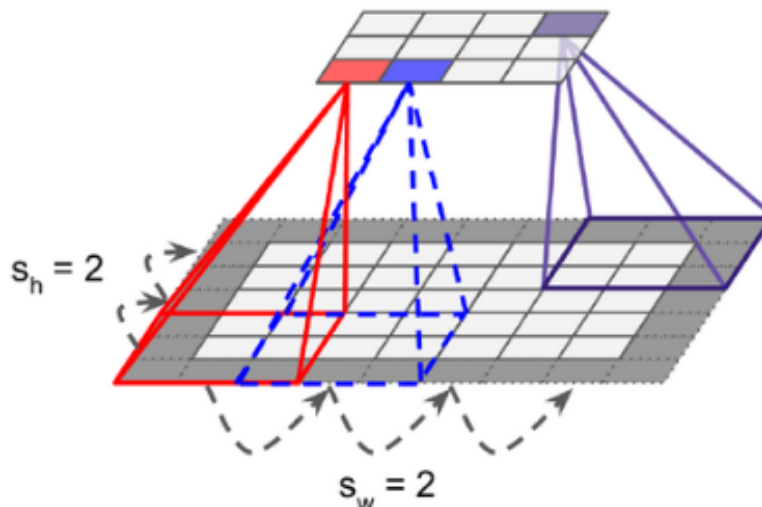


Figura 4.10: Reducción de dimensión de una imagen (capa de entrada) de 5×7 con padding de ceros (7×9), conectada a una capa más pequeña de 3×4 , usando un tamaño del campo receptivo local de $f_h = 3 \times f_w = 3$ y un *stride* de 2, [Géron, 2019]. Las medidas se consideran en píxeles.

lucional mucho más pequeña, aplicando cambios de un campo receptivo (la altura s_h y la anchura s_w) a otro, a través de *strides* o saltos. Véase la Figura 4.10 y compárese con la Figura 4.9.

La operación de convolución se realizará entre la imagen de entrada y las *características* (imágenes) que se desean detectar, los cuales actúan como *filtros* de lo que se busca. En la próxima sección se explican más a fondo estos filtros, los cuales corresponden a los pesos de la capa y se comparten para todas las neuronas que buscan una misma *característica*.

4.5.1. Filtros

Los pesos de una neurona pueden ser representados como una pequeña imagen del tamaño del campo receptivo local, mismos que se definen como parámetros de la red, y sirven como **filtros** para buscar características. Más aún, los filtros se combinan para ubicar características más complejas. De esta forma, cada capa convolucional de la red presenta múltiples filtros y produce un *mapa de características* por filtro. Un ejemplo de mapas de características se representa en la Figura 4.11.

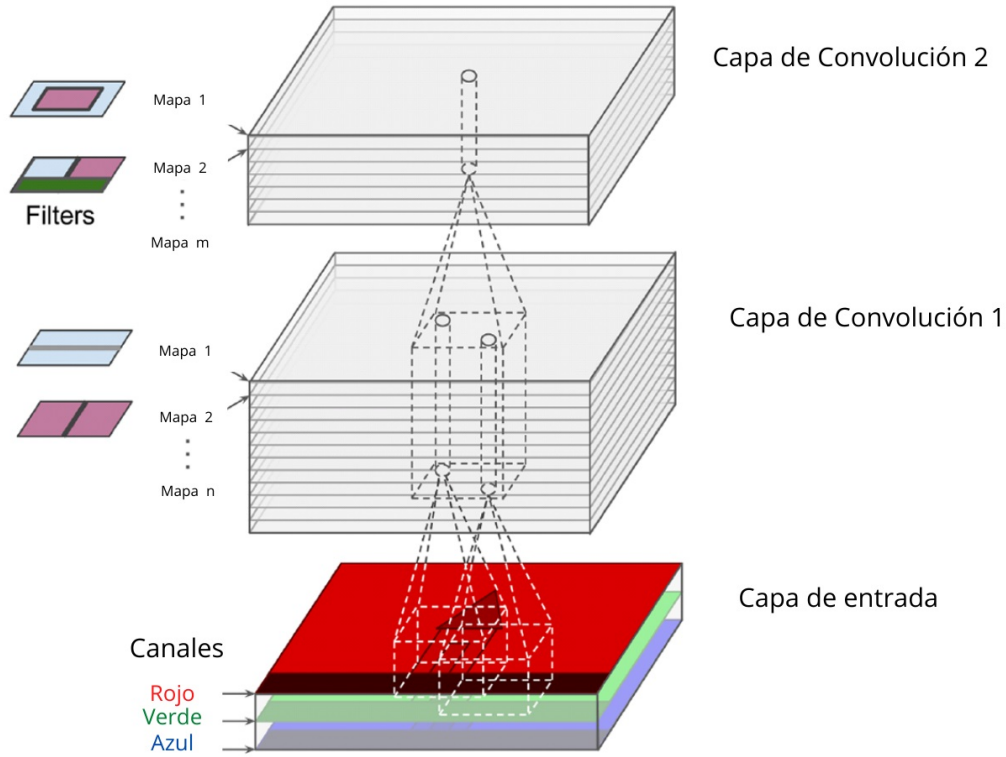


Figura 4.11: Mapas de *características* de una CNN, tomado de [Géron, 2019]

Cabe mencionar que para cada mapa de características los parámetros son distintos, pero las neuronas que generan el mismo mapa comparten los parámetros. Este concepto de *mismos pesos por mapa*, son un concepto muy importante, ya que con esto la CNN obtiene invariancia espacial, además de que se reduce el número de parámetros global en el modelo, a diferencia de las redes neuronales tradicionales. La invariancia espacial le permite a una CNN ser capaz de detectar las características buscadas en cualquier posición donde estos se encuentren en la entrada analizada.

El cálculo de la salida de una neurona dada, en una capa convolucional, se determina con la Ecuación siguiente:

$$z_{i,j,k} = b_k + \sum_{u=0}^{f_h-1} \sum_{v=0}^{f_w-1} \sum_{k'=0}^{f_n-1} x_{i',j',k'} \times w_{u,v,k',k}, \text{ con } \begin{cases} i' = i \times s_h + u \\ j' = j \times s_w + v \end{cases} \quad (4.1)$$

- $z_{i,j,k}$ es la salida de la neurona ubicada en el renglón i , columna j y mapa de

características k de la capa convolucional l .

- s_h y s_w son los saltos verticales (h) y horizontales (w), f_h y f_w corresponde a la altura y anchura del campo receptivo local y f'_n es el número de mapas de características en la capa anterior $l - 1$.
- $x_{i',j',k'}$ es la salida de la neurona ubicada en la capa anterior $l - 1$, renglón i' , columna j' y mapa de características k' .
- b_k es el término de sesgo para el mapa de características k en la capa l .
- $w_{u,v,k',k}$ es la conexión de pesos dentro de cada neurona en el mapa de características k de la capa l y su entrada ubicada en el renglón u , columna v y el mapa de características k' .

A continuación, se detalla el segundo tipo de capa más común usado en una arquitectura de CNNs.

4.6. Capas de *Pooling*

El objetivo de la capa de *pooling* es reducir el tamaño de los mapas de características. Como sucede en las capas de convolución, cada neurona en la capa de *pooling* está conectada a las salidas de un número limitado de neuronas en la capa anterior, ubicadas dentro de un pequeño campo receptivo local (de tamaño s_h y s_w , correspondientes a la altura y al ancho, respectivamente). Este tipo de capas también pueden considerar saltos y relleno, en caso de requerirse. Además, una capa de *pooling* no tiene pesos asociados, ya que solamente aplica a su entrada una función de agregación, es decir, toma la media o el valor máximo para reducir la imagen. Este proceso ayuda a reducir la carga computacional, el uso de memoria y disminuye el número de parámetros de la CNN.

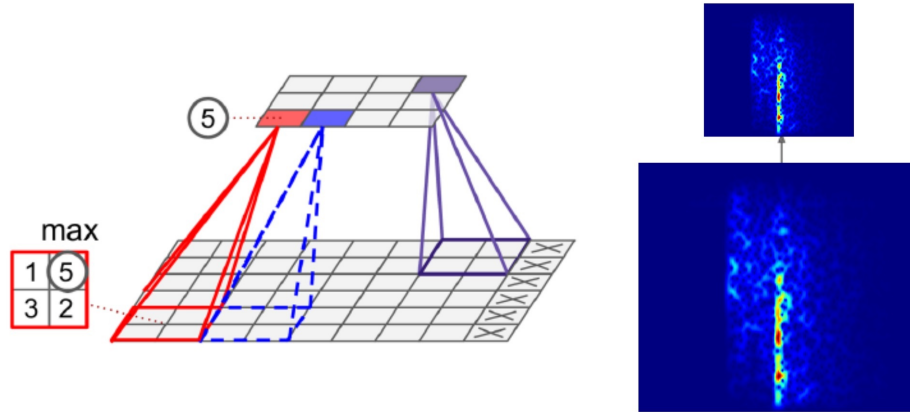


Figura 4.12: Capa de Pooling que reduce el mapa de características (imagen) de 8×6 a un nuevo mapa de tamaño 4×3 tomando el valor máximo (con $s_h, s_w = 2$, salto de 2 y sin padding), modificado de [Géron, 2019]. Los tamaños se consideran en píxeles.

4.7. Funciones de error

Las funciones de error también son conocidas como funciones de costo o de pérdida y se usan para evaluar la predicción del modelo. En específico, calcular esta función da una idea de qué tan lejos la predicción está de la respuesta correcta (la clase).

Calcular el error se puede ver como un problema de optimización, dado que lo que se busca es minimizar la función de error. En la siguiente sección se habla sobre optimizadores a mayor detalle, por ahora solo se dejará la idea. El objetivo es intentar encontrar las variables (pesos en las neuronas) que logren minimizar la función de error tanto como se requiera. A este proceso se le llama optimización del error.

Existen diferentes funciones de error para evaluar una misma predicción, y este tiene un efecto considerable en el rendimiento del modelo. En esta tesis solo se describirá la más común para problemas de clasificación: Entropía cruzada.

La Entropía cruzada cuantifica la diferencia entre dos distribuciones de probabilidad P . Por ejemplo, para el caso de dos clases C_1 (sismos) y C_2 (ruido), se tiene la distribución correcta de una instancia (ejemplo) de entrenamiento: $P_{C_1} = 100.0$ y $P_{C_2} = 0.0$

Esta distribución de probabilidad se puede interpretar de la siguiente manera. La clase C_1 tiene 100 % de probabilidad de ser sismo, mientras que la clase C_2 , tiene 0 % de probabilidad ser sismo.

Si se tiene un algoritmo de aprendizaje profundo que predice, por ejemplo la siguiente distribución de probabilidad: $P_{C_1} = 75.8$ y $P_{C_2} = 24.2$, se observa que la P_{C_1} (sismo) es más alta que la de P_{C_2} y para determinar que tan certera es la distribución predicha con respecto a la distribución real, se aplica la ecuación general de la Entropía Cruzada para m clases, dada por:

$$E(W, b) = - \sum_{i=1}^m y_i \log(P_i)$$

Donde y es la probabilidad verdadera de la clase C_i y P la probabilidad predicha i .

Para el caso en que se tiene únicamente 2 clases, se le conoce a esta función como **Entropía cruzada binaria** y su ecuación está dada por:

$$E(W, b) = -(y_{C_1} \log(P_{C_1}) + y_{C_2} \log(P_{C_2}))$$

En el mejor de los casos la entropía E vale 0 o es cercana a 0. Para calcular el error de la entropía cruzada binaria para todas las instancias (n) del conjunto de entrenamiento, se usa la fórmula general:

$$E(W, b) = - \sum_{i=1}^n \sum_{j=1}^{m=2} y_{i,j} \log(P_{i,j}) \quad (4.2)$$

Para saber que la red neuronal está aprendiendo, la función de error debe decrecer en cada iteración y mantenerse cercana o igual a 0. En el caso ideal se observa que mientras la función de pérdida comienza a decrecer y es más cercana a 0, la exactitud comenzara a aumentar y se mantendrá cercana a 1.

4.8. Optimizadores y tasa de aprendizaje

Durante el entrenamiento de las RNAs se le muestran miles de ejemplos (conjunto de entrenamiento) de tal manera que, al finalizar el proceso, la red es capaz de hacer predicciones. De este modo se puede hacer la comparación con las etiquetas o clases

correctas para calcular **el error**. Con la finalidad de ir disminuyendo **el error**, se va ajustando de manera iterativa los pesos de las neuronas de manera automática. Para lograr encontrar estos pesos que logren disminuir el error y aumentar la exactitud del modelo, se usan algoritmos de optimización. Optimización es la manera de encontrar el valor máximo o mínimo de una función.

4.8.1. Descenso de gradiente

El gradiente es una función que obtiene la pendiente o la tasa de cambio de la tangente a la curva en un punto dado (ver Figura 4.13).

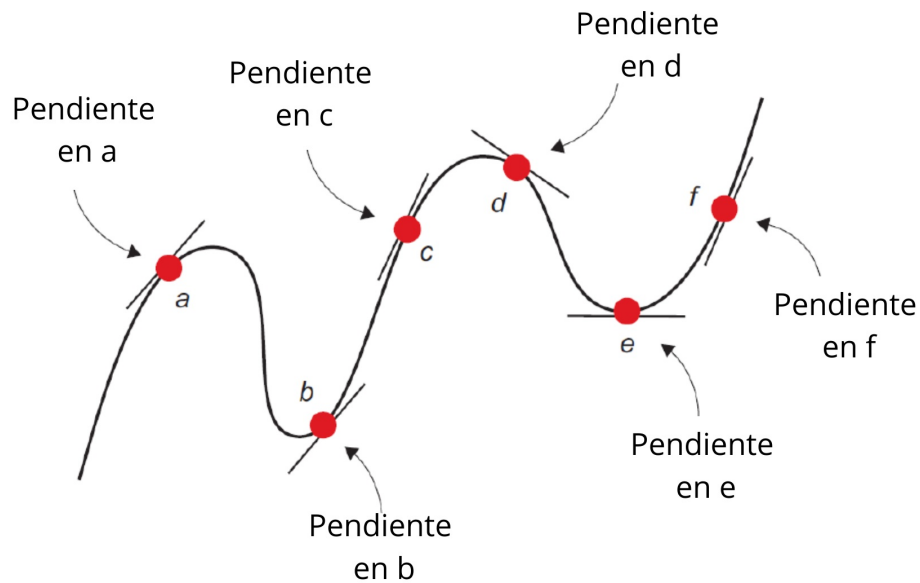


Figura 4.13: Ejemplo de pendientes obtenidos a partir del cálculo de la derivada en los puntos de la función.

Una forma sencilla de comprender como funciona el descenso de gradiente es pensar en un lugar donde haya muchas montañas y valles, cómo en la Figura 4.14, donde el objetivo es bajar hasta el punto más bajo. Imaginemos que se comienza a descender desde un punto aleatorio A, para saber en qué dirección y cuanto caminar es necesario definir lo siguiente:

- La dirección de paso (gradiente)
- El tamaño del paso (*Learning rate*)

La dirección estará dada por el gradiente y el tamaño del paso denotado por la letra α . α significa denota la tasa de aprendizaje que la red dará para descender (**el error**) de la montaña. El valor de α se considera el metaparámetro más importante debido a que si no se determina bien, podría ser caótico el aprendizaje.

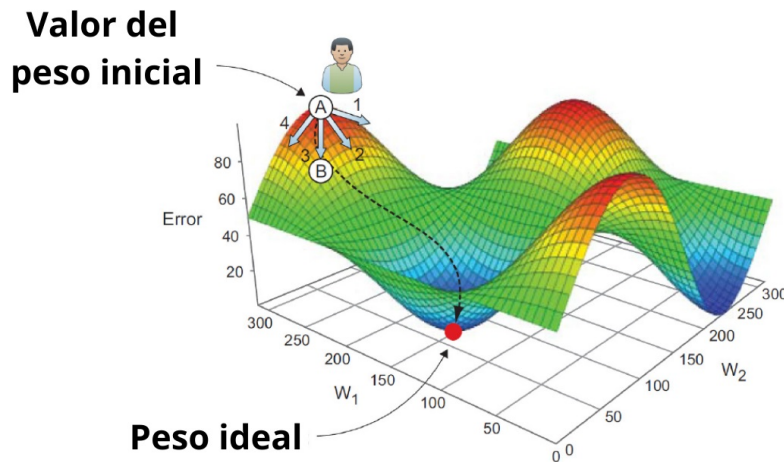


Figura 4.14: Descenso de gradiente para una gráfica de dos dimensiones (2D)

En la Figura 4.15 se observan los dos posibles casos que ocurren cuando se intenta resolver un problema optimización. En el inciso a) para un valor de α grande, significa que la red aprenderá muy rápido, debido a que descenderá con pasos agigantados, mientras que para un valor α pequeño podría tardar demasiado en descender. En el inciso b) pese a encontrar un valor mínimo en la función, no se puede asegurar que se encontró el mínimo global, por lo que caer en mínimos locales es muy factible.

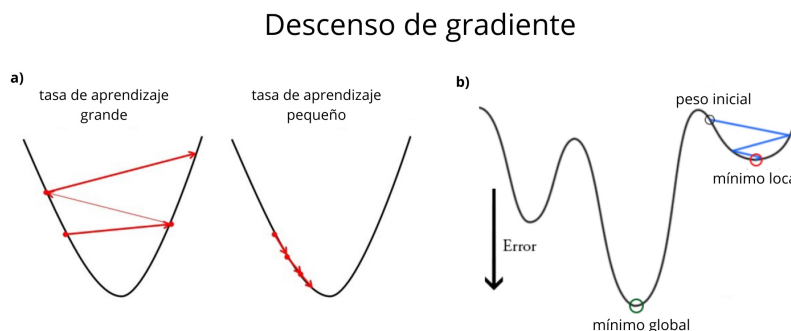


Figura 4.15: En la figura se muestran dos casos que pueden ocurrir cuando se intenta optimizar una función: a) tamaño de la tasa de aprendizaje y b) estancamiento en mínimos locales.

4.8.2. Aprendizaje adaptativo

La estimación del momento de forma adaptativa -*Adaptive Moment estimation (Adam)*- es el optimizador más usado en aprendizaje profundo debido a que permite controlar el descenso de forma dinámica o adaptativa, lo que reduce el tiempo de cálculo [Ruder, 2016]. La idea general de como funciona es imaginarse una bola pesada que inicia en un punto aleatorio de una montaña. Si la bola comenzara a descender, en un principio el impulso hacia abajo será muy veloz, pero conforme se acerca a una planicie sin tanta pendiente (mínimo local), comenzara a dar saltos (el valor de α) más pequeños.

Los metaparámetros que controlan la tasa de aprendizaje son:

- Tasa de aprendizaje α
- término de momento β_1 , un valor común es $\beta = .9$
- Un ϵ , por general del orden de 10^{-8}

Adam hace uso del momento, que es un término que produce una reducción en las oscilaciones que hace la tasa de aprendizaje en la función. En otras palabras, aumenta la velocidad del aprendizaje si la dirección es de forma horizontal y se reduce si es de forma vertical (ver Figura 4.16). Lo anterior logra disminuir el tiempo de convergencia y reducir el número de paso u oscilaciones [Ruder, 2016].

De esta forma el momento es muy útil para tener un mayor control al momento de descender por el gradiente. Esto se logra añadiendo un término de velocidad Δ a la ecuación que actualiza el peso de la neurona.

$$w_{nueva} = w_{anterior} - \alpha * gradiente + \Delta \quad (4.3)$$

Donde el término de velocidad se denota por la letra Δ , que es determinado a partir de calcular la media ponderada de los gradientes anteriores.

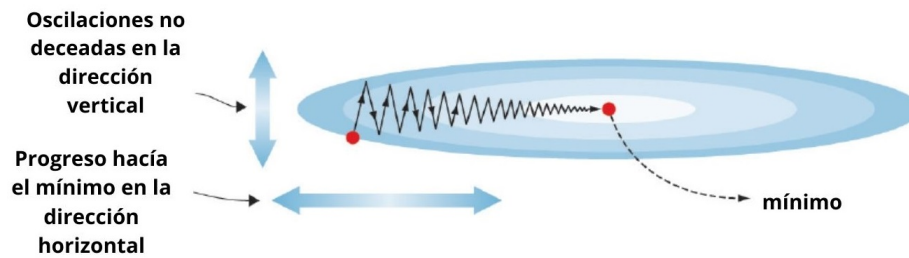


Figura 4.16: Ejemplo de descenso de gradiente adaptativo.

4.9. Parámetros y metaparámetros

Los metaparámetros son aquellos valores dentro de la CNN que definen la estructura de la red, estas variables determinan como la red será entrenada. Los metaparámetros son asignados antes del entrenamiento y se van ajustando hasta obtener los mejores resultados. Ejemplos de metaparámetros son: tasa de aprendizaje, tamaño de lote, número de épocas, número de capas ocultas, entre otros. Estos metaparámetros se definirán en la sección siguiente.

Los parámetros al contrario son variables aprendidas y actualizadas de forma automática durante el entrenamiento. En RNA los parámetros son los **pesos** de las neuronas y se actualizan automáticamente con la finalidad de ir optimizando el error.

4.9.1. Metaparámetros de RNAs

Los algoritmos de RNAs tienen variables que controlan algunos aspectos del comportamiento del modelo. Algunos de estos metaparámetros afectan el costo en memoria y tiempo de ejecución.

Encontrar la mejor configuración de los metaparámetros del modelo no es trivial. El reto es encontrar a prueba y error los valores de los metaparámetros que mejor resuelvan el problema, dado que no existen valores establecidos que te aseguren que funcionen para el problema y los datos con el que se cuenta.

Para entender en que afecta cada uno de los metaparámetros, se dividen en 3 categorías:

1. Arquitectura de la red.

- Número de capas ocultas: estas pueden ser de distintos tipos (capas de convolución, de pooling, densas, etc.) y determinan la profundidad de la red.
- Número de neuronas en cada capa: esta variable puede ser distinta en cada capa y es determinada a partir de la complejidad de tus datos.
- Tipo de función de activación (RELU, Sigmoide, etc.): estas funciones estimulan las neuronas para que emitan un valor de salida, en las capas de convolución es recomendado usar la función RELU y sus variantes, por entre otras ventajas, ser menos costosas de calcular computacionalmente [Elgendy, 2020].
- Número de mapas de características: este valor determina el número de características que la red intentará buscar para cada una de las capas de convolución, a mayor número de mapas que se intente buscar, más características de la imagen aprenderá la red.

2. Optimización y aprendizaje.

- Tasa de aprendizaje: este valor determina el tamaño del salto que se da dentro de una función de error cuando se está buscando el valor mínimo.
- Tamaño de lote.
- Algoritmos de optimización.
- Número de iteraciones de entrenamiento o épocas (criterio de paro).

3. Técnicas de regularización para evitar el overfitting.

- Regularizador L2.
- Capas de *dropout*: el dropout es una capa que se añade a la red. Fue propuesta por [Hinton et al., 2012] y consiste en tomar de forma aleatoria en cada época o iteración un porcentaje de una capa de neuronas y desactivarlas. Por lo común se usa un dropout de 10, 20 o 30 %.

- Aumento de datos -*Data augmentation*-.

En la sección siguiente, se detallan las tareas de visión por computadora que existen y algunos ejemplos para el caso de sismos.

4.10. Tareas de visión por computadora

La Visión por Computadora (VC) es un campo de la inteligencia artificial que ha tomado relevancia en los últimos años debido a que las computadoras comienzan a realizar procesos que se creían eran únicos para los humanos, como el reconocimiento y detección de imágenes, los coches autónomos, etc.

En la VC se distinguen tres tareas principales: la clasificación, la localización y la detección. Estas se describen enseguida.

4.10.1. Clasificación

En los últimos años la clasificación es sin duda la tarea de visión por computadora más conocida y estudiada, debido a que los modelos de clasificación (como las RNAs) han superado el rendimiento humano, el problema de la clasificación se considera prácticamente resuelto. La clasificación consiste en determinar la etiqueta correcta para una imagen dada. En nuestro problema se presentan únicamente dos clases: la clase sismo (positivo) y la clase ruido (negativo) (Figura 4.17).

4.10.2. Localización

La localización obtiene la ubicación a través de un rectángulo que delimita al objeto dentro de la imagen. Un rectángulo delimitador es el rectángulo que mejor se ajusta de acuerdo a sus coordenadas horizontal x , vertical y , anchura w y altura h , es decir una tupla de 4 elementos (x, y, w, h) que determina la ubicación del objeto, Figura 4.18, inciso a).

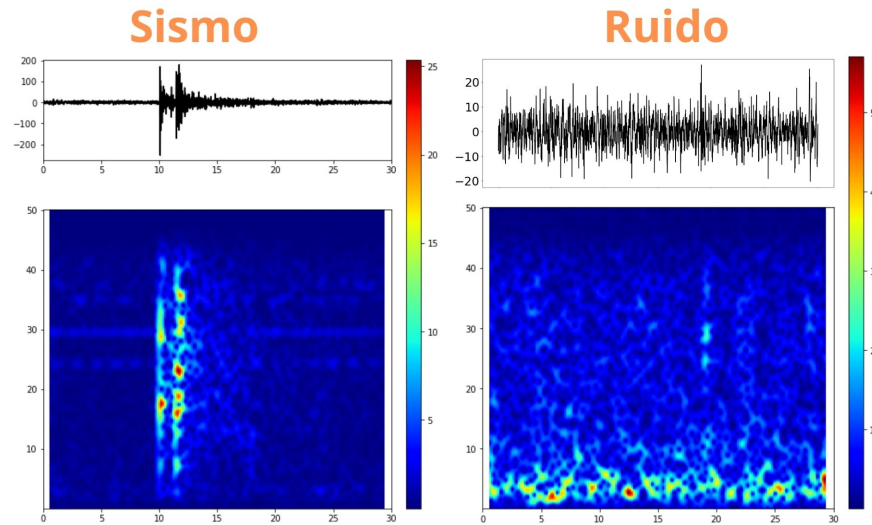


Figura 4.17: Ejemplo de la tarea de clasificación, donde al lado izquierdo se tiene la clase sismo y de lado derecho la clase ruido.

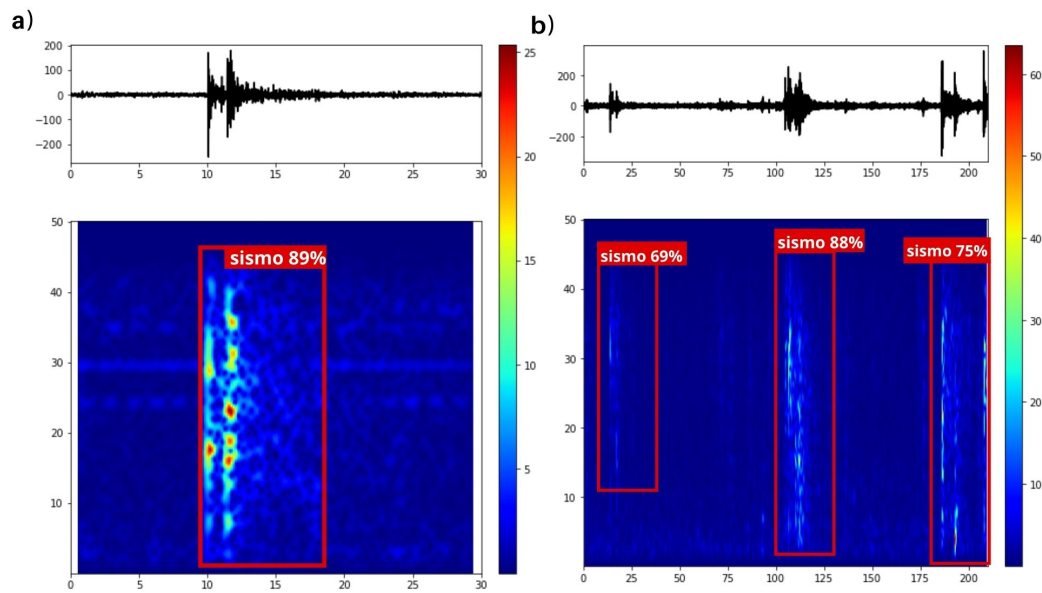


Figura 4.18: Ejemplos de la tarea de a) localización y b) detección de sismos en una imagen.

4.10.3. Detección de objetos

La detección de objetos es una tarea de visión por computadora que se obtiene a partir de localizar y clasificar uno o más objetos en una imagen. Un ejemplo de este proceso se puede observar en la Figura 4.18, inciso b). Este proceso consiste en predecir la ubicación de los objetos en una imagen a través de rectángulos delimitadores, así

como la clase a la que pertenece cada uno de los objetos localizados [Elgendy, 2020], para el caso en particular de la detección de sismos, solamente nos interesa una clase (la clase sismo) por lo que la clase ruido, no se toma en cuenta.

4.11. Detección de objetos utilizando una CNN

La tarea de la detección de objetos se puede aproximar usando una CNN que, en principio, es entrenada para detectar un solo objeto. El proceso para detectar objetos es utilizar ventanas que deslizarán a través de toda la imagen, un ejemplo de este proceso se detalla en la Figura 4.19. En la imagen de ejemplo se genera una malla en la imagen de 6x8 y con una ventana de 3x3 (rectángulo negro) se irá deslizando a través de la imagen de izquierda a derecha y de arriba hacia abajo.

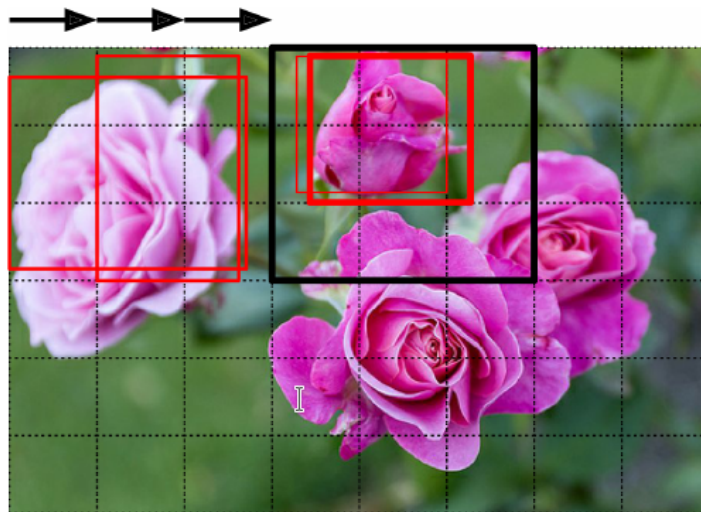


Figura 4.19: Ejemplo con múltiples detecciones. Obtenido a partir de deslizar una ventana (color negro) a través de la imagen, las ventanas en color rojo son los rectángulos que delimitan al objeto, tomado de [Géron, 2019]

Esta aproximación utilizando una CNN resuelve el problema, pero es necesario aplicar un postproceso debido a que realiza saltos pequeños (generalmente 2 o 3 píxeles) y tiende a detectar el mismo objeto múltiples veces, generando múltiples rectángulos delimitadores (ver las ventanas en color rojo de la Figura 4.19).

En el caso particular de la detección de sismos, se tiene una señal en función del tiempo representada a través de un espectrograma tipo imagen, por lo que solamente

se deslizaran ventanas de tiempo de tamaño 30s a través del tiempo, lo que provocara múltiples detecciones para un mismo sismo (Figura 4.20).

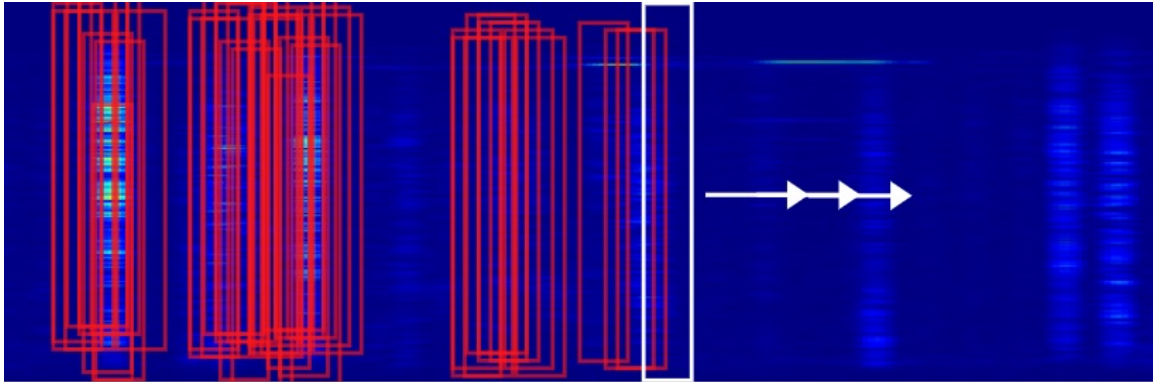


Figura 4.20: Ejemplo de detección de sismos deslizando ventanas de 30s a través del espectrograma (imagen).

Para resolver esto es necesario añadir un postproceso que logre descartar todos los rectángulos delimitadores innecesarios y quedarse con el que presenta la probabilidad máxima para el objeto. A esto se le conoce como supresión de no máximos y se detallará en la siguiente sección.

4.12. Supresión de no máximos

La Supresión de No Máximos (SNM) es una técnica usada para asegurar que el modelo detecta para cada objeto un sólo rectángulo delimitador. El proceso consiste en:

- Obtener todos los rectángulos delimitadores para un solo objeto en la imagen.
- Encontrar el rectángulo con la probabilidad de predicción más alta y suprimir (eliminar) los demás.
- Repetir el proceso para cada objeto en la imagen.

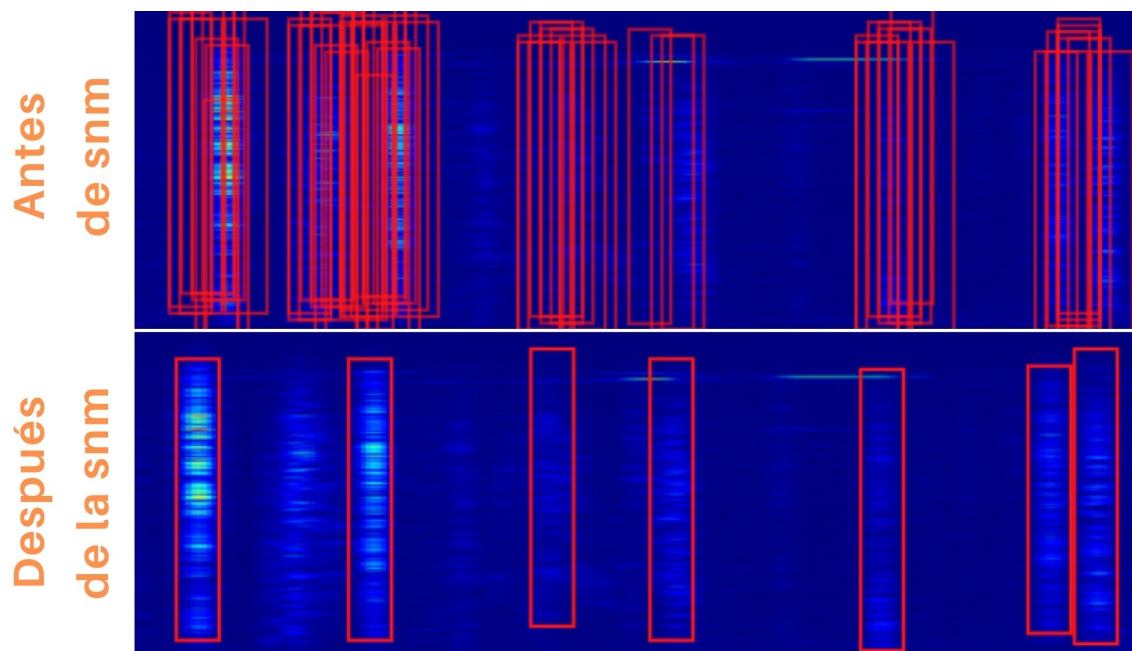


Figura 4.21: Ejemplo del antes (arriba) y después (abajo) de aplicar la snm para el caso de detección de sismos.

4.13. Modelos de aprendizaje profundo

Existen dos tipos de algoritmos de aprendizaje profundo para resolver la tarea de la detección de objetos.

- Los algoritmos de dos fases llevan a cabo la generación de un conjunto de rectángulos delimitadores que son clasificados por una CNN como primer paso (clasificación) y por último se aplica un filtro para obtener la detección final (detección). Los más conocidos son las R-CNN [Girshick et al., 2014], Fast R-CNN [Girshick, 2015] y Faster R-CNN [Ren et al., 2015].
- Los algoritmos de una fase dividen la imagen en regiones de tamaño $S \times S$ que son la capa de entrada de la CNN, para después ser modificadas y agrupadas dependiendo de la clasificación para cada región. Los más conocidos son SSD [Liu et al., 2016], YOLO [Redmon et al., 2016, Redmon and Farhadi, 2017], así como [Redmon and Farhadi, 2018].

En general los algoritmos de dos fases suelen ser más lentos, pero más precisos

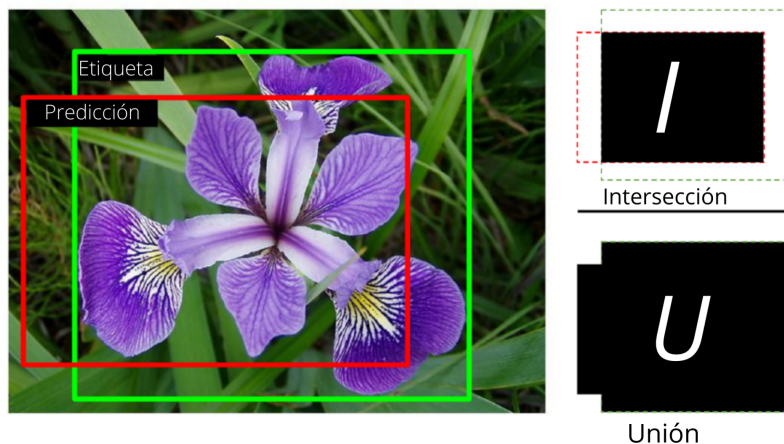


Figura 4.22: Ejemplo de IoU como métrica para evaluar un rectángulo delimitador predicho (rectángulo rojo) del correcto (rectángulo verde), tomado de [Géron, 2019].

que los de una fase [Elgendy, 2020], por lo que se decidió aplicar este algoritmo de dos fases.

La última sección de este capítulo presenta las métricas de evaluación para el método de detección de sismos implementado en la presente investigación.

4.14. Métricas de evaluación

Para evaluar la exactitud de los rectángulos delimitadores se usa la Intersección sobre la unión -*Intersection over Union (IoU)*-. Esta métrica evalúa el traslape (Figura 4.22) de dos rectángulos delimitadores: la clase correcta C_{gt} y la clase predicha C_{pred} al aplicar él (IoU), el cual se define como:

$$IoU = \frac{C_{gt} \cap C_{pred}}{C_{gt} \cup C_{pred}} = \frac{I}{U} \quad (4.4)$$

La IoU es una métrica que define si la predicción fue correcta, donde correcta significa que es un caso que pertenece a la clase positiva y ocurre cuando el IoU es mayor o igual a un umbral establecido, generalmente entre 0 y 1.

La razón por la que se utiliza esta métrica de evaluación es porque resulta más sencilla de calcular, ya que da una idea bastante clara de si la predicción realizada a la imagen contiene el objeto o no. La Figura 4.23 ilustra el IoU para el caso de la

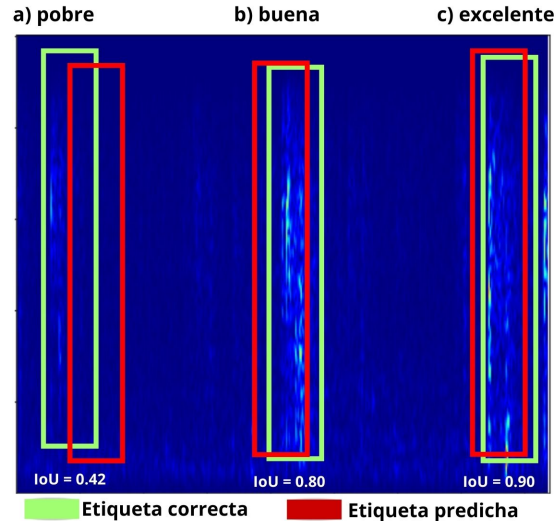


Figura 4.23: Ejemplo de uso de la IoU , donde se considera que una $IoU \geq 0.8$ es un buen resultado.

TN	FP
FN	TP

Tabla 4.1: Matriz de confusión

detección de sismos.

La métrica para cuantificar qué tan buena fue la detección final es la *-Mean Average precision (MAP)-*. Pero antes de definirla es necesario explicar los siguientes conceptos.

La Matriz de Confusión (MC) es una tabla donde las columnas son la clase real y las filas la clase predicha por el modelo (Tabla 4.1). Para el caso de nuestra tarea de clasificación donde se tienen únicamente dos clases (sismo o ruido) se compone de los siguientes cuatro casos:

- TP (*True Positive* - Verdaderos Positivos): casos que sí son sismos y que el modelo neuronal propuesto clasificó como sismos.
- TN (*True Negative* - Verdaderos Negativos): casos que no son sismos y que el modelo propuesto clasificó como no sismos.
- FP (*False Positive* - Falsos Positivos): casos que no son sismos y que el modelo clasificó como sismos.

- FN (*False Negative* - Falsos Negativos): casos que sí son sismos y que el modelo clasificó como no sismos.

Con base en la MC, la métrica más utilizada para evaluar modelos de clasificación es la exactitud (*accuracy*), la cual se define de la manera siguiente:

$$Exactitud = \frac{TP + TN}{TP + FP + TN + FN} \quad (4.5)$$

En virtud de las fallas a las que puede conllevar la exactitud, en situaciones como, por ejemplo, en las que el conjunto de datos está desbalanceado; tal y como el que corresponde a esta investigación en cuanto a que la señal sísmica presenta un porcentaje de tiempo mucho más alto respecto a ruido que a una verdadera señal de sismos, se utilizan las medidas como *Precisión*, *Recall* o *sensibilidad* y el valor F_β para determinar una evaluación más certera sobre el rendimiento final del modelo. La definición de dichas métricas también se realiza con base en la MC:

$$Precisión = \frac{TP}{TP + FP} \quad (4.6)$$

$$Recall = \frac{TP}{TP + FN} \quad (4.7)$$

La métrica F_1 , para $\beta = 1$ es la media armónica de las medidas de evaluación anteriores.

$$F_1 = \frac{TP}{TP + \frac{FN+FP}{2}} = 2 \cdot \frac{Precisión \cdot Recall}{Precisión + Recall} \quad (4.8)$$

La fórmula general para F_β , con β en los reales positivos es:

$$F_\beta = (1 + \beta^2) \cdot \frac{Precisión \cdot Recall}{(\beta^2 \cdot Precisión) + Recall} \quad (4.9)$$

En el ejemplo anterior sobre desbalance de clases, β juega un rol muy importante, debido a que este parámetro determina a qué medida (*Precisión* o *Recall*) se le dará más importancia. De este modo, para dar mayor importancia a la Precisión (menos falsos positivos), β debe tener un valor pequeño; por el contrario, para darle mayor

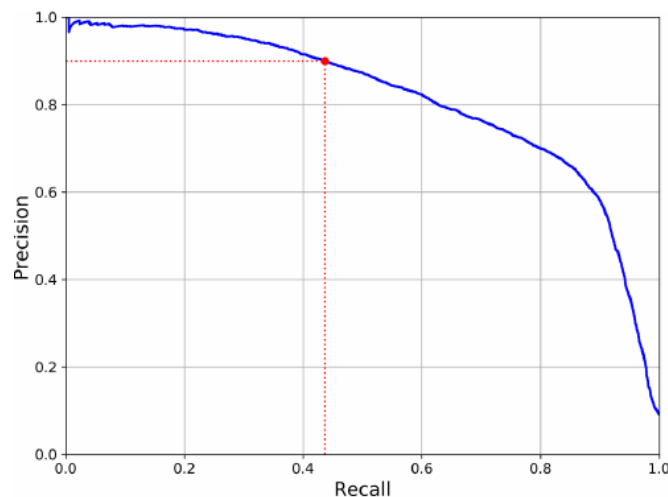


Figura 4.24: Curva Recall-Precisión, tomada de [Géron, 2019]

importancia al Recall (menos falsos negativos), β debe tener un valor grande. Los valores más comunes para β son: 0.5, 1.0 y 2.0

Existen otras métricas basadas en estos conceptos que se usan específicamente para evaluar un modelo de detección de objetos, como el mAP o el AP.

Para entender cómo se calcula el mAP es necesario calcular primero el IoU y la Curva Precisión-Recall (curva PR, Figura 4.24).

En la Figura 4.24 se observa que la curva PR contiene algunas secciones donde la Precisión tiende a disminuir más rápido cuando el Recall incrementa, marcándose más esta correlación en los valores bajos del Recall.

Después de obtener la curva PR por cada clase a detectar se puede obtener su Precisión promedio -*Average precision (AP)*- calculando el área bajo la curva -*Area Under the Curve (AUC)*- y, finalmente el mAP se determina promediando la AP de todas las clases.

El proceso para calcular el mAP es el siguiente:

1. Cada objeto detectado (rectángulo delimitador) de una clase C tendrá un puntaje asociado, el cual representa la probabilidad de que en ese rectángulo exista el objeto.
2. Con base en la MC se calcula la precisión y el Recall para cada clase de objetos a detectar.

3. Se calcula la curva PR para cada clase C de objetos a detectar.
4. El AP se calcula para cada clase C , es decir, el área bajo la curva (AUC) de PR. En este paso el AP es calculada para cada clase.
5. mAP: Se calcula a partir de sacar el promedio de las AP de todas las clases en C , ver Ecuación 4.10.

$$mAP = \frac{\sum_{c \in C} AP_c}{|C|} \quad (4.10)$$

En el siguiente capítulo se detallan los pasos que se ejecutaron durante el desarrollo de la metodología para resolver el problema de detección utilizando aprendizaje profundo. Además, se presentan los valores para cada una de estas métricas de evaluación referidas al modelo empleado en esta investigación.

Capítulo 5

Metodología

El desarrollo de los modelos de aprendizaje profundo se planteó en concordancia con la metodología sugerida en [Géron, 2019]. La metodología facilita la aplicación exitosa de los modelos de aprendizaje automático y se conforma de los siguientes pasos:

1. Detección del problema
2. Obtención de los datos
3. Análisis exploratorio de datos
4. Preparación de los datos (preprocesamiento)
5. Selección de uno o más modelos candidatos
6. Afnación de los parámetros del modelo
7. Análisis de los resultados obtenidos
8. Despliegue en producción del sistema

5.1. Obtención y análisis de datos

Los datos que se usaron para el trabajo de esta tesis fueron obtenidos del observatorio con clave PNIG, perteneciente a la Red de Banda Ancha del SSN, ubicado en Pinotepa Nacional, Oaxaca; en el sureste de México (Figura 5.1).



Figura 5.1: Ubicación de la estación del SSN con clave PNIG y del lugar donde ocurrió el sismo de Ometepec el 20 de marzo de 2012.

El observatorio PNIG cuenta con un sismómetro de 3 componentes de respuesta plana: vertical (Z), Norte-Sur (N) y Este-Oeste (E). La información capturada por el sismómetro fue registrada a 100 muestras por segundo (mps) y se digitalizó con un dispositivo de 26 bits.

Los datos fueron tomados durante el periodo que comprende del 1 de enero al 20 de abril de 2012. En este periodo se obtuvo un catálogo de 17,786 sismos por componente etiquetados por un analista. Cada uno de los 17,786 sismos, fueron delimitados por una ventana de tiempo con duración de 30 segundos (s) para cada una de las componentes de respuesta plana (Z, N y E) en formato SAC¹.

¹Formato de archivo con una estructura de datos especializada para almacenamiento de información correspondiente a series de tiempo [IRIS-SAC, 2017].

Durante todo el desarrollo de la investigación se asume que la señal es continua en el dominio del tiempo y que es afectada por una cierta cantidad de ruido, entendiéndose como ruido, todas aquellas señales provenientes de otras fuentes distintas a la señal que se genera cuando ocurre un sismo (ondas de cuerpo y/o superficiales).

Un señalamiento importante con respecto a los sismos del conjunto de datos es que en la mayoría de ellos no se presentan ondas superficiales debido a que son réplicas de un sismo ocurrido en Ometepec, Guerrero, con magnitud $M_w = 7.4$ el 20 de marzo de 2012 y por su naturaleza son sismos con una tasa señal a ruido baja, por lo que la tarea de detección se complica ante este problema de discriminación.

El sismo de Ometepec no se agregó al catálogo, pues provocó la saturación del instrumento dada su magnitud y cercanía con respecto a la estación PNIG (Figura 5.1).

En la siguiente sección se detalla el preprocesamiento que se aplicó a los datos, la forma de generar el conjunto negativo y la obtención de los conjuntos de entrenamiento, validación y prueba.

5.2. Preprocesamiento

Para resolver la tarea de detección de sismos, se usó un algoritmo de dos fases basado en una CNN: en la primera fase se resuelve la tarea de clasificación y en la segunda la tarea de detección. Para la primera fase es necesario crear un conjunto de entrenamiento y otro de prueba para nuestro modelo de RNA. Cada uno de estos conjuntos deberá tener ejemplos positivos (sismos) y negativos (ruido) para que el modelo logre aprender a discriminar sobre lo que es un sismo y lo que no lo es.

Para la primera fase se utilizó únicamente la componente Z, teniendo así un conjunto de 17,786 sismos, al cual nos referiremos como conjunto positivo. Los sismos del conjunto positivo varían en cuanto a duración (Figura 5.2) y magnitud. Los sismos menores a 3s generalmente ocurren cerca de la estación, lo que provoca la saturación del instrumento, por lo que es preferible descartarlos y, además, se descartó un evento por no tener una ventana de 30s (*id = event_rank00000_time1848.0HHZ.sac*).

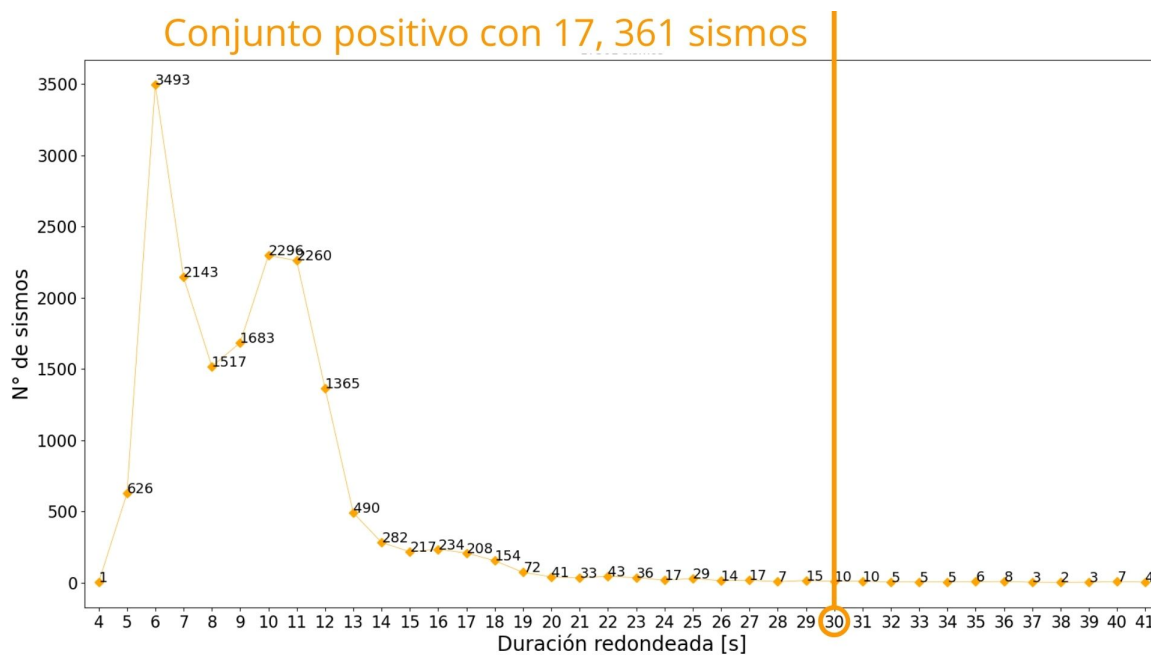


Figura 5.2: Histograma de duraciones de sismos que conforman el conjunto positivo, usando la componente Z y filtrando 425 eventos (ver texto).

Con este primer filtro se obtiene un total de 17,361 eventos positivos y 425 eventos descartados.

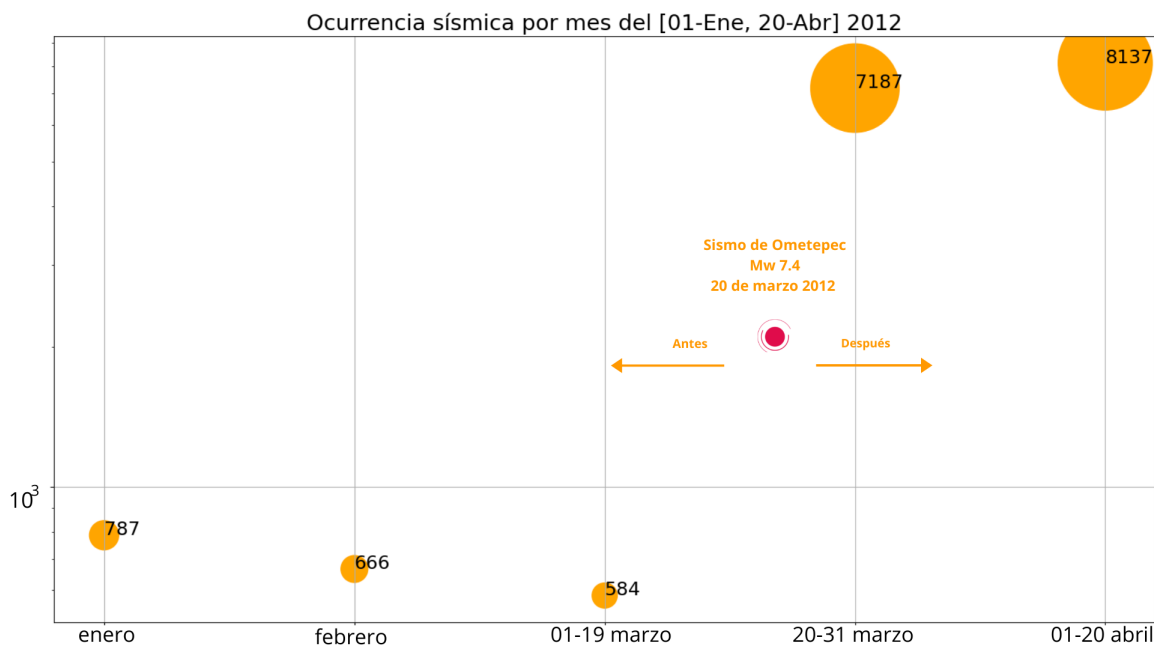


Figura 5.3: Ocurrencia sísmica durante el periodo de enero-abril de 2012 registrada por la estación PNIG en Pinotepa Nacional, Oaxaca.

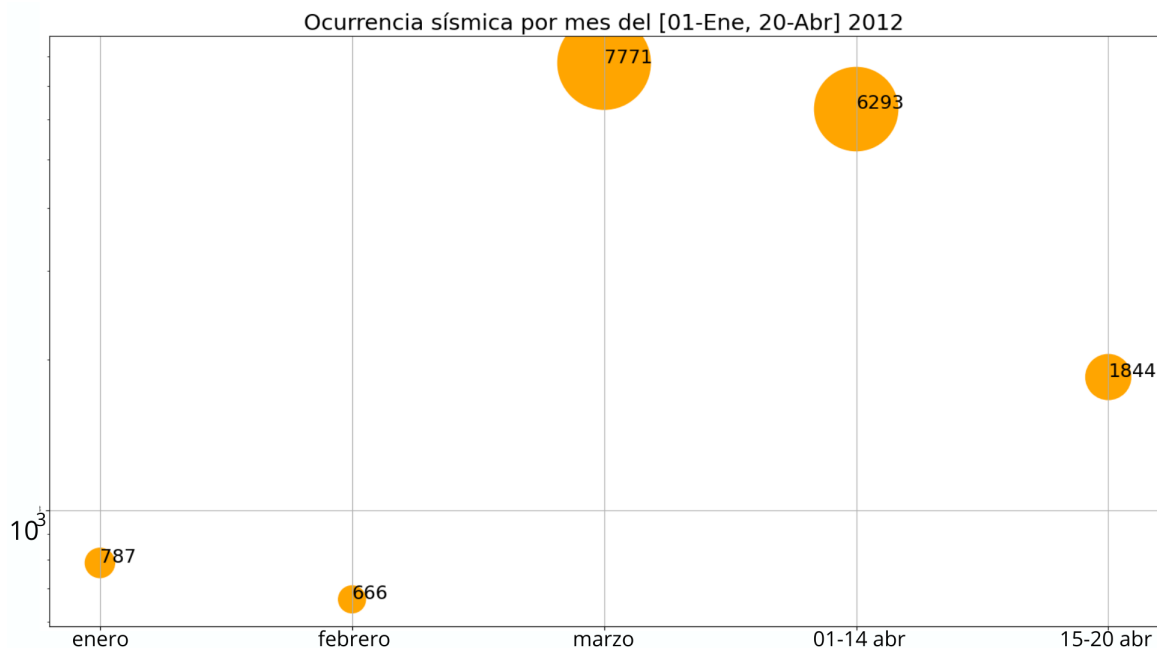


Figura 5.4: Ocurrencia de sismos por intervalos. El conjunto positivo se divide en dos subconjuntos: el primero se usa para la primera fase de entrenamiento del modelo, dentro del lapso del 1 de enero al 14 de abril de 2012 y, el segundo, para la fase de detección en una traza de 6 días, del 15-20 de abril de 2012

Durante el periodo en que se tomaron los datos se observan dos estados principales, uno antes del sismo de Ometepec y otro después de este (Figura 5.3). Antes del sismo de Ometepec se presentan sólo 2,037 sismos durante el periodo del 01 de enero al 20 de marzo y, del 21 marzo al 20 de abril (después del sismo de Ometepec), se tienen 15,324 sismos. De este último intervalo en su mayoría son réplicas del sismo de Ometepec con una baja tasa señal a ruido, lo que representa el 88 % del total de sismos existentes en el conjunto positivo.

Para la primera fase del modelo es necesario crear los conjuntos de entrenamiento y prueba que contengan tanto eventos positivos (sismos) como eventos negativos (no sismos o ruido), en un porcentaje 80/20, debido a que esta separación agrupa muy bien al conjunto global, permitiendo entrenar con el 80 % de los datos al modelo de RNA y evaluarlo con el 20 % restante. En la siguiente sección se describe este proceso.

5.2.1. Conjuntos de datos para entrenamiento, validación y prueba

Como se habló en la sección anterior, es preciso dividir nuestra traza sísmica del [01-ene, 20-abr] de 2012 en dos conjuntos principales: un primer conjunto para entrenamiento y prueba del modelo con el que se resuelva la tarea de clasificación; y un segundo conjunto para resolver la tarea de detección. Con esta finalidad, la traza sísmica de casi 4 meses se dividió en un primer lapso del [01-ene, 14-abr] para la tarea de clasificación (15,517 sismos, que representa el 80%) y dejar los últimos 6 días ([15-abr, 20-abr]) para la tarea de detección (1,844 sismos, que representa el 20% del total), Figura 5.4.

El conjunto de datos positivo fue etiquetado por un analista y contiene los 17,361 archivos SAC de la componente Z, con ventanas de tiempo de 30s, los cuales contienen al sismo (ondas P y S). En cambio, el conjunto de datos negativo se conforma de ventanas de tiempo de 30s que no contienen eventos del conjunto positivo (sismos), es decir, solamente ruido capturado por el instrumento.

El conjunto negativo se generó usando el Algoritmo 1 dentro del lapso de tiempo que comprende del [01-ene, 14-abr] de 2012.

Algoritmo 1: Algoritmo para la generación del conjunto negativo

Resultado: Una lista con los lapsos donde exista únicamente ruido

Entradas: *Lapsos_libres*, *eventos_neg*

Lapsos_libres <- Lista con las marcas de tiempo de los lapsos donde no existen sismos durante el periodo [01-ene, 14-abr];

eventos_neg <- Lista de las marcas de tiempo generadas de forma aleatoria siguiendo una distribución uniforme, durante el periodo [01-ene-14-abr];

```

1 /* Realizar búsqueda para insertar cada una de las ventanas de tiempo
   generadas en los lapsos libres, cuidando que no se traslapen con los sismos */
2 l_i <- índice i de cada lapso generado en eventos_neg inicializado en 0
   mientras que aun existan eventos negativos que se tienen que agregar
   hacer
3   l_0 <- marca de tiempo inicial de la ventana de tiempo i;
4   l_f <- marca de tiempo final de la ventana de tiempo i;
5   para lapso libre en Lapsos_libres hacer
6     t_0 <- marca de tiempo inicial del lapso libre;
7     t_f <- marca de tiempo final del lapso libre;
8     si l_0 es  $\geq t_0$  y l_f  $< t_f$  entonces
9       agregar el lapso de [l_0, l_f] como lapso no libre;
10      actualizar Lapsos_libres;
11      incrementar i = i + 1;
12     en otro caso
13       Descartar evento negativo
14     fin
15   fin
16 fin

```

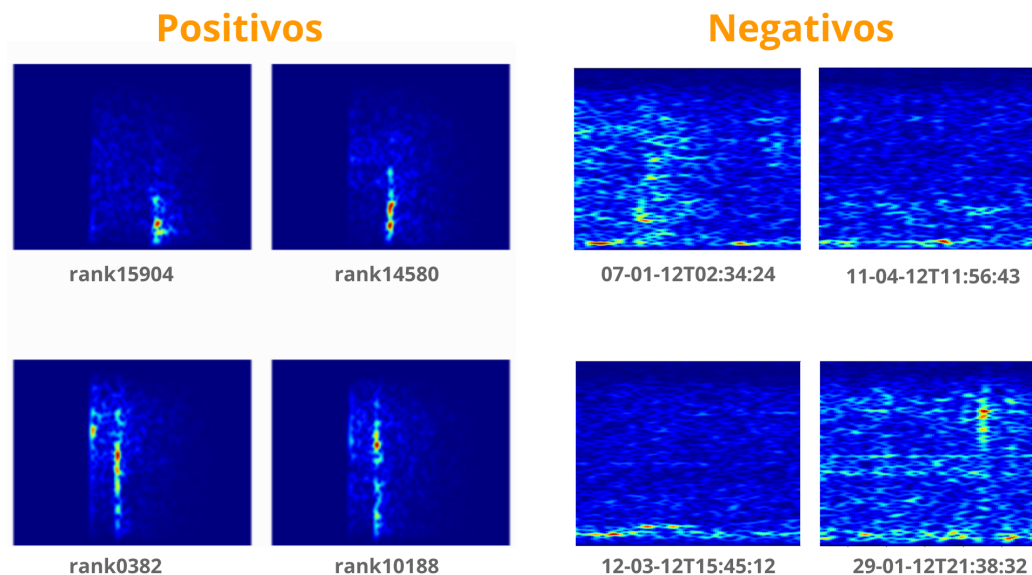


Figura 5.5: A la izquierda se muestran 4 espectrogramas positivos (sismos, donde se observan las ondas P y S que componen el sismo). A la derecha se muestran 4 ejemplos de espectrogramas negativos (ruido) generados para el conjunto de entrenamiento. Se añaden los identificadores alfanuméricos de los sismos (id), así como un id para los eventos negativos, que se basa en las marcas de tiempo.

En la Figura 5.5 se muestran 4 ejemplos de espectrogramas positivos y 4 negativos a manera de ilustrar cómo se ven estos eventos, donde para cada una de estas imágenes se observa el espectro de la señal en una ventana de 30s.

A partir del conjunto positivo, se determinaron los conjuntos de datos como se detalla en la Tabla 5.1. De esta forma el conjunto positivo se compone de 15,517 espectrogramas y el conjunto negativo de 15,496. Aquí es importante mencionar que no se asignaron eventos negativos para la parte de detección, debido a que dicho proceso tendrá como entrada una traza continua de 6 días de muestreo, del 15 al 20 de abril de 2012.

Al sumar los 15,496 espectrogramas negativos más los 15,517 positivos se tiene un conjunto de 31,013 espectrogramas en total. De este conjunto combinado de espectrogramas positivos y negativos, se toman de forma aleatoria 24,812 espectrogramas para entrenamiento, que representan el 80% del total; y el restante, que representa el 20% de los espectrogramas, para el conjunto de prueba con 6,201 espectrogramas.

Fase	Algoritmo	Conjunto	Positivos	Negativos	Espectrogramas
1. Clasificación	CNN	Entrenamiento	12,414	12,398	24,812
		Prueba	3,103	3,098	6,201
		Total	15,517	15,496	31,013
2. Detección	CNN	Traza de 6 días	1,844		

Tabla 5.1: Conjuntos de datos para el modelo de detección de sismos

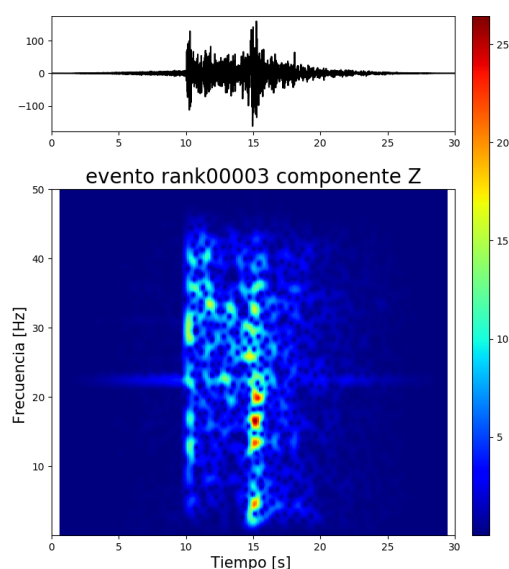


Figura 5.6: Ejemplo de un sismograma (arriba) y su respectivo espectrograma (abajo) registrado por la estación sismológica PNIG en Pinotepa Nacional, Oaxaca.

5.2.2. Generación de espectrogramas usando Obspy

Las redes neuronales convolucionales trabajan mejor con imágenes, es por ello que se utilizaron los espectrogramas debido a que son semejantes a una imagen, como se observa en la Figura 5.6.

Obspy es un proyecto de código abierto enfocado al análisis y procesamiento de datos sísmicos que se puede instalar como biblioteca de Python². Con esta biblioteca se generaron los espectrogramas a partir de los datos observados.

Con el apoyo de la biblioteca Obspy, a cada ventana de tiempo de 30s de los conjuntos de datos positivos y negativos, se le aplicó un filtro *detrend*³ que estabiliza

²Lenguaje de programación que ha tomado en los últimos años gran aceptación y relevancia por su fácil uso y comprensión [Pedregosa et al., 2011].

³Genera una función lineal definida por la primera y última muestra de la traza [(devs@obspsy.org), 2020].

la traza sísmica y un filtro pasa altas (*high pass*) con una frecuencia de 2 Hz, para filtrar señales con frecuencias menores a ésta (para eliminar, por ejemplo, la señal captada por el sismómetro proveniente del oleaje marino que alcanza a detectar la estación de Pinotepa).

El resultado de calcular el espectrograma de la señal sísmica de los eventos positivos y negativos permite obtener el conjunto de datos final para el desarrollo del modelo de detección de sismos, con base en los conjuntos de datos definidos en la Tabla 5.1.

Los primeros conjuntos se utilizaron para resolver la tarea de clasificación utilizando un modelo de redes neuronales convolucionales y, durante la fase de entrenamiento, se ajustaron parámetros y metaparámetros del modelo hasta obtener un resultado óptimo que resolviera la tarea de clasificación. En la siguiente sección se detalla la arquitectura y parámetros, tiempo de ejecución y valores de las métricas con las que se evaluó el modelo.

5.3. Primera fase: entrenamiento de la CNN

Las CNNs trabajan con imágenes que se pueden representar como una matriz de píxeles de $n \times n$. Cada uno de estos píxeles representa un valor de tonalidad en la imagen.

Las imágenes de nuestros conjuntos de datos en un principio mostraban distintas escalas de dimensión al momento de generar el espectrograma. El tamaño de la imagen es determinado por el espectro de la señal resultante de aplicar la FFT. Para solucionar ese problema las imágenes fueron redimensionadas a 3 canales en RGB con valores de 0 a 255 y tamaño de 60×60 píxeles (px), ver Figura 5.7.

Con ello los conjuntos de datos quedaron normalizados para ser manipulados en una matriz multidimensional conocida como tensor (ver Tabla 5.2), pues además de disminuir el uso de espacio en la memoria del equipo de cómputo, también reduce el número de cálculos que hace la CNN en el proceso de entrenamiento.

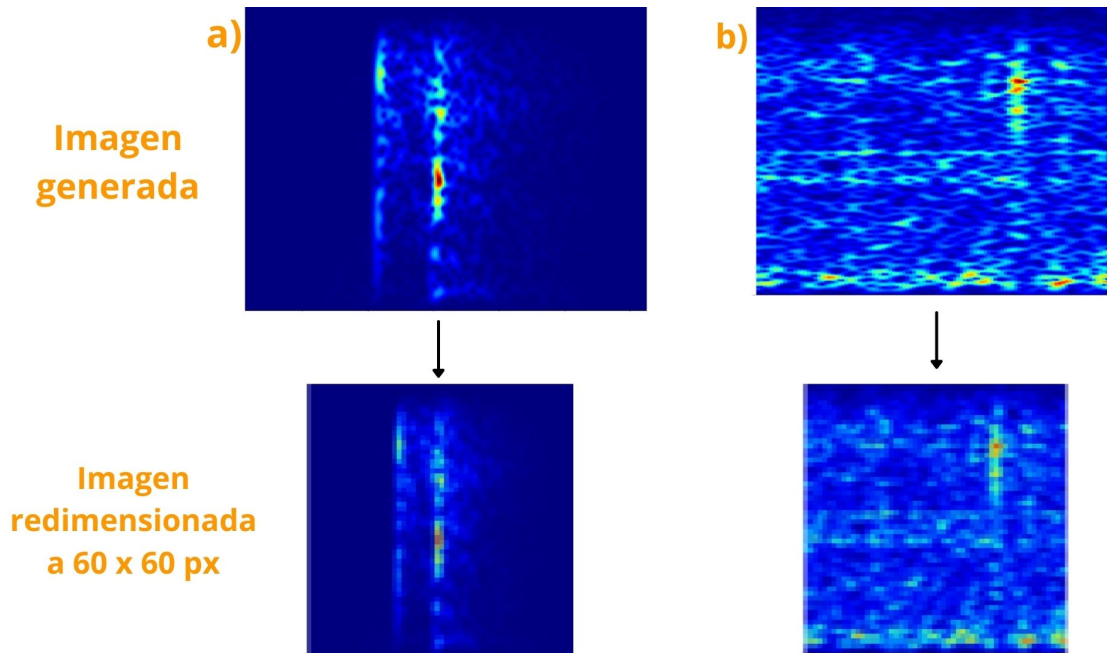


Figura 5.7: Ejemplos de espectrogramas generados para el caso a) positivo y b) negativo (arriba) junto con su respectiva imagen redimensionada (abajo) de tamaño 60×60 px.

Tensor	Dimensión
Entrenamiento	$24,812 \times 60 \times 60 \times 3$
Prueba	$6,201 \times 60 \times 60 \times 3$

Tabla 5.2: Dimensiones de los conjuntos de datos para el modelo de detección de sismos

Configuración del entorno para aprendizaje profundo

Para resolver la tarea de clasificación se entrenó una CNN implementada con la biblioteca Keras, la cual es una biblioteca para desarrollo de modelos de redes neuronales de alto nivel desarrollada en Python. Además, permite la compilación y ejecución desde diversas unidades de procesamiento, tales como la CPU o GPU.

En particular el uso de la unidad de procesamiento gráfico (GPU) mejora el rendimiento en el tiempo de ejecución de los modelos de redes neuronales, por lo que se hizo uso de algunas tarjetas de aceleración de gráficos en una computadora portátil y un servidor con las siguientes características:

1. Equipo 1: Computadora portátil Dell Precisión 3520

- Procesador Intel(R) Core(TM) i7-7700HQ CPU @ 2.80GHz 2.80 GHz
- Tarjeta gráfica NVIDIA Quadro M620 2GB de RAM DDR4

2. Equipo 2: Servidor

- Procesador Intel(R) Core(TM) i7
- Nvidia GTX 1060 Ti, con 6GB de RAM DDR5

Como complemento a estas características, en la Tabla 5.3 se presentan los detalles en cuanto al software utilizado en cada uno de los equipos de cómputo, para la realización de las pruebas de clasificación y detección de sismos.

Equipo	SO	GPU	Cuda	Toolkit	CuDNN	Python
1	Windows	Quadro M620	11.1	10.1.243	7.6.5	3.8.5
2	Ubuntu 18.04	GTX 1060 Ti	10.1	10.0	7.4	3.7

Tabla 5.3: Configuración del entorno de software para aprendizaje profundo en cada uno de los equipos.

La arquitectura que se usó para la CNN de los experimentos se definió tomando como base la arquitectura descrita en la Tabla 5.4, la cual define un modelo con 4 capas convolucionales, 4 capas de pooling y un último bloque de neuronas completamente conectadas.

Metaparámetro	Valor
Primer bloque	
Tamaño de entrada	$60 \times 60 \times 3$
Capas de convolución	4
Stride	5
Capas de pooling	4
Tipo de pooling	Max pooling
Función de activación	RELU
Segundo bloque	
Capa densa	100 neuronas
Función de activación	RELU
Bloque de salida	
Capa de salida	1 neurona
Función de activación	Sigmoide

Tabla 5.4: Parámetros de la red.

Para cada una de las neuronas en las capas de convolución, se le aplicó una función de activación RELU. La función RELU (y sus variantes) es la función que más se recomienda usar en estas capas [Ng, 2017], por dos razones principales: la primera debido a que es menos costosa computacionalmente comparada con otras funciones, y la segunda porque se obtienen los mejores resultados en cuanto a rendimiento y aprendizaje en las capas ocultas de este tipo de redes de convolución [Elgendy, 2020].

El último bloque de la red es un conjunto de neuronas completamente conectadas, donde cada una de las neuronas sinápticas está conectada a todas las neuronas de la capa anterior. Dentro de este bloque se usó la función de activación RELU. En la última capa de salida, se optó por usar una sola neurona con una función de activación sigmoide, cuyo valor de salida es una probabilidad $u \in [0, 1]$. Esta función en particular será muy útil para la segunda fase del algoritmo, dado que asigna a la imagen de entrada su probabilidad de ser sismo.

El hecho de escoger los parámetros y metaparámetros de la red no es trivial, es por ello que se siguió un proceso metodológico para ir ajustando estos valores en distintas configuraciones de la arquitectura, hasta obtener el mejor resultado posible. En la Tabla 5.5 se ve cada una de las configuraciones de los metaparámetros que se fueron modificando hasta obtener el mejor resultado.

conf	lr	épocas	val	lote	pérdida	exac	precisión	recall	F_1
1	1^{-6}	10	0.1	128	0.93	75.46	74.18	75.13	74.49
2	1^{-6}	10	0.2	256	0.78	80.54	83.28	80.34	79.29
3	1^{-5}	10	0.2	512	0.57	95.73	95.51	95.97	95.74
4	1^{-5}	50	0.2	512	0.0626	99.37	98.76	100.0	99.38

Tabla 5.5: Configuración de metaparámetros de la red: configuración, learning rate (tasa de aprendizaje α), número de épocas, porcentaje de validación, tamaño de lote, valor de la función de pérdida, exactitud, precisión, recall (sensibilidad) y métrica F_1 .

Se comenzó con la arquitectura base de la red mostrada en la Tabla 5.4 y con la Configuración 1, que son los valores de los metaparámetros con los que se recomienda comenzar [Elgendy, 2020]. Dado que el número de épocas determina cuántas veces itera la red, se recomienda comenzar con un número pequeño, para que no tarde demasiado en terminar y de esta forma hacer los ajustes necesarios en el menor tiempo

posible. Para este caso se consideró un valor de 10 iteraciones por entrenamiento. Un valor de la tasa de aprendizaje pequeño $\alpha = 1^{-6}$ y un tamaño de lote pequeño. Con esta primera configuración se obtuvo una exactitud considerable de 75.46 %, por lo que en la siguiente configuración se decidió aumentar el tamaño de lote y el tamaño del conjunto de validación, lo que ayudó a mejorar el rendimiento del modelo en un 5 % más en la exactitud y disminuir la función de pérdida. Para la tercera configuración se aumentó el tamaño de lote a 512 y se incrementó el valor de $\alpha = 1^{-5}$ lo que ayudó a aumentar drásticamente la exactitud. Durante la aplicación de la Configuración 3 se observó que presentaba un patrón de aprendizaje el modelo. En cada época la exactitud aumentaba, mientras que la función de pérdida disminuía, aunque en menor medida. Para lograr que la exactitud continuara aumentando y la función de pérdida disminuyendo, se aumentó el número de épocas hasta 50 para observar en qué momento la gráfica se estabilizaba, para tomar la decisión de paro.

En la Figura 5.8 se observa claramente el patrón de aprendizaje, donde en cada iteración la función de exactitud continuaba incrementándose, mientras que la función de pérdida se mantenía disminuyendo. En la época 50 se observa que la exactitud se estancó en un valor muy cercano a 1 y la función de pérdida disminuyó de manera considerable cerca del orden de 6×10^{-2} , el valor 50 se tomó entonces como condición de paro y se tomó la Configuración 4 como la que mejor resultados dió en la fase de entrenamiento.

De este modo la red aprendió de forma autónoma un total de 5,488 parámetros (pesos de las neuronas) y 200 metaparámetros definidos al momento de crear la configuración y arquitectura de la red.

La arquitectura final de la CNN se muestra en la Figura 5.9.

5.3.1. Curva de aprendizaje y función de pérdida

Las curvas de aprendizaje sirven para observar el comportamiento de la red en la fase de entrenamiento, debido a que para cada una de las épocas se puede observar la tasa de cambio del aprendizaje de la red y de la función de pérdida.

En cada iteración (época) de entrenamiento que se le da al modelo, se separan

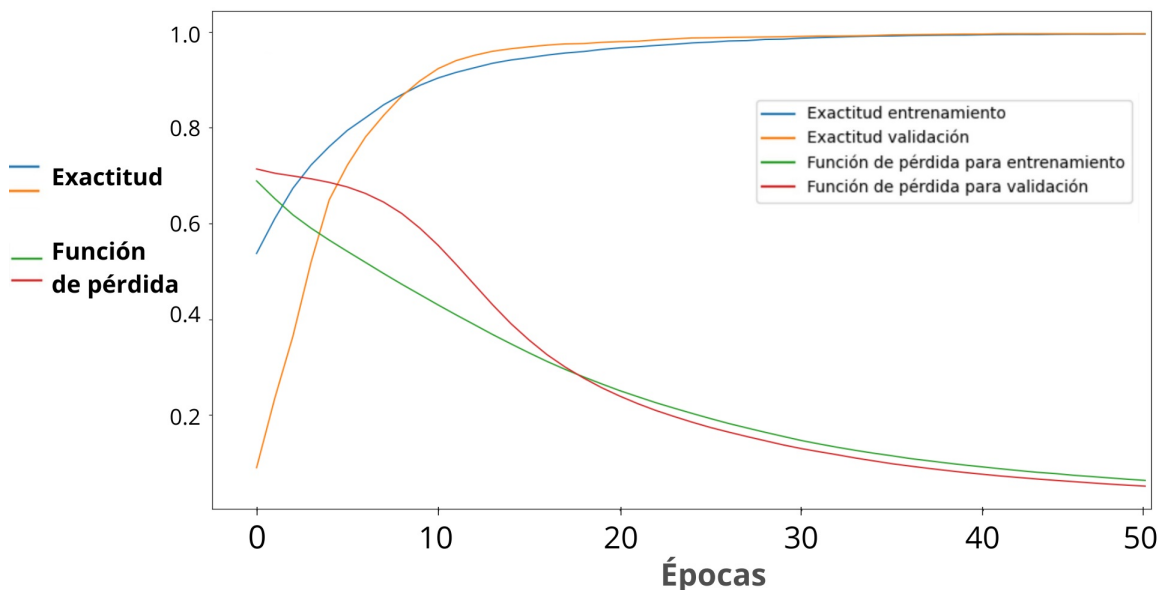


Figura 5.8: Desempeño del modelo de CNN durante el proceso de determinación de metaparámetros para el proceso de clasificación de imágenes (sismos).

de forma aleatoria los 24,812 espectrogramas del conjunto de entrenamiento en dos subconjuntos: uno con proporción del 80 % (alrededor de 19,850 espectrogramas) que será usado para ajustar pesos y parámetros de la CNN en cada iteración; y otro con la proporción restante (4,962 espectrogramas que representan el 20 %) conocido como conjunto de validación. El conjunto de validación se conforma de ejemplos de espectrogramas que probablemente la red haya visto en épocas anteriores y cada una de estas imágenes se le da de entrada a la CNN previamente entrenada para **validar** que el conocimiento que está aprendiendo esté mejorando en cada iteración.

En la Figura 5.8 se observa el comportamiento de la CNN en su fase de entrenamiento. Las curvas de exactitud son las que se muestran en color azul y naranja, éstas se mantienen aumentando conforme el número de épocas incrementa para ambos conjuntos y, dado que las curvas se mantienen crecientes, se sabe que la CNN está aprendiendo en cada época. A su vez, esto se confirma con las funciones de pérdida, que son las curvas en color verde y rojo, y que decrecen de manera monótona, lo que implica que el error en la predicción también decrece.

De acuerdo con la Figura 5.8, el modelo propuesto muestra una tasa de aprendizaje excelente debido a que las funciones de exactitud mantienen valores cercanos a 1 y,

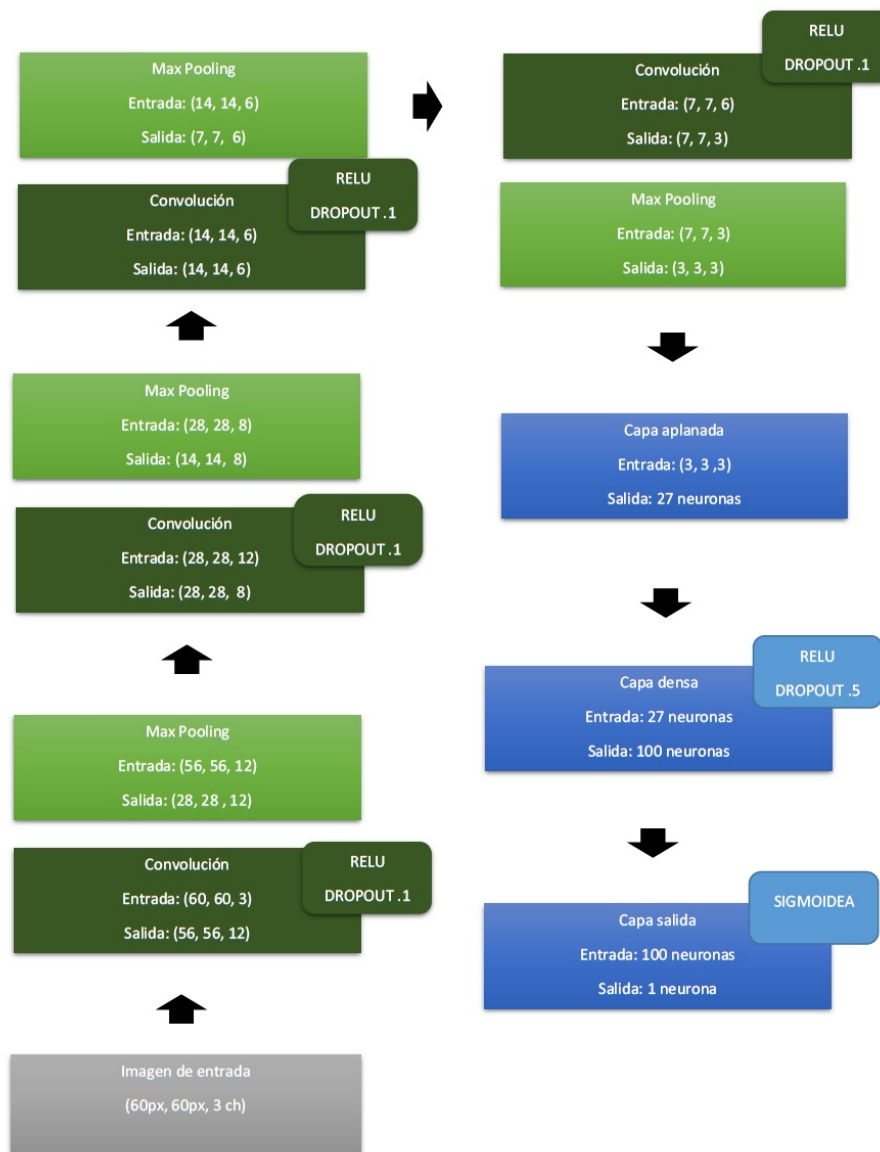


Figura 5.9: Arquitectura de la CNN usada para resolver la tarea de clasificación de imágenes (sismos).

para las funciones de error, los valores se mantienen cercanos a 0. Sin embargo, resulta necesario probar su nivel de generalización en cuanto al aprendizaje, es decir, utilizar la CNN para clasificar ejemplos de espectrogramas de señales sísmicas (o ruido) que no haya analizado con anterioridad. El proceso referido se realiza empleando el conjunto de prueba, a la par que se evalúa con las métricas expuestas en la siguiente sección.

		Real	
		Sismo	Ruido
Predicha	Sismo	3103	39
	Ruido	0	3059

Figura 5.10: Matriz de confusión

5.3.2. Evaluación de la primera fase del modelo

Una vez entrenada la CNN, se le dan nuevos ejemplos de espectrogramas que nunca ha visto, estos ejemplos se encuentran en el conjunto de prueba y se le pide que clasifique cada uno de ellos.

La matriz de confusión (MC) que se obtiene con base en el conjunto de prueba se muestra en la Figura 5.10 y, de acuerdo con ella, se puede calcular la métrica siguiente:

$$Exactitud = \frac{TP + TN}{TP + FP + TN + FN} = \frac{3103 + 3059}{3103 + 39 + 3059} = \frac{6162}{6201} = 0.9937$$

A pesar del nivel de excelencia de este resultado, alcanzando casi el 100%, la exactitud puede ser una métrica poco efectiva para conjuntos desbalanceados, como es el caso de la clasificación de sismos. Por este motivo, los resultados en cuanto a precisión, recall o sensibilidad y métrica F_1 del desempeño del modelo también se calcularon y se presentan enseguida:

$$Precisión = \frac{TP}{TP + FP} = \frac{3103}{3103 + 39} = 0.9876$$

$$Recall = \frac{TP}{TP + FN} = \frac{3103}{3103 + 0} = 1.0$$

La métrica F_1 es la media armónica de las medidas de evaluación anteriores, la

cual da preferencia a valores altos de ambas y cuyo valor es:

$$F_1 = \frac{TP}{TP + \frac{FN+FP}{2}} = \frac{3103}{3103 + \frac{39}{2}} = 0.9938$$

La Tabla 5.6 presenta en forma condensada las métricas citadas, las cuales hacen ver que los resultados son excelentes para la primera fase. Con lo anterior se puede concluir que la red neuronal resuelve la clasificación de forma satisfactoria, por lo que se aplicará esta misma red en la segunda fase, la detección.

En la Tabla 5.7 se muestran los tiempos de duración de cada época de entrenamiento de la CNN en cada uno de los equipos de cómputo utilizados. El equipo portátil se utilizó principalmente para hacer las primeras pruebas y ajustes en los parámetros y metaparámetros del modelo hasta observar una configuración óptima. Una vez visto que los resultados para las primeras épocas mejoraban, con esa misma configuración se probó el modelo en el equipo servidor con hasta 50 épocas, observando que dicho número presentaba los mejores resultados, tal y como se ve en la Figura 5.8.

Métrica	Porcentaje
Exactitud	99.37 %
Precisión	98.76 %
Recall	100 %
F_1	99.38 %

Tabla 5.6: Métricas de evaluación de la clasificación de sismos con la CNN propuesta.

Equipo	Tiempo	GPU
Computadora portátil	5.74 min	NVIDIA Quadro M620
Servidor	1.9 min	Nvidia GTX 1060 Ti

Tabla 5.7: Tiempos de duración de cada época de entrenamiento de la CNN en los distintos equipos utilizados.

En el capítulo siguiente se detallan los experimentos y resultados realizados para la segunda fase, etapa en la cual se probó la detección de sismos usando el modelo de CNN generado anteriormente y, además, se evaluó su desempeño con algunas métricas adicionales, como es el mAP y la curva ROC.

Capítulo 6

Experimentos y resultados

Una vez encontrada la mejor configuración y arquitectura para el modelo de la CNN propuesta, se realizó la segunda fase (la detección de sismos) poniéndola a prueba con la traza de 6 días que se separó del conjunto de datos original. Esta traza comprende del primer segundo del 15 de abril hasta el último segundo del 20 de abril de 2012 y contiene 1, 844 sismos etiquetados por un analista.

El modelo tardó casi 25 horas en procesar toda la traza (detectar sismos) en el servidor que cuenta con una unidad de procesamiento gráfico Nvidia GTX 1060 Ti.

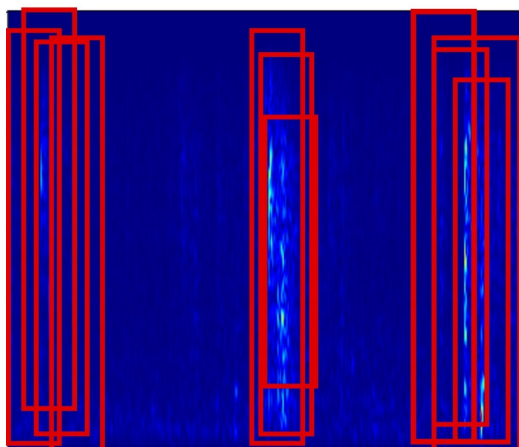


Figura 6.1: Ejemplo de múltiples rectángulos delimitadores detectados para cada sismo en la imagen.

La detección se llevó a cabo deslizando ventanas de 30s a través de toda la traza de 6 días, con saltos de 1s. De manera que cada una de estas ventanas es la entrada

de la CNN entrenada y la salida que devuelve es la probabilidad o umbral $u \in [0, 1]$ de que el rectángulo delimitador contenga un sismo. Para considerar la existencia de un sismo en el rectángulo que delimita al objeto se toma una $u \geq 50\%$.

Con esta aproximación y debido principalmente a que se realizan saltos de 1s, para cada sismo se presentan múltiples rectángulos delimitadores. En la Figura 6.1 se puede observar que, para cada uno de los 3 sismos que existen en la imagen, se detectaron los mismos objetos múltiples veces.

Para eliminar estas detecciones múltiples, se llevó a cabo un postproceso que consiste en seleccionar el rectángulo delimitador con la probabilidad más alta de ser sismo y eliminar las demás (Figura 6.2). Este postproceso se conoce como "Supresión de no máximos"(SNM) y se detalla en la siguiente sección.

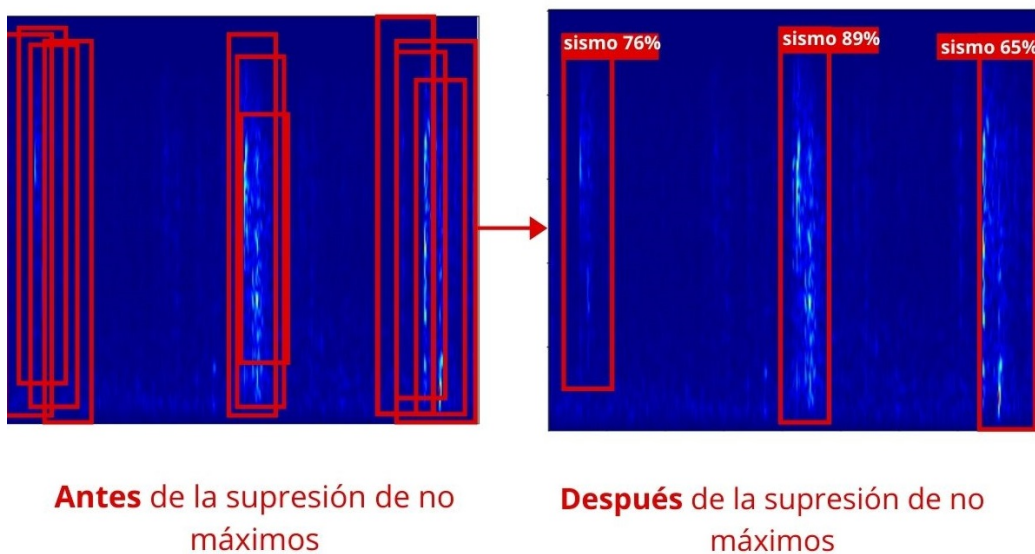


Figura 6.2: Ejemplo del antes y después de la supresión de no máximos. Después de la SNM sólo el rectángulo delimitador que cuente con la probabilidad más alta de ser sismo se mantendrá y el resto se eliminará.

6.1. Postproceso: Supresión de no máximos

La supresión de no máximos consiste en tomar, para un objeto en una imagen, todos los rectángulos delimitadores que la CNN con la ventana deslizante detectó al

momento de recorrer la imagen. De este conjunto de rectángulos se elige el que tiene la probabilidad más alta y se eliminan (suprimen) los demás. Después de hacer ese proceso, en vez de tener múltiples detecciones para el mismo objeto, sólo quedará el rectángulo que delimita mejor al objeto (Figura 6.2).

Para el caso de sismos, se tiene una traza de 6 días y debido a que la señal es continua en el tiempo, se puede ir iterando de un punto inicial t_0 a un punto final t_f dentro de nuestra traza continua.

El postproceso se llevó a cabo de la siguiente manera:

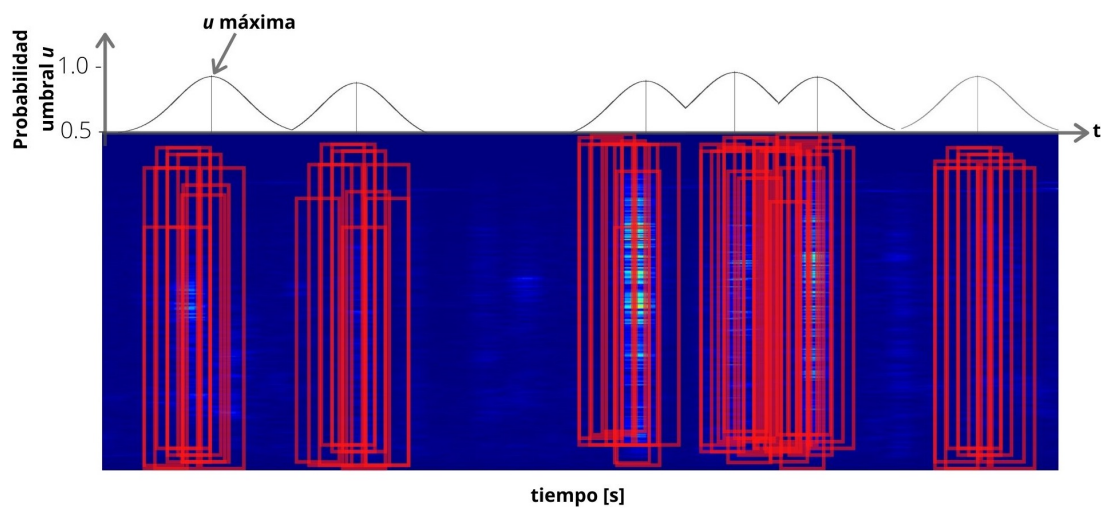


Figura 6.3: Probabilidades del conjunto de rectángulos delimitadores para cada sismo, donde se observa una distribución normal de la distribución numérica de los datos.

1. Se descartan todos los rectángulos que presenten una probabilidad $u < 0.5$
2. Se obtienen todos los rectángulos que pertenecen a un mismo sismo; estos son todos los rectángulos delimitadores para un objeto en la imagen del espectrograma donde existe una probabilidad $u \geq 0.5$.
3. Una vez obtenidos estos rectángulos, se elige aquel que tenga la probabilidad (umbral) más alta y se suprimen los demás, mediante el proceso descrito en el punto siguiente.
4. Suponiendo una distribución normal a partir de la distribución numérica de la probabilidad u , se observa el cambio en la probabilidad que se va presentando en

los rectángulos donde existe sismo (umbral $u \geq 0.5$), por lo que el rectángulo que delimita mejor al sismo se encuentra en el valor máximo de la campana de Gauss (Figura 6.3).

5. Se repite este proceso para todos los sismos detectados al recorrer la ventana deslizante a través de toda la imagen.

El algoritmo 2 muestra el procedimiento completo.

Algoritmo 2: Algoritmo de supresión de no máximos

Resultado: Una lista con los rectángulos delimitadores de los sismos detectados, después de aplicar la supresión de no máximos

Entradas: *detecciones*

detecciones <- Una lista con las probabilidades de cada una de las múltiples detecciones que realizó la CNN;

sismos <- Una lista vacía donde se guardarán las probabilidades de los sismos después de aplicar la SNM;

```

1 /* Realizar búsqueda para obtener todas las ventanas que pertenecen a un
   mismo sismo, tomar la de umbral más alto y suprimir las demás */
2 prob_max <- 0.0, probabilidad máxima de que un evento sea un sismo;
3 para sismo en detecciones hacer
4   | p_eq <- Probabilidad de ser sismo;
5   | si p_eq >= prob_max entonces
6     |   prob_max = p_eq;
7   | en otro caso
8     |   sismos <- Anexar sismo con el último valor máximo encontrado
9   | fin
10 fin

```

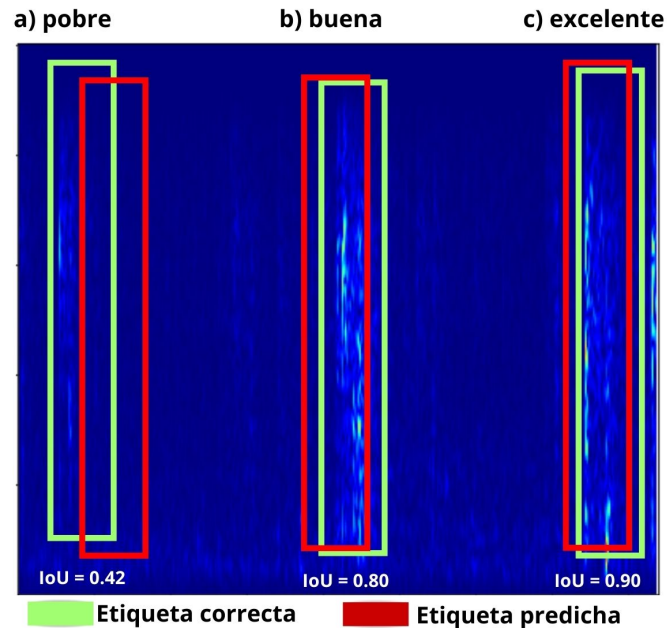


Figura 6.4: Ejemplo de aplicar la métrica IoU para cada ventana donde el modelo detecta un sismo.

Después de la SNM se calculó el traslape de la ventana donde el analista etiqueta el sismo y la que encontró el modelo, utilizando la métrica $IoU \geq 0.8$ (Figura 6.4).

Al hacer una primera prueba con un umbral $u \geq 0.5$ y un valor $IoU \geq 0.8$ se obtuvo un 98 % de exactitud y 5,023 nuevos **posibles sismos**¹.

Debido a que la cantidad de nuevos posibles sismos es del orden de miles, se repitió el cálculo en incrementos de 0.01 para $u \in [0.5, 0.85]$ (Figura 6.5), con la finalidad de disminuir el número de éstos. En la figura se puede observar que el valor de u resulta inversamente proporcional a la exactitud, al igual que con relación al número de los nuevos posibles sismos que encontró el modelo.

Dado que la mayoría de los sismos del conjunto de datos positivo presenta una baja tasa señal a ruido, dificultando en gran medida su detección, en conjunto con el analista se estableció que una exactitud del 90 % es un resultado muy competitivo para el modelo propuesto. De esta manera, según los resultados mostrados en la gráfica superior de la Figura 6.4, se decidió tomar una $u = 0.76$.

Con ese valor de u , el modelo encontró 1,651 sismos de los 1,844 etiquetados

¹Se define como *posible sismo* a las nuevas detecciones que hizo el modelo y que no fueron etiquetadas como tales por el analista.

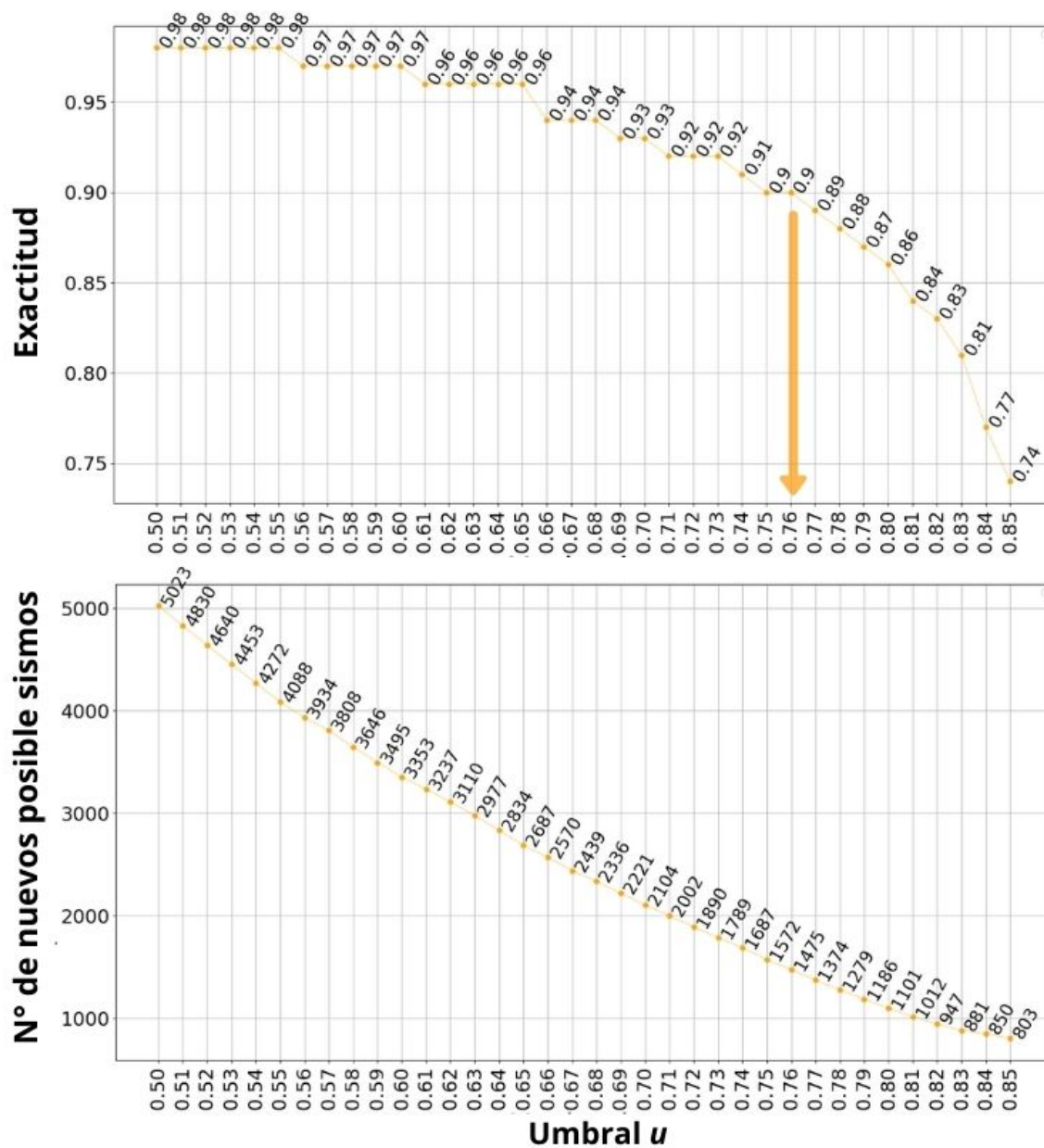


Figura 6.5: Resultados de la segunda fase (la detección de sismos) del modelo propuesto, variando el umbral de probabilidad de detección u .

por el analista; así también encontró 1,475 nuevos posibles sismos que superaban el umbral $u \geq 0.76$.

Para analizar estas nuevas detecciones, se generaron 1,475 espectrogramas, los cuales se tuvieron que clasificar a mano para determinar si efectivamente cada una de estas imágenes contenía un sismo. Después de ser revisadas por el analista, se determinó que un total de 1,340 ventanas son casos de tipo TP, es decir, sismos; y que, además, 135 son de tipo FP (no sismos).

Los FP en su mayoría presentan una baja tasa señal a ruido por lo que se encuentran en el límite de las capacidades de detección del algoritmo. En algunos casos, se pudieron identificar de manera manual como sismos de baja frecuencia o bien, un tipo de evento sísmico conocido como **tremor**² que, para fines de este trabajo, se toma como un evento distinto a un sismo.

De lo anterior, se tiene en la traza de 6 días que:

$$sismos = 1,844 + 1,475 = 3,319$$

$$FN = 1,844 - 1,651 = 193$$

$$FP = 1,475 - 1,340 = 135$$

$$TN = 0 \quad (\text{solamente se analizan sismos verdaderos})$$

$$TP = sismos - FN - FP = 3,319 - 193 - 135 = 2,991$$

Los datos anteriores se presentan en la matriz de confusión mostrada en la Figura 6.6 y, con ésta, se determinan los valores de las métricas de exactitud, precisión, recall o sensibilidad, F_1 y F_β que se enlistan en la Tabla 6.1.

Debido a que es de interés que el modelo no emita falsas alarmas (resultados FP), se penaliza esta acción utilizando un valor de $\beta = 0.5$. Con esta precaución se calcula la métrica $F_{\beta=0.5}$:

$$F_{0.5} = (1 + 0.5^2) \cdot \frac{0.9568 \cdot 0.9394}{(0.5^2 \cdot 0.9568) + 0.9394} = \frac{1.25 \cdot 0.8988}{0.2392 + 0.9394} = \frac{1.1235}{1.1786} = 0.9532$$

²Los tremores son un tipo de movimiento sísmico que se atribuye a la migración de fluidos a profundidad. Este movimiento se caracteriza por tener una baja relación señal a ruido y no tiene fases sísmicas (ondas P y S) bien definidas, [Gilbert, 1909].

		Real	
		Sismo	Ruido
Predicha	Sismo	2991	135
	Ruido	193	0

Figura 6.6: Matriz de confusión para los resultados obtenidos en la segunda fase del algoritmo de detección de sismos.

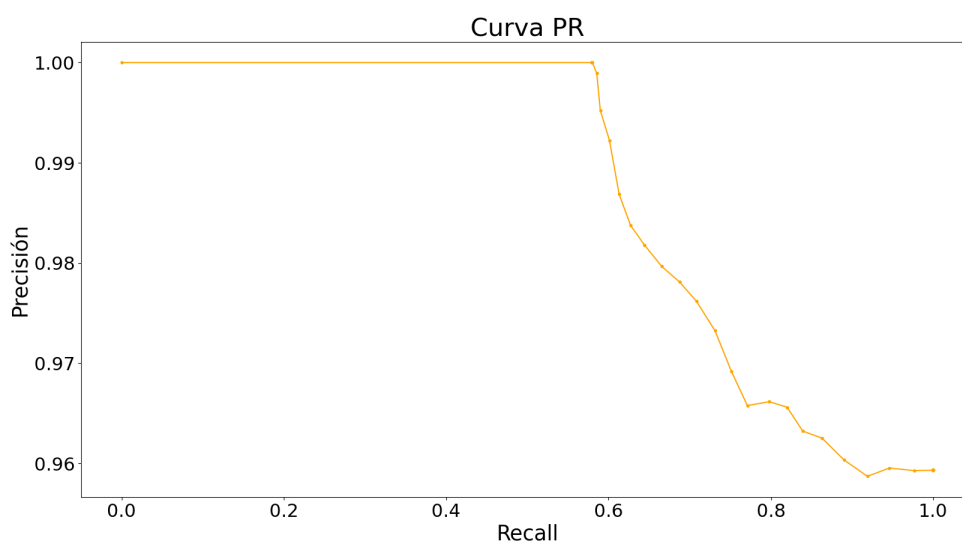


Figura 6.7: Curva PR correspondiente a la segunda fase del modelo propuesto.

Métrica	Valor
exactitud	90.12 %
precisión	95.68 %
recall	93.94 %
F_1	94.80 %
$F_{0.5}$	95.32 %
mAP	98.74 %

Tabla 6.1: Métricas de evaluación del modelo de detección

El mAP se obtuvo calculando el AUC de la curva PR (Figura 6.7), donde se observa un AUC-PR = 98.74 %. Para la detección de sismos el AUC-PR es igual al AP, debido a que solamente se tiene una clase de objetos a clasificar (sismos), por lo tanto, el $mAP = 98.74\%$.

Modelo	Tarea	RNA	Acc (%)	Pre (%)	Rec (%)	Conjunto de datos	Magnitud
A	C	CNN	99.00	99.00	96.00	273, 882	$[-0.81, 5.7]$
B	C	CNN	97.00	nsr	nsr	276, 163	nsr
C	C	CNN	99.50	nsr	99.30	1, 320, 000	$[3, 9.1]$
P	C	CNN	99.37	98.76	100.0	31, 013	≤ 3.5
D	D	CNN	99.00	94.80	100.0	702, 748	$[-0.2, 5.8]$
E	D	Hibrida	97.50	95.70	98.70	39, 648	$[0, 6]$
P	D	CNN	90.12	95.68	93.94	1, 844	≤ 3.5

Tabla 6.2: Desempeño de los distintos modelos de clasificación y detección de sismos: A. [Ross et al., 2018], B. [Zhou et al., 2019], C. [Meier et al., 2019], D. [Perol et al., 2018], E. [Zhu et al., 2018] y P. Modelo propuesto en esta tesis. Donde $nsr =$ no se reporta. Tarea: C - Clasificación y D - Detección. Acc - Exactitud. Pre - Precisión. Rec - Recall.

La Tabla 6.2 presenta los resultados reportados en los artículos del estado del arte (ver Sección 1.1.1), así como los obtenidos en este trabajo de investigación. Cabe mencionar que la columna “Conjunto de datos” incluida en dicha tabla, hace referencia a la cantidad de instancias o muestras utilizadas (información de sismos) en cada uno de los desarrollos y que en este trabajo no estamos haciendo una comparación directa sino, más bien, lo que se busca es mostrar que se obtuvieron resultados similares pese a tener un conjunto de datos más pequeño. Para poder realizar dicha comparativa, se debe estar en condiciones iguales, es decir, emplear el mismo conjunto de datos, los

mismos recursos de cómputo y el mismo algoritmo.

Los resultados mostrados en la Tabla 6.2 hacen ver que el algoritmo de dos fases propuesto es muy competitivo, dado que sus resultados se encuentran en niveles de desempeño semejantes a los resultados de los otros modelos. Además, los conjuntos de datos positivo (sismos) y negativo (no sismos) se encuentran en desbalance, en virtud de que hay una cantidad de señales de sismos mucho menor que las señales de no sismos. Por tal motivo, la exactitud resulta una métrica sesgada que puede ser engañosa, situación por la cual se decidió presentar el uso de métricas adicionales para aumentar el nivel de confianza en la evaluación de los modelos, tales como la métrica F_{beta} y el mAP.

La propuesta de modelo presentada en este trabajo tiene un desempeño superior en la solución del problema de clasificación que el correspondiente a la detección. La diferencia se debe al enfoque de esta investigación para la detección de sismos con una tasa señal a ruido baja, el cual es un problema más difícil de discriminación. Aunado a lo anterior, el tamaño del conjunto de datos es muy pequeño, con relación a los que se suelen emplear en este tipo de modelos (ver Tabla 6.2).

Finalmente, el desempeño de los modelos de aprendizaje profundo es lineal con relación a la cardinalidad (tamaño) del conjunto de datos de entrenamiento con el que se cuenta [Ng, 2017]. Bajo esta premisa se puede decir que, a pesar de utilizar un conjunto de datos que en cantidad es inferior al de los modelos del estado del arte analizados (10 % aproximadamente); y aunado a que el objetivo de esta investigación es la detección de sismos que presentan una baja tasa señal a ruido, el modelo es muy competitivo y cumple con la detección de forma satisfactoria. Más aún, incluso podría implementarse en alguna estación sísmica y usarse como apoyo en la determinación de la ocurrencia de este tipo de eventos.

En el siguiente y último capítulo se describen las conclusiones y el posible trabajo futuro que se puede realizar con los datos para obtener nuevas características, como la duración real del sismo y los tiempos de llegada de las ondas P y S (picado de ondas).

Capítulo 7

Conclusiones y trabajo futuro

El modelo propuesto en la presente investigación de tesis logra obtener resultados competitivos en comparación a los propuestos en otras investigaciones, aún y cuando el problema resuelto de detectar sismos con baja tasa señal a ruido es muy complejo. Más aún, la dificultad se incrementa al contar con un conjunto de datos relativamente pequeño. Una comparativa directa de rendimiento de los modelos se pretende realizar como trabajo futuro para contender en las mismas condiciones, tanto del conjunto de datos como recursos de cómputo.

Vista la cantidad de nuevos **posibles** sismos encontrados (1,475 de los cuáles 1,340 efectivamente fueron sismos, con una tasa de exactitud del 90%), se intuye que al crear el conjunto negativo se incluyeran señales sísmicas como ruido, lo que pudo haber contribuido a que el modelo propuesto se confundiera.

Como trabajo futuro es recomendable hacer una revisión más detallada de los casos FP (en ambas fases del modelo), dado que algunos de éstos, al momento de hacer el análisis manualmente, presentan tipos de onda similares a las ondas P y S, por lo que para confirmar que correspondan a sismos (o no) es necesario verificar su arribo empleando las tres componentes (N, E y Z).

Por otra parte, se sugiere probar con ventanas de distintos tamaños, diferentes a los 30s y con saltos mayores a 1s, lo que además de reducir el tiempo de ejecución, se cree podría ayudar a ajustar el tamaño de las ventanas a los sismos, con la finalidad de obtener la duración real de éstos.

Los datos con que se cuenta contienen, además de las ventanas de tiempo de 30s, los tiempos de llegada (picado) de las ondas P y S, por lo que añadir otro proceso al algoritmo podría ayudar a encontrar un picado de ondas ante nuevos eventos. Además, se debe probar el algoritmo sobre una traza sísmica proveniente de otras estaciones, para ver qué tan bien generaliza ante sismos de otras regiones.

Por último, el problema planteado también se puede intentar resolver con algoritmos de una fase, que se sabe son más rápidos, pero que no son tan precisos a la hora de la detección. En el estado del arte se encuentra las redes convolucionales basadas en regiones [Girshick et al., 2014] (mejor conocida como R-CNN, por sus siglas en inglés) y YOLO [Redmon et al., 2016, Redmon and Farhadi, 2017, Redmon and Farhadi, 2018], el cual es uno de los algoritmos más utilizados (en la parte de visión) para la conducción autónoma de vehículos.

Apéndices

Apéndice A

Teorema del muestreo de Nyquist-Shannon

El teorema de muestreo de Nyquist-Shannon afirma que una señal de banda, con una frecuencia máxima de F_c [Hz], puede ser reconstruida a partir de sus muestras si la velocidad de muestreo es mayor a $2F_c$ muestras por segundo [Vaseghi, 2008].

En la Figura A.1 se calculan los valores máximos y mínimos de la frecuencia de la señal sísmica obtenida en la estación sismológica de Pinotepa Nacional, Oaxaca; del primer segundo del día 1° de enero de 2012 hasta el último segundo del día 20 de abril de 2012, a una tasa de 100 muestras por segundo (mps) con el fin de observar su comportamiento y probar que la tasa de muestreo que se usa en dicha estación es la correcta, según el teorema de Nyquist-Shannon.

Para corroborar que la tasa de muestreo en los datos es la ideal, se graficó la frecuencia de la señal por mes (Figura A.1), tomando como valor máximo F_c y dado que la tasa de muestreo es mayor a $2F_c$, para todos los casos de enero a abril, se concluye que nuestra tasa de muestreo de 100 mps es la correcta.

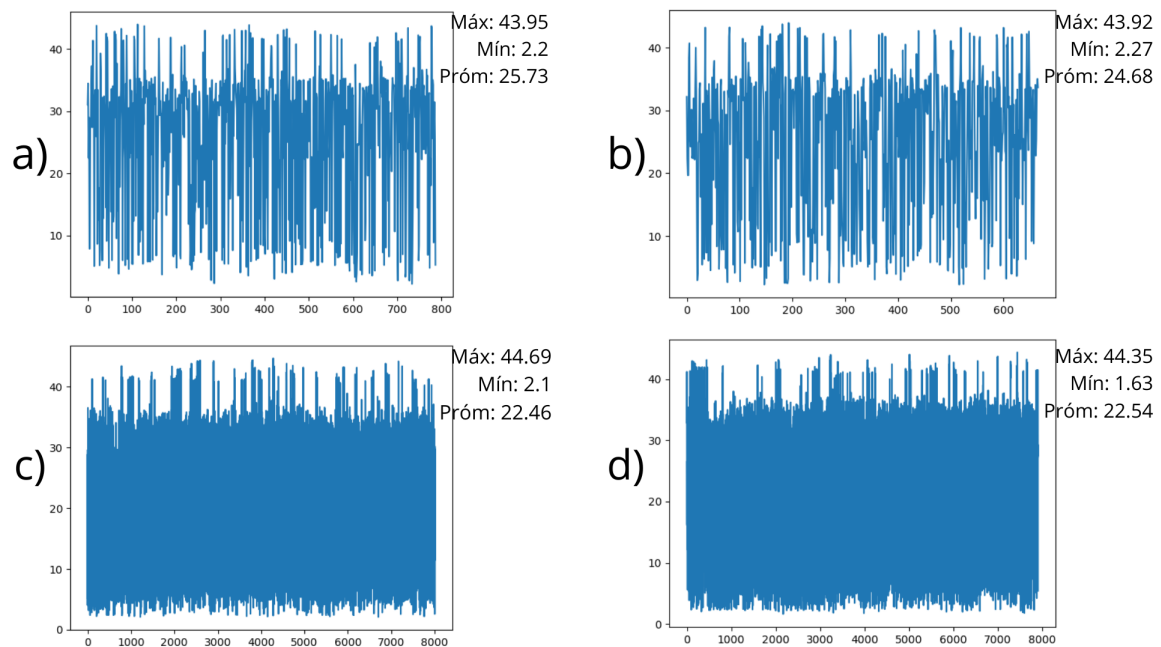


Figura A.1: Medidas estadísticas descriptivas de la señal sísmica obtenida por la estación sismológica de Pinotepa Nacional, Oaxaca; entre el 1° de enero y el 20 de abril de 2012, a 100 mps. a) enero, b) febrero, c) marzo y d) abril.

Apéndice B

Conceptos básicos sobre redes neuronales

B.1. Función de pérdida o de error

Esta es una función que calcula el error que emite el modelo ante la predicción que está haciendo, en otras palabras, esta función indica qué tan bien o mal fue la predicción del modelo. El *-Mean Square Error (MSE)-* [Willmott and Matsuura, 2005] es la más común para problemas de regresión y la entropía cruzada [Shore and Johnson, 1981] es muy común para problemas de clasificación.

B.2. Optimizador

El optimizador es un algoritmo usado para encontrar los pesos de las neuronas que minimizan la función de error. Existen múltiples algoritmos que se pueden usar para optimizar al momento de entrenar un modelo como: *-Batch Gradient Descent (BGD)-* [Yuan et al., 2016], *Stochastic GD* [Amari, 1993] y *Adam* [Kingma and Ba, 2014].

B.3. Tasa de aprendizaje

Una de las entradas del optimizador que se puede ajustar es el ritmo de aprendizaje o *-learning rate (lr)-*. Teóricamente, mientras más pequeño sea el *lr* se garantiza encontrar el error mínimo (si se tuviera un tiempo infinito), en cambio, si se usa un *lr* muy grande, aumentará la rapidez con la que se encuentra un mínimo, pero no se garantiza que sea el error mínimo o mínimo global.

B.4. Tamaño de lote

Este es un número entero de instancias (un lote) que se van a tomar del conjunto de entrenamiento, con el fin de entrenar al modelo con estos lotes en cada época o iteración y no con todo el conjunto completo. En la práctica se ha visto que el tamaño de lote proporciona mejores resultados si es múltiplo de dos, generalmente se usa 32, 64, 128 ó 256 [Géron, 2019].

B.5. Conjunto de validación

El conjunto de validación es un porcentaje del conjunto de entrenamiento, de este modo, del conjunto de entrenamiento se obtienen dos subconjuntos, uno para entrenar y otro para validar el modelo. Generalmente se usa un porcentaje de 10 %, 20 % ó 30 %.

B.6. Número de épocas

El número de veces que el conjunto de entrenamiento será mostrado a la red neuronal mientras se entrena. Un número grande de épocas provocará que la exactitud del conjunto de validación comience a decrecer y el de entrenamiento a aumentar (overfitting¹).

¹Ocurre cuando el modelo está aprendiendo exclusivamente las particularidades de los datos de entrenamiento y no generaliza bien ante un conjunto de datos diferente.

B.7. Dropout

Es una técnica para mejorar el desempeño de los modelos de redes neuronales profundas, fue propuesta por [Hinton et al., 2012] y consiste en tomar de forma aleatoria en cada época o iteración un porcentaje de una capa de neuronas y desactivarlas. Esto es poco intuitivo, pero ayuda a disminuir el sobreajuste al obligar a aprender de forma más generalizada a las neuronas que no se desactivaron. Por lo común se usa un dropout de 10, 20 ó 30 %.

Bibliografía

- [Allen, 1978] Allen, R. V. (1978). Automatic earthquake recognition and timing from single traces. *Bulletin of the seismological society of America*, 68(5):1521–1532.
- [Amari, 1993] Amari, S.-i. (1993). Backpropagation and stochastic gradient descent method. *Neurocomputing*, 5(4-5):185–196.
- [Barón de Fourier, 1822] Barón de Fourier, J. B. J. (1822). *Théorie analytique de la chaleur*. Firmin Didot.
- [Bennett et al., 2007] Bennett, J., Lanning, S., et al. (2007). The netflix prize. In *Proceedings of KDD cup and workshop*, volume 2007, page 35. Citeseer.
- [Brigham, 1988] Brigham, E. O. (1988). *The fast Fourier transform and its applications*. Prentice-Hall, Inc.
- [Brodsky, 2019] Brodsky, E. E. (2019). The importance of studying small earthquakes. *Science*, 364(6442):736–737.
- [Brown et al., 2020] Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. (2020). Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.
- [Cooley and Tukey, 1965] Cooley, J. and Tukey, J. (1965). An Algorithm for the Machine Calculation of Complex Fourier Series. *Mathematics of Computation*, 19(90):297–301.

- [(devs@obspy.org), 2020] (devs@obspy.org), T. O. D. T. (2020). *obspy.core.trace.Trace.detrend*. Accesado Dic 10 2020, <https://docs.obspy.org/packages/autogen/obspy.core.trace.Trace.detrend.html>.
- [Elgendy, 2020] Elgendy, M. (2020). *Deep Learning for Vision Systems*. Manning Publications.
- [Géron, 2019] Géron, A. (2019). *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. O'Reilly Media.
- [Gilbert, 1909] Gilbert, G. K. (1909). Earthquake forecasts. *Science*, 29(734):121–138.
- [Girshick, 2015] Girshick, R. (2015). Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448.
- [Girshick et al., 2014] Girshick, R., Donahue, J., Darrell, T., and Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587.
- [Goodfellow et al., 2014] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. *Advances in neural information processing systems*, 27.
- [Gulli and Pal, 2017] Gulli, A. and Pal, S. (2017). *Deep learning with Keras*. Packt Publishing Ltd.
- [Gurovich, 2008] Gurovich, E. V. (2008). *Introducción a la Teoría de la Computación*. UNAM.
- [Gutenberg, 1956] Gutenberg, B. (1956). The energy of earthquakes. *Quarterly Journal of the Geological Society*, 112(1-4):1–14.

- [Hanks and Kanamori, 1979] Hanks, T. C. and Kanamori, H. (1979). A moment magnitude scale. *Journal of Geophysical Research: Solid Earth*, 84(B5):2348–2350.
- [Helfrich and Wood, 2001] Helfrich, G. R. and Wood, B. J. (2001). The earth’s mantle. *Nature*, 412(6846):501–507.
- [Hinton et al., 2012] Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. R. (2012). Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*.
- [Ho, 1995] Ho, T. K. (1995). Random decision forests. In *Proceedings of 3rd International Conference on Document Analysis and Recognition*, volume 1, pages 278–282 vol.1.
- [Hochreiter and Schmidhuber, 1997] Hochreiter, S. and Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780.
- [Hubel, 1959] Hubel, D. H. (1959). Single unit activity in striate cortex of unrestrained cats. *The Journal of physiology*, 147(2):226.
- [Hubel and Wiesel, 1968] Hubel, D. H. and Wiesel, T. N. (1968). Receptive fields and functional architecture of monkey striate cortex. *The Journal of physiology*, 195(1):215–243.
- [Husker et al., 2008] Husker, A., Stubailo, I., Lukac, M., Naik, V., Guy, R., Davis, P., and Estrin, D. (2008). Wilson: The wirelessly linked seismological network and its application in the middle american subduction experiment. *Seismological Research Letters*, 79(3):438–443.
- [IRIS, 2020] IRIS (2020 (accesado Julio 2, 2020)). *The cumulative volume of seismic data archived at the IRIS Data Management Center, 2020*.
- [IRIS-SAC, 2017] IRIS-SAC (2017). *What is the SAC file format?* Accesado 10 Diciembre, 2020. <https://ds.iris.edu/ds/support/faq/17/sac-file-format/>.

- [Jordan and Mitchell, 2015] Jordan, M. I. and Mitchell, T. M. (2015). Machine learning: Trends, perspectives, and prospects. *Science*, 349(6245):255–260.
- [Jung, 1910] Jung, C. G. (1910). The association method. *The American journal of psychology*, 21(2):219–269.
- [Kearey et al., 2013] Kearey, P., Brooks, M., and Hill, I. (2013). *An introduction to geophysical exploration*. John Wiley & Sons.
- [Kingma and Ba, 2014] Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
- [Krizhevsky et al., 2012] Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105.
- [Lay and Wallace, 1995] Lay, T. and Wallace, T. C. (1995). *Modern global seismology*. Elsevier.
- [LeCun et al., 1989] LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., and Jackel, L. D. (1989). Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551.
- [Liu et al., 2016] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., and Berg, A. C. (2016). Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer.
- [Lukac et al., 2009] Lukac, M., Davis, P., Clayton, R., and Estrin, D. (2009). Recovering temporal integrity with data driven time synchronization. In *2009 International Conference on Information Processing in Sensor Networks*, pages 61–72. IEEE.
- [McClelland et al., 1986] McClelland, J. L., Rumelhart, D. E., Group, P. R., et al. (1986). *Parallel distributed processing*, volume 2. MIT press Cambridge, MA.

- [McCulloch and Pitts, 1943] McCulloch, W. S. and Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133.
- [Meier et al., 2019] Meier, M., Ross, Z. E., Ramachandran, A., Balakrishna, A., Nair, S., Kundzicz, P., Li, Z., Andrews, J., Hauksson, E., Yue, Y., and et al. (2019). Reliable real-time seismic signal/noise discrimination with machine learning. *Journal of Geophysical Research: Solid Earth*, 124(1):788–800.
- [Melgar and Pérez-Campos, 2011] Melgar, D. and Pérez-Campos, X. (2011). Imaging the Moho and subducted oceanic crust at the Isthmus of Tehuantepec, Mexico, from receiver functions. *Pure and applied geophysics*, 168(8-9):1449–1460.
- [Mousa et al., 2011] Mousa, W. A., Al-Shuhail, A. A., and Al-Lehyani, A. (2011). A new technique for first-arrival picking of refracted seismic data based on digital image segmentation. *Geophysics*, 76(5):V79–V89.
- [Ng, 2017] Ng, A. (2017). Machine learning yearning. *URL: [http://www. mlyearning.org/\(96\)](http://www.mlyearning.org/(96))*.
- [Nunez-Cornu, 1996] Nunez-Cornu, F. J. (1996). A double seismic front and earthquake cycles along the coast of Oaxaca, Mexico. *Seismological Research Letters*, 67(6):33–39.
- [Nussbaumer, 1981] Nussbaumer, H. J. (1981). The fast Fourier transform. In *Fast Fourier Transform and Convolution Algorithms*, pages 80–111. Springer.
- [Pearl, 1936] Pearl, R. (1936). Karl Pearson, 1857–1936. *Journal of the American Statistical Association*, 31(196):653–664.
- [Pedregosa et al., 2011] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al. (2011). Scikit-learn: Machine learning in Python. *the Journal of machine Learning research*, 12:2825–2830.

- [Pérez-Campos et al., 2018] Pérez-Campos, X., Espíndola, V. H., Pérez, J., Estrada, J. A., Monroy, C. C., Bello, D., González-López, A., Ávila, D. G., Esparza, M. G. C. R., Maldonado, R., et al. (2018). The mexican national seismological service: An overview. *Seismological Research Letters*, 89(2A):318–323.
- [Perol et al., 2018] Perol, T., Gharbi, M., and Denolle, M. (2018). Convolutional neural network for earthquake detection and location. *Science Advances*, 4(2):e1700578.
- [Priemer, 1991] Priemer, R. (1991). *Introductory signal processing*, volume 6. World Scientific.
- [Rao and Swamy, 2018] Rao, K. D. and Swamy, M. N. (2018). *Digital signal processing: theory and practice*. Springer.
- [Redmon et al., 2016] Redmon, J., Divvala, S., Girshick, R., and Farhadi, A. (2016). You Only Look Once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788.
- [Redmon and Farhadi, 2017] Redmon, J. and Farhadi, A. (2017). YOLO9000: better, faster, stronger. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7263–7271.
- [Redmon and Farhadi, 2018] Redmon, J. and Farhadi, A. (2018). YOLOv3: An incremental improvement. *arXiv preprint arXiv:1804.02767*.
- [Ren et al., 2015] Ren, S., He, K., Girshick, R., and Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99.
- [Roller et al., 2020] Roller, S., Dinan, E., Goyal, N., Ju, D., Williamson, M., Liu, Y., Xu, J., Ott, M., Shuster, K., Smith, E. M., et al. (2020). Recipes for building an open-domain chatbot. *arXiv preprint arXiv:2004.13637*.
- [Rosenblatt, 1958] Rosenblatt, F. (1958). The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386.

- [Ross et al., 2018] Ross, Z. E., Meier, M., Hauksson, E., and Heaton, T. H. (2018). Generalized seismic phase detection with deep learning. *Bulletin of the Seismological Society of America*, 108(5A):2894–2901.
- [Ruder, 2016] Ruder, S. (2016). An overview of gradient descent optimization algorithms. arxiv 2016. *arXiv preprint arXiv:1609.04747*.
- [Sabbione and Velis, 2010] Sabbione, J. I. and Velis, D. (2010). Automatic first-breaks picking: New strategies and algorithms. *Geophysics*, 75(4):V67–V76.
- [Samuel, 1959] Samuel, A. (1959). Some studies in machine learning using the game of checkers. *IBM Journal of Research and Development*, 3(3):210–29.
- [Schmidhuber, 2015] Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural Networks*, 61:85–117.
- [Seismology Lab, 2010] Seismology Lab, B. (17 abril 2010). *Today in Earthquake History: Japan/Potsdam 1889*.
- [Shore and Johnson, 1981] Shore, J. and Johnson, R. (1981). Properties of cross-entropy minimization. *IEEE Transactions on Information Theory*, 27(4):472–482.
- [Silver et al., 2016] Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., et al. (2016). Mastering the game of Go with deep neural networks and tree search. *nature*, 529(7587):484.
- [Sipe et al., 2013] Sipe, M. A., Schneiderman, H. W., and Nechyba, M. C. (2013). Facial recognition. US Patent 8,457,367.
- [SSN, 2018] SSN (2018). *Catálogo de sismos del SSN*. Accesado Julio 2, 2020. <http://www2.ssn.unam.mx:8080/catalogo/>.
- [SSN, 2020] SSN, Instituto de Geofísica, U. N. A. M. (2020 (accesado Junio 18, 2020)). *Red de estaciones sismológicas UNAM México*.

- [Stanley et al., 1988] Stanley, W. D., Dougherty, G. R., Dougherty, R., and Saunders, H. (1988). Digital signal processing. *Reston Pub Co; Edición 2 Sub*.
- [Tibi et al., 2019] Tibi, R., Linville, L., Young, C., and Brogan, R. (2019). Classification of local seismic events in the utah region: A comparison of amplitude ratio methods with a spectrogram-based machine learning approach. *Bulletin of the Seismological Society of America*, 109(6):2532–2544.
- [Vaseghi, 2008] Vaseghi, S. V. (2008). *Advanced digital signal processing and noise reduction*. John Wiley & Sons.
- [Watkins et al., 2018] Watkins, W. D., Thurber, C. H., Abbott, E. R., and Brudzinski, M. R. (2018). Local earthquake tomography of the Jalisco, Mexico region. *Tectonophysics*, 724:51–64.
- [Willmott and Matsuura, 2005] Willmott, C. J. and Matsuura, K. (2005). Advantages of the mean absolute error (mae) over the root mean square error (rmse) in assessing average model performance. *Climate research*, 30(1):79–82.
- [Yoo and Han, 2009] Yoo, J.-C. and Han, T. H. (2009). Fast normalized cross-correlation. *Circuits, systems and signal processing*, 28(6):819–843.
- [Yuan et al., 2016] Yuan, K., Ying, B., Vlaski, S., and Sayed, A. H. (2016). Stochastic gradient descent with finite samples sizes. In *2016 IEEE 26th International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 1–6. IEEE.
- [Zhang, 2019] Zhang, D. (2019). *Fundamentals of Image Data Mining*. Springer.
- [Zhou et al., 2019] Zhou, Y., Yue, H., Kong, Q., and Zhou, S. (2019). Hybrid event detection and phase-picking algorithm using convolutional and recurrent neural networks. *Seismological Research Letters*, 90(3):1079–1087.
- [Zhu et al., 2018] Zhu, L., Peng, Z., and McClellan, J. (2018). Deep learning for seismic event detection of earthquake aftershocks. *2018 52nd Asilomar Conference on Signals, Systems, and Computers*.