



Universidad Nacional Autónoma de México  
Posgrado en Ciencia e Ingeniería de la Computación

*Análisis de Sismogramas mediante Redes  
Neuronales Recurrentes*

**TESIS**

Que para optar por el grado de:  
Maestro en Ciencia e Ingeniería de la Computación

PRESENTA:

Luis Alfonso Hernández Vanegas

DIRECTORA DE TESIS:

Dra. Suemi Rodríguez Romo  
Facultad de Estudios Superiores Cuautitlán, UNAM

Ciudad Universitaria, CD. MX.

Agosto, 2021



Universidad Nacional  
Autónoma de México

Dirección General de Bibliotecas de la UNAM

**Biblioteca Central**



**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.



*AGRADECIMIENTOS.*

*A mis padres, hermanas y personas cercanas, familia y amigos,  
que han influido en mi vida a través del tiempo,  
que con su apoyo y palabras de aliento,  
definitivamente contribuyeron de forma invaluable  
a la finalización de este proyecto.*



---

# Reconocimientos

A mis profesores, Dr. Óscar Ibáñez, Dr. Ricardo Paramont (q.e.p.d) y al Maestro Felipe Vargas por su disponibilidad y deseo de colaborar con sus alumnos, en especial a mi tutora de tesis, la Dra. Suemi Rodríguez, por sugerirme un tema interesante para la elaboración del proyecto de tesis, no solo por eso, sino también por su paciencia y valiosos comentarios durante el desarrollo de este proyecto de tesis.

A mis sinodales de tesis: Dra. Wendy Aguilar Martínez, Dra. María Elena Martínez Pérez, Dra. Katya Rodríguez Vázquez y Dr. Sergio Rogelio Marcellin Jacques. Les agradezco el haber aceptado ser parte de mi jurado de tesis.

A la Universidad Nacional Autónoma de México por la oportunidad de desarrollarme como profesionista.

Al posgrado de Ciencia e Ingeniería de la Computación por tenderme la oportunidad de ir un paso más allá en mis estudios como profesionista.

Al proyecto PAPIIT IT102920 por el apoyo para la realización de este proyecto de tesis.

Al Centro de Investigaciones Teóricas de la Facultad de Estudios Superiores Cuautitlán por las facilidades para desarrollar nuestras actividades escolares.

Al Consejo Nacional de Ciencia y Tecnología (CONACyT) por la beca que me otorgaron.

Parte de los datos sísmológicos que se utilizaron para la realización de este proyecto fueron obtenidos por el *Servicio Sísmológico Nacional* (México). Agradezco a todo el personal del mismo por el mantenimiento de las estaciones, la adquisición y distribución de los datos.

Otra parte de los datos sísmológicos fueron obtenidos a través de *IRIS Wilber 3 system*. Agradezco la facilidad y rapidez con la cual se puede acceder a los registros sísmológicos.



---

# Resumen

Un evento sísmico es consecuencia de energía liberada durante interacciones entre placas tectónicas, erupciones volcánicas, explosiones provocadas por el hombre, cuerpos celestes que impactan en la tierra, hundimientos, etc. Dicha energía se libera en diversas formas, una de ellas es en forma de energía mecánica la cual a su vez da origen a las diversas ondas sísmicas que se propagan a través de la tierra las cuales se registran en sismogramas mediante sensores conocidos como: sismómetros y acelerómetros. Al registro de lecturas de velocidades ó aceleraciones (según el sensor que se use) del terreno provocadas por las ondas sísmicas a través del tiempo se le conoce como sismogramas.

Como primer paso en este proyecto de tesis se recolectaron sismogramas de eventos sísmicos originados alrededor del mundo y en una extensión de tiempo que va de la década de 1990 hasta fechas actuales, esto gracias al acceso que se tuvo las bases de datos de sismogramas en formato digital del *SSN*<sup>1</sup> y de *IRIS*<sup>2</sup>. El segundo paso fue extraer las lecturas contenidas en los sismogramas descargados para formar series de tiempo y, posteriormente, hacer uso de redes neuronales recurrentes como elemento principal, para intentar predecir valores futuros de dichas series de tiempo apoyándose en valores pasados de las mismas, es decir, se planteará un problema de aprendizaje supervisado para realizar predicciones en las series de tiempo extraídas de diversos sismogramas. Adicionalmente, se hará uso de algunos tipos de redes neuronales como apoyo a las redes neuronales recurrentes, entre ellas: redes neuronales convolucionales unidimensionales y mecanismos de atención. La dificultad de analizar estas series de tiempo generadas por sismogramas mediante redes neuronales recurrentes está en que dichas series de tiempo no presentan tendencia o estacionalidad, haciéndolas más complicadas de analizar y caracterizar. El reto será encontrar configuraciones de redes neuronales basadas en los modelos de redes neuronales recurrentes existentes (*LSTM* y *GRU*) y alguna selección de hiperparámetros que permita que las redes neuronales propuestas sean capaces de aprender el patrón subyacente que caracteriza y rige el comportamiento de este tipo

---

<sup>1</sup><http://www2.ssn.unam.mx:8080/catalogo/>

<sup>2</sup>[https://ds.iris.edu/wilber3/find\\_event](https://ds.iris.edu/wilber3/find_event)

## Resumen

---

de series de tiempo de tal forma que se obtengan resultados favorables en cuanto a la generalidad y calidad de las predicciones de lecturas recabadas por sismómetros o acelerómetros contenidas en sismogramas de eventos sísmicos ocurridos alrededor del mundo y en distintas fechas. Estos objetivos se cumplieron puesto que existe una física que gobierna el fenómeno de la propagación de las ondas sísmicas.

---

# Índice general

<b>Resumen</b>	<b>I</b>
<b>1. Introducción</b>	<b>1</b>
1.1. Presentación . . . . .	3
1.2. Motivación . . . . .	5
1.3. Objetivo General . . . . .	6
1.3.1. Objetivos particulares . . . . .	6
1.4. Hipótesis . . . . .	6
1.5. Planteamiento del problema . . . . .	7
1.6. Metodología . . . . .	8
1.7. Contribuciones . . . . .	8
1.8. Estructura de la tesis . . . . .	9
<b>2. Marco teórico</b>	<b>11</b>
2.1. Introducción . . . . .	11
2.2. Sismología . . . . .	11
2.2.1. Ecuación de ondas sísmicas . . . . .	11
2.2.2. Ondas sísmicas de cuerpo y superficiales . . . . .	13
2.2.3. Ondas P . . . . .	15
2.2.4. Ondas S . . . . .	16
2.2.5. Ondas Rayleigh . . . . .	18
2.2.6. Ondas Love . . . . .	19
2.2.7. Sismógrafos . . . . .	20
2.2.8. Sismogramas . . . . .	22
2.3. Series de Tiempo . . . . .	24
2.4. Redes Neuronales . . . . .	25
2.4.1. Introducción . . . . .	25
2.4.2. Aprendizaje Profundo ( <i>Deep Learning</i> ) . . . . .	26
2.4.3. Técnicas de aprendizaje automático . . . . .	27
2.4.4. Neurona biológica . . . . .	29
2.4.5. Perceptrón . . . . .	30

## Índice general

---

2.4.6.	Redes neuronales artificiales (RNA)	31
2.4.7.	Entrenamiento de una red neuronal artificial	33
2.4.8.	Propagación hacia atrás ( <i>Back-Propagation</i> )	34
2.4.9.	Redes Neuronales Recurrentes Simples	36
2.4.10.	<i>Long Short-Term Memory</i> (LSTM)	39
2.4.11.	<i>Gated Recurrent Unit</i> (GRU)	42
2.4.12.	Redes neuronales convolucionales de una dimensión ( <i>Conv1D</i> )	46
2.4.13.	Mecanismos de Atención	53
<b>3.</b>	<b>Redes neuronales recurrentes y convolucionales unidimensionales aplicadas a sismogramas para predicción</b>	<b>57</b>
3.1.	Introducción	57
3.2.	Preparación de datos	58
3.2.1.	Ventana deslizante ( <i>Sliding Window</i> ) y Normalización de datos	58
3.3.	Selección de las diferentes configuraciones de redes neuronales recurrentes	60
3.4.	Algoritmos para la optimización de hiperparámetros	61
3.5.	Métricas para evaluar el desempeño de las redes neuronales recurrentes en la predicción de lecturas contenidas en sismogramas	63
3.6.	Resultados obtenidos con la redes neuronales recurrentes	64
3.6.1.	Análisis de sismogramas en la componente espacial Este-Oeste para el terremoto del 19 de septiembre de 2017 en Puebla, México	65
3.6.2.	Análisis de sismogramas en las componentes espaciales Norte-Sur y Vertical para el terremoto del 19 de septiembre de 2017 en Puebla, México	68
3.6.3.	Análisis de sismogramas de eventos sísmicos ocurridos en México, pero, diferentes al que tomó lugar el 19 de septiembre de 2017 en Puebla, México	71
3.6.4.	Análisis de sismogramas de eventos sísmicos ocurridos fuera de México	77
3.6.5.	Comparando resultados de diferentes configuraciones de redes neuronales recurrentes con los obtenidos por las redes Vanilla-LSTM y Vanilla-GRU	87
<b>4.</b>	<b>Análisis de Resultados</b>	<b>91</b>
4.1.	Resumen	91
4.2.	Resultados obtenidos	91
4.2.1.	Acerca de los sismos ocurridos en México que se analizaron con las redes neuronales recurrentes	92

4.2.2. Acerca de los sismogramas analizados pertenecientes a sismos con epicentro fuera del territorio mexicano . . . . .	94
4.2.3. Acerca del desempeño de los modelos de redes neuronales LSTM y GRU . . . . .	95
4.2.4. Acerca de los resultados obtenidos con las diferentes configura- ciones de redes neuronales recurrentes . . . . .	96
<b>5. Conclusiones</b>	<b>99</b>
5.1. Conclusiones . . . . .	99
5.2. Trabajo a futuro . . . . .	100
 <b>Ápéndices</b>	 <b>103</b>
<b>A. Acceso a la base de datos del SSN</b>	<b>103</b>
A.1. Procedimiento . . . . .	103
<b>B. Acceso a la base de datos de IRIS</b>	<b>107</b>
B.1. Procedimiento . . . . .	107



---

# Índice de figuras

2.1.	<i>Capas internas de la Tierra.</i> Imagen del usuario Jeremy Kemp de Wikipedia (CC BY-SA 3.0), recuperada de <a href="https://commons.wikimedia.org/wiki/File:Earth-crust-cutaway-spanish.svg">https://commons.wikimedia.org/wiki/File:Earth-crust-cutaway-spanish.svg</a> .	14
2.2.	<i>Movimiento de las ondas primarias.</i> Adaptado de <i>Passage of P waves through the earth's crust</i> , traducida al idioma español, imagen por Giuseppe Olivadoti, 2001, Analog Dialogue ( <a href="https://www.analog.com/jp/analog-dialogue/articles/sensing-analyzing-and-acting-during-an-earthquake.html#">https://www.analog.com/jp/analog-dialogue/articles/sensing-analyzing-and-acting-during-an-earthquake.html#</a> ) . . . . .	15
2.3.	<i>Movimiento de las ondas secundarias.</i> Adaptado de <i>Passage of S waves through the earth's crust.</i> , traducida al idioma español, imagen por Giuseppe Olivadoti, 2001, Analog Dialogue ( <a href="https://www.analog.com/jp/analog-dialogue/articles/sensing-analyzing-and-acting-during-an-earthquake.html#">https://www.analog.com/jp/analog-dialogue/articles/sensing-analyzing-and-acting-during-an-earthquake.html#</a> ) . . . . .	17
2.4.	<i>Velocidades de las ondas P y S en diferentes materiales</i> [30]. . . . .	18
2.5.	<i>Movimiento de las ondas Rayleigh.</i> Adaptado de <i>Passage of Rayleigh waves through earth's crust.</i> , traducida al español, imagen por Giuseppe Olivadoti, 2001, Analog Dialogue ( <a href="https://www.analog.com/jp/analog-dialogue/articles/sensing-analyzing-and-acting-during-an-earthquake.html#">https://www.analog.com/jp/analog-dialogue/articles/sensing-analyzing-and-acting-during-an-earthquake.html#</a> ) . . . . .	19
2.6.	<i>Movimiento de las ondas Love.</i> Adaptado de <i>Passage of Love waves through earth's crust.</i> , traducida al español, imagen por Giuseppe Olivadoti, 2001, Analog Dialogue ( <a href="https://www.analog.com/jp/analog-dialogue/articles/sensing-analyzing-and-acting-during-an-earthquake.html#">https://www.analog.com/jp/analog-dialogue/articles/sensing-analyzing-and-acting-during-an-earthquake.html#</a> ) . . . . .	20
2.7.	<i>Sismógrafo.</i> Imagen del usuario Yamaguchi de Wikipedia (CC BY-SA 3.0), recuperada de <a href="https://commons.wikimedia.org/wiki/File:Kinematics_seismograph.jpg">https://commons.wikimedia.org/wiki/File:Kinematics_seismograph.jpg</a> . . . . .	21
2.8.	<i>Sismogramas obtenidos de la estación sísmica IGIG en las tres componentes espaciales (Vertical, Norte-Sur y Este-Oeste) para el terremoto del 19 de septiembre de 2017.</i> . . . . .	23

## Índice de figuras

---

2.9. <i>Componentes de una serie de tiempo: Tendencia, estacionalidad y aleatoriedad.</i> . . . . .	25
2.10. <i>Inteligencia Artificial Simbólica y Aprendizaje Máquina.</i> . . . . .	27
2.11. <i>Neurona biológica y neurona artificial.</i> Reproducido de “What is the differences between artificial neural network (computer science) and biological neural network?”, por Abhi S, 2018, ( <a href="https://www.quora.com/What-is-the-differences-between-artificial-neural-network-computer-science-and-biological-neural-network">https://www.quora.com/What-is-the-differences-between-artificial-neural-network-computer-science-and-biological-neural-network</a> ). Obra de Dominio Publico. . . . .	30
2.12. <i>Red neuronal artificial de una sola capa totalmente conectada.</i> . . . .	32
2.13. <i>Red neuronal multicapa alimentada hacia adelante totalmente conectada.</i> Imagen del usuario Gengiskanhg de Wikipedia (CC BY-SA 3.0), recuperada de <a href="https://commons.wikimedia.org/wiki/File:RedNeuronalArtificial.png">https://commons.wikimedia.org/wiki/File:RedNeuronalArtificial.png</a> . . . . .	33
2.14. <i>Comparación de la estructura de una red neuronal recurrente y una alimentada hacia adelante.</i> . . . . .	37
2.15. <i>Red neuronal recurrente extendida en el tiempo.</i> Imagen del usuario MingxianLin de Wikipedia (CC BY-SA 4.0), recuperada de <a href="https://commons.wikimedia.org/wiki/File:RNN.png">https://commons.wikimedia.org/wiki/File:RNN.png</a> . . . . .	38
2.16. <i>Arquitectura Estándar de la Red Neuronal Recurrente Long Short-Term Memory</i> [11]. . . . .	40
2.17. <i>Redes neuronales recurrentes LSTM extendidas en el tiempo.</i> Imagen del usuario MingxianLin de Wikipedia (CC BY-SA 4.0), recuperada de <a href="https://commons.wikimedia.org/wiki/File:LSTM.jpg">https://commons.wikimedia.org/wiki/File:LSTM.jpg</a> . . . . .	43
2.18. <i>Red neuronal recurrente GRU.</i> Imagen del usuario MingxianLin de Wikipedia (CC BY-SA 4.0), recuperado de <a href="https://commons.wikimedia.org/wiki/File:GRU.png">https://commons.wikimedia.org/wiki/File:GRU.png</a> . . . . .	44
2.19. <i>Arquitectura interna de las redes neuronales recurrentes simples, LSTM y GRU.</i> Adaptado de “RNN, LSTM & GRU”, por dprogrammer, 2019 ( <a href="http://dprogrammer.org/rnn-lstm-gru">http://dprogrammer.org/rnn-lstm-gru</a> ) . . . . .	46
2.20. <i>Arquitectura de la red neuronal convolucional conocida como ‘LeNet’</i> [19]. . . . .	47
2.21. <i>Arquitectura de la red neuronal convolucional presentada por Alex Krizhevsky et al., conocida como ‘AlexNet’</i> [25]. . . . .	48
2.22. <i>Capa convolucional en una RNC.</i> Adaptado de <i>Kernel sliding over the Image</i> por Towards Data Science, 2019, recuperada de <a href="https://towardsdatascience.com/understanding-1d-and-3d-convolution-neural-network-keras-9d8f76e29610">https://towardsdatascience.com/understanding-1d-and-3d-convolution-neural-network-keras-9d8f76e29610</a> . . . . .	50

2.23. <i>Ejemplo de RNC bidimensional</i> . Adaptado de <i>A CNN sequence to classify handwritten digits</i> por Towards Data Science, 2018, recuperada de <a href="https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53">https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53</a>	51
2.24. <i>Ejemplo de RNC unidimensional</i> [36]	52
2.25. <i>Modelo Secuencia a Secuencia</i> . Adaptado de <i>Encoder-decoder sequence to sequence model</i> por Towards Data Science, 2019, recuperada de: <a href="https://towardsdatascience.com/understanding-encoder-decoder-sequence-to-sequence-model-679e04af4346">https://towardsdatascience.com/understanding-encoder-decoder-sequence-to-sequence-model-679e04af4346</a>	54
2.26. <i>Mecanismo de Atención de Bahdanau</i> . Adaptado de <i>Overall process for Bahdanau Attention seq2seq model</i> por FloydHub Blog, 2019, recuperada de: <a href="https://blog.floydhub.com/attention-mechanism/#bahdanau-att-step2">https://blog.floydhub.com/attention-mechanism/#bahdanau-att-step2</a>	55
2.27. <i>Desarrollo del mecanismo de atención</i> . Adaptado de <i>Flow of calculating Attention weights in Bahdanau Attention</i> por FloydHub Blog, 2019, recuperada de: <a href="https://blog.floydhub.com/attention-mechanism/#bahdanau-att-step2">https://blog.floydhub.com/attention-mechanism/#bahdanau-att-step2</a>	56
3.1. <i>Valores reales (color rojo) del sismograma de la estación AAIG (Aguascalientes, Aguascalientes) contra valores predichos (color verde) por la red neuronal Vanilla-GRU para el terremoto del 19 de septiembre de 2017 en Puebla, México.</i>	67
3.2. <i>Valores reales (color rojo) del sismograma de la estación ZAIG (Zacatecas, Zacatecas) contra valores predichos (color verde) por la red neuronal Vanilla-GRU para el terremoto del 19 de septiembre de 2017 en Puebla, México.</i>	67
3.3. <i>Valores reales (color rojo) del sismograma de la estación THIG (Tapachula, Chiapas) contra valores predichos (color verde) por la red neuronal Vanilla-GRU para el terremoto del 19 de septiembre de 2017 en Puebla, México.</i>	68
3.4. <i>Sismograma de la estación ZAIG (Zacatecas, Zacatecas) componente Norte-Sur contra valores predichos (color verde) por la red neuronal Vanilla-GRU para el terremoto del 19 de septiembre de 2017 en Puebla, México.</i>	70
3.5. <i>Sismograma de la estación ZAIG (Zacatecas, Zacatecas) componente Vertical contra valores predichos (color verde) por la red neuronal Vanilla-GRU para el terremoto del 19 de septiembre de 2017 en Puebla, México.</i>	70
3.6. <i>Sismograma de la estación IGIG (Irapuato, Guanajuato) del terremoto del 8 septiembre de 2017 en Pijijiapan, Chiapas, contra valores predichos (color verde) por la red neuronal Vanilla-GRU.</i>	72

## Índice de figuras

---

3.7.	<i>Sismograma de la estación ZIIG (Zacatecas, Zacatecas) del terremoto del 8 septiembre de 2017 en Pijijiapan, Chiapas, contra valores predichos (color verde) por la red neuronal Vanilla-GRU. . . . .</i>	73
3.8.	<i>Sismograma de la estación ZIIG (Zacatecas, Zacatecas) del terremoto del 1 febrero de 2019 en Ciudad Hidalgo, Chiapas, contra valores predichos (color verde) por la red neuronal Vanilla-GRU. . . . .</i>	74
3.9.	<i>Sismograma de la estación PZIG (EL Pozo, Ciudad de México) del terremoto del 1 febrero de 2019 en Ciudad Hidalgo, Chiapas, contra valores predichos (color verde) por la red neuronal Vanilla-GRU. . . . .</i>	75
3.10.	<i>Sismograma de la estación PZIG (El Pozo, Ciudad de México) en la componente espacial vertical para el terremoto del 19 noviembre de 2019 en Ciudad Hidalgo, Chiapas, contra valores predichos (color verde) por la red neuronal Vanilla-GRU. . . . .</i>	76
3.11.	<i>Sismograma de la estación TRQA (Tornsquist, Argentina) para el terremoto del 3 de junio de 2020 al norte de Chile, contra valores predichos (color verde) por la red neuronal Vanilla-GRU. . . . .</i>	78
3.12.	<i>Sismograma de la estación KIEV (Kiev, Ucrania) para el terremoto del 3 de junio de 2020 al norte de Chile, contra valores predichos (color verde) por la red neuronal Vanilla-GRU. . . . .</i>	79
3.13.	<i>Sismograma de la estación GRFO (Grafenberg, Alemania) para el terremoto del 2 de mayo de 2020 en Creta, Grecia, contra valores predichos (color verde) por la red neuronal Vanilla-GRU. . . . .</i>	80
3.14.	<i>Sismograma de la estación WVT (Tennessee, USA) para el terremoto del 2 de mayo de 2020 en Creta, Grecia, contra valores predichos (color verde) por la red neuronal Vanilla-GRU. . . . .</i>	81
3.15.	<i>Sismograma de la estación PFO (California, USA) para el terremoto del 20 de mayo de 1990 en Sudan, África, contra valores predichos (color verde) por la red neuronal Vanilla-GRU. . . . .</i>	82
3.16.	<i>Sismograma de la estación KIV (Kislovodsk, Rusia) para el terremoto del 20 de mayo de 1990 en Sudan, África, contra valores predichos (color verde) por la red neuronal Vanilla-GRU. . . . .</i>	83
3.17.	<i>Sismograma de la estación MBWA (Marble Bar, Australia) para el terremoto del 11 de marzo de 2011 en Honshu, Japón, contra valores predichos (color verde) por la red neuronal Vanilla-GRU. . . . .</i>	84
3.18.	<i>Sismograma de la estación TEIG (Nayarit, México) para el terremoto del 11 de marzo de 2011 en Honshu, Japón, contra valores predichos (color verde) por la red neuronal Vanilla-GRU. . . . .</i>	85
3.19.	<i>Sismograma de la estación TAU (Tasmania, Australia) para el terremoto del 21 de agosto de 2003 en Isla Sur, Nueva Zelanda, contra valores predichos (color verde) por la red neuronal Vanilla-GRU. . . . .</i>	86

3.20. *Sismograma de la estación SBA (Scott Base, Antártica) para el terremoto del 21 de agosto de 2003 en Isla Sur, Nueva Zelanda, contra valores predichos (color verde) por la red neuronal Vanilla-GRU.* 87

A.1. *Menú que se obtiene cuando la conexión con la base de datos del SSN ha sido exitosa.* . . . . . 104

A.2. *Ayuda que se obtiene cuando se introduce el comando **HELP**.* . . . . 105

B.1. *Pantalla de inicio de la página de IRIS para descargar sismogramas de eventos sísmicos.* . . . . . 107

B.2. *Búsqueda de sismos con magnitud mayor o igual 8 en un intervalo de tiempo que va desde enero 1 de 1990 hasta febrero 5 de 2021.* . . 108

B.3. *Pantalla que aparece al seleccionar un sismo.* . . . . . 109

B.4. *Pantalla que aparece al seleccionar ‘Request Data’.* . . . . . 110

B.5. *Pantalla que aparece después de dar clic en Request Data.* . . . . . 111

B.6. *Pantalla que aparece al estar lista la descarga.* . . . . . 111

## Índice de figuras

---

---

## Índice de cuadros

3.1. Comparación del desempeño de los modelos LSTM y GRU para el evento sísmico del 19 de septiembre de 2017 en Puebla, México. Se presentan los resultados de sismogramas de siete estaciones sísmicas ubicadas en la república mexicana, todas ellos en la componente espacial Este-Oeste y sus respectivas métricas. En letra negrita se resaltan los mejores resultados. . . . .	66
3.2. Desempeño de modelos LSTM y GRU para sismogramas en las componentes espaciales Norte-Sur y Vertical para el terremoto del 19 de septiembre en Puebla, México. Se presentan los resultados de sismogramas de cuatro estaciones sísmicas. . . . .	69
3.3. Desempeño de modelos LSTM y GRU para sismogramas del terremoto ocurrido el 8 de septiembre de 2017 en Pijijiapan, Chiapas. Se presentan los resultados de sismogramas de dos estaciones sísmicas. . . . .	72
3.4. Desempeño de modelos LSTM y GRU para sismogramas del terremoto ocurrido el 1 de febrero de 2019 en Ciudad Hidalgo, Chiapas. Se presentan los resultados de sismogramas de dos estaciones sísmicas. . . . .	74
3.5. Desempeño de modelos LSTM y GRU para sismogramas del terremoto ocurrido el 19 de noviembre de 2019 en Ciudad Hidalgo, Chiapas. Se presentan los resultados de tres sismogramas. . . . .	76
3.6. Desempeño de modelos LSTM y GRU para sismogramas en la componente Este-Oeste del terremoto ocurrido el 3 de junio de 2020 al norte de Chile. Se presentan los resultados de sismogramas de tres estaciones sísmicas. . . . .	78
3.7. Desempeño de modelos LSTM y GRU para sismogramas en la componente Este-Oeste del terremoto ocurrido el 2 de mayo de 2020 en Creta, Grecia. Se presentan los resultados de sismogramas de tres estaciones sísmicas. . . . .	80
3.8. Desempeño de modelos LSTM y GRU para sismogramas en la componente Este-Oeste del terremoto ocurrido el 20 de mayo de 1990 en Sudan, África. Se presentan los resultados de sismogramas de tres estaciones sísmicas. . . . .	82

## Índice de cuadros

---

3.9. <i>Desempeño de modelos LSTM y GRU para sismogramas en la componente Este-Oeste del terremoto ocurrido el 11 de marzo de 2011 en Honshu, Japón. Se presentan los resultados de sismogramas de tres estaciones sísmicas. . . . .</i>	84
3.10. <i>Desempeño de modelos LSTM y GRU para sismogramas en la componente Este-Oeste del terremoto ocurrido el 21 de agosto de 2003 en Isla Sur, Nueva Zelanda. Se presentan los resultados de sismogramas de tres estaciones sísmicas. . . . .</i>	86
3.11. <i>Comparación del desempeño de las diferentes configuraciones de redes neuronales recurrentes para el evento sísmico del 19 de septiembre de 2017 en Puebla, México. Se presentan los resultados de sismogramas de siete estaciones sísmicas ubicadas en la república mexicana, todas ellos en la componente espacial Este-Oeste y sus respectivas métricas. En letra negrita se resaltan los mejores resultados. . . . .</i>	90

# Capítulo 1

---

## Introducción

El estudio de la sismología es y será de suma importancia en el mundo entero, pero se vuelve más crítico en partes del mundo consideradas de alta actividad sísmica. Tal es el caso de México, puesto que el territorio mexicano se encuentra asentado en una zona activa de altísima sismicidad, la cual forma parte de una extensión territorial de más de 40 000 km de largo que rodea el océano pacífico conocida a nivel mundial como el *Anillo de Fuego del Pacífico*, donde se encuentran la mayoría de volcanes activos, aproximadamente el 75 % y se genera gran parte de la actividad sísmica mundial, cerca del 90 %. A lo largo de esta zona se encuentran las fronteras de varias placas tectónicas interactuando entre ellas, incluidas la placa del Pacífico, Juan de Fuca, Cocos, India-Australiana, Nazca, Norteamérica y Filipina.

En particular, parte del territorio mexicano se encuentra asentado en la unión de cinco placas tectónicas, a saber: placa de Norteamérica, placa del Pacífico, placa de Rivera, placa de Cocos y placa del Caribe. Concretamente, se tiene que la placa de Cocos y la placa de Rivera se encuentran en subducción con la placa de Norteamérica. La placa de Cocos se sumerge bajo los estados de Oaxaca, Chiapas, Guerrero y Michoacán, mientras que la placa de Rivera lo hace bajo los estados de Jalisco y Colima; a toda esta extensión de territorio costero se le conoce como la zona de subducción mexicana.

Otra fuente de sismos en el territorio mexicano (Baja California Norte, Baja California Sur y Sonora) es la Falla de San Andrés, la cual es consecuencia del movimiento encontrado entre la placa de Norteamérica que se mueve hacia el suroeste y la placa del Pacífico que se mueve hacia el noreste. Se le da el nombre particular de *falla* porque es una zona donde se puede observar una aparente ruptura en la corteza terrestre, aunque lo que realmente sucede es que dos placas tectónicas están moviéndose relativamente una con respecto de la otra haciendo contacto en sus fronteras provocando a su vez movimientos telúricos.

Por otra parte, la zona que comprende la Ciudad de México y su periferia en su origen fue una zona lacustre que se componía de cinco lagos: Texcoco, Xochimilco,

## Capítulo 1. Introducción

---

Chalco, Xaltocan y Zumpango. Por tanto, buena parte de los suelos de la Ciudad de México se catalogan como suelos blandos y suelos de transición; suelos que tienen la característica de amplificar y prolongar peligrosamente las ondas sísmicas generadas durante un evento sísmico. En el país se tiene registro de varios volcanes activos que también contribuyen a la actividad sísmica, entre ellos: Popocatépetl, Pico de Orizaba, Bárcena, Ceboruco, Nevado de Toluca, Paricutín, etc.

La ecuación diferencial parcial que describe la propagación de las ondas sísmicas a través de la tierra es una ecuación realmente complicada y hasta el momento imposible de resolver directamente, no por su complejidad en sí misma, sino sobre todo por las variables cambiantes en el tiempo y en el espacio de la cual es dependiente y las complejas interacciones con sus medios de transmisión, donde dichas ondas sísmicas pueden ser: atenuadas, dispersadas, refractadas, reflejadas, extinguidas, amplificadas, etc.

La ecuación que describe el movimiento de las ondas sísmicas es una ecuación que se obtiene al aplicar las relaciones básicas entre las fuerzas y deformaciones en el material por el cual se propagan las ondas sísmicas. Para poder resolver esta ecuación diferencial parcial de una forma más sencilla se hacen suposiciones tales como que el material por el cual se propagan las ondas sísmicas es infinito, homogéneo e isotrópico; suposiciones que claramente están lejos de ser realistas. Lo que realmente sucede es que las propiedades físicas inherentes de los materiales por los cuales se propagan las ondas sísmicas cambian constantemente, tanto en densidad, como en coeficiente de elasticidad. Con las suposiciones anteriores se puede calcular la velocidad aproximada de la onda primaria y la onda secundaria, sin embargo, para simular modelos realistas de la propagación de ondas sísmicas se toma otra aproximación a la solución de la ecuación de ondas sísmicas. Un método más cercano a condiciones realistas para resolver la ecuación diferencial parcial es por medio de métodos numéricos apoyados en una buena capacidad de cómputo para potenciar la posibilidad de llegar a resolver la ecuación y el tiempo para alcanzar dicha solución. Ésto con el objetivo de realizar simulaciones lo más realistas posible de la interacción de las ondas sísmicas y su medio de propagación, estas simulaciones y el análisis de sismogramas arrojan información que ayuda a modelar la estructura interna y las propiedades físicas del interior de la tierra puesto que hacer mediciones a grandes profundidades es inviable y para determinadas profundidades imposible con la tecnología actual. Todo el conocimiento que se tiene de la estructura interna de la tierra es gracias a este tipo de análisis.

Las lecturas recopiladas en los sismogramas reflejan totalmente las interacciones que tienen las ondas sísmicas con los medios por los cuales se han propagado hasta

llegar al punto donde han sido registradas. Por tanto, al aplicar un modelo de aprendizaje supervisado enfocado a la predicción de las lecturas de los sismogramas heredará completamente la complejidad inherente en la interacción entre ondas sísmicas y medios de propagación.

Lo que se hace habitualmente en sismología es hacer un análisis en reversa; se le conoce así a esta metodología porque primero se analizan los resultados de algún fenómeno medible, en el caso que se aborda en este proyecto de tesis son lecturas obtenidas por sismógrafos, y posteriormente se intenta llegar a un modelo que describa lo mejor posible el desarrollo del fenómeno en cuestión. Se hace de esta forma porque no se tiene el conocimiento completo de la física que describe las interacciones de las placas tectónicas, de tener el conocimiento total de como interactúan las placas tectónicas se estaría en posición de poder predecir el lugar, fecha, hora y magnitud de los eventos sísmicos, lo cual, como bien es sabido, actualmente es imposible.

Se ha sugerido que previo a un evento sísmico de gran magnitud se presentan eventos precursoros que delatan la ocurrencia de un sismo, tal es el caso de una serie de pequeños terremotos antes del terremoto principal, o una disminución de resistividad eléctrica en el suelo donde se generará la falla sísmica, o deformación del suelo, o cambios en intensidad del campo electromagnético. Otro precursor que se suele reportar con frecuencia es el comportamiento inusual de animales. En Japón se ha llegado a invertir billones de dólares en dichas investigaciones sin tener resultados contundentes, Japón es un país con alta actividad sísmica, al igual que México está en la zona conocida como el Anillo de Fuego del Pacífico. China es otro país que ha invertido en este tipo de investigaciones tratando de predecir la ocurrencia de eventos sísmicos.

Otra ruta diferente en querer adelantarse a la ocurrencia de un evento sísmico se ha basado en los registros de eventos sismológicos a lo largo del tiempo y con base a éstos estimar cuándo se presentará el siguiente terremoto, lo cual hasta el momento también ha probado ser no confiable.

## 1.1. Presentación

Se sabe que para aplicar un modelo de aprendizaje profundo y elevar las posibilidades de éxito se necesita de una base de datos extensa, de tal forma que el modelo que se plantee tenga la posibilidad de aprender por medio de exposición repetitiva y abundante a dichos datos. Otro requisito es presentar los datos de entrenamiento de tal forma que el modelo tenga mayor probabilidad de extraer rasgos significativos y caracterizadores de los datos a los cuales se exponga, en

## Capítulo 1. Introducción

---

otras palabras, se necesita presentar cantidad y calidad de datos al modelo de aprendizaje profundo. Por último, y no menos importante, se necesita encontrar funciones de pérdida, funciones de optimización e hiperparámetros adecuados que permitan que el modelo entregue los resultados esperados.

La inclusión de modelos de aprendizaje profundo en diversas áreas del conocimiento humano ha tenido un progreso continuo en los últimos años y es que han demostrado ser una gran herramienta de apoyo para muchas ciencias. La sismología no ha sido la excepción. El gran poder que entregan los modelos de aprendizaje profundo es la capacidad de aprender rasgos o características que escapan al escrutinio humano, además de automatizar estos procesos de extracción de información valiosa.

Las lecturas presentes en los sismogramas contienen mucha información de las ondas sísmicas, por ejemplo, acerca de las propiedades físicas de los medios por los cuales se han propagado, del tipo de fenómeno (natural o provocado por el hombre) qué dio origen a dichas ondas sísmicas, de la distancia al hipocentro y epicentro, el tipo de falla que ocurrió, etc.

En este trabajo de tesis se planteó el objetivo de analizar los sismogramas por medio de redes neuronales recurrentes con ayuda de redes neuronales convolucionales unidimensionales y mecanismos de atención, tratando las lecturas contenidas en los sismogramas como series de tiempo y extrayendo de las mismas pequeñas sucesiones para exponerlas a redes neuronales a través de un entrenamiento supervisado con un enfoque dirigido a predicción.

Se espera que las diferentes configuraciones de redes neuronales que se planteen en este trabajo de tesis sean capaces de aprender una función que establezca una relación entre una sucesión de valores consecutivos de entrada y valores de salida posteriores a esta sucesión de datos de entrada, por supuesto, se desea que los valores que entregue la función sean muy cercanos a los valores reales. En otras palabras, lo que se hará será exponer a la red neuronal durante el entrenamiento a pequeñas sucesiones de valores consecutivos extraídas de los sismogramas esperando que la red neuronal sea capaz de reconocer o aprender la relación existente entre estos valores sucesivos de entrada y los valores que se espera a la salida de la red neuronal, relación que se sabe existe puesto que el comportamiento de estas ondas sísmicas está gobernado por un sistema de ecuaciones diferenciales parciales.

## 1.2. Motivación

La concepción de este proyecto de tesis surgió del interés y la curiosidad de entender, por una parte, las interacciones y lo complejo de las ondas sísmicas, y por otra parte, los fenómenos que las originan, sobre todo, aquellas que surgen debido a las interacciones de las placas tectónicas. Fundamentalmente, el deseo fue poder aplicar métodos de aprendizaje profundo a algún fenómeno tan interesante como lo es la propagación de las ondas sísmicas a través de la tierra. Es la combinación de hacer converger en un proyecto de tesis dos ciencias tan interesantes por sí mismas como lo son las diversas técnicas de aprendizaje profundo y la sismología lo que motiva y vuelve aún más interesante el proyecto. Específicamente, en esta tesis, los temas que se hacen converger son las redes neuronales y el fenómeno de propagación de ondas sísmicas.

La elección de darle a este trabajo de tesis el enfoque de predicción mediante aprendizaje supervisado usando redes neuronales surgió como una idea alternativa para tratar de tener otra aproximación al problema tan complejo de describir la propagación de las ondas sísmicas en la tierra. Se sabe que las redes neuronales recurrentes han probado tener un gran desempeño en series de tiempo o sucesiones de datos, las cuales inherentemente presentan un orden o relación existente en cada uno de sus valores, y puesto se sabe que hay una física que rige la propagación de las ondas sísmicas, la idea de aplicar redes neuronales a las lecturas contenidas en los sismogramas surge con naturalidad.

Afortunadamente, en el mundo hay varias redes de estaciones sísmicas que hacen pública sus bases de datos, y que por lo tanto, facilitan cumplir con el requerimiento de tener una base de datos extensa y variada. Cabe mencionar que esta gran proliferación de estaciones sísmicas tuvo como principal motivación monitorear los ensayos nucleares de la Unión Soviética a inicios de la década de 1960 en la época de la guerra fría. Actualmente, a las redes de estaciones sísmicas se les da un gran uso en exploración de suelos para hallar la ubicación de materias primas valiosas como petróleo y gas. En el ámbito científico se usan para recrear representaciones de la composición interna de la tierra, también para comprender la respuesta de ciertos materiales de construcción a diferentes tipos de ondas sísmicas y con ello lograr construcciones más resistentes al efecto de las ondas sísmicas; y aún más sorprendente es saber que se han enviado sismómetros a la Luna y a Marte para exploración de la estructura interna de dichos cuerpos celestes y, obviamente, monitoreo de la actividad sísmica.

### 1.3. Objetivo General

Demostrar que existen configuraciones de redes neuronales capaces de aprender el comportamiento subyacente existente en el fenómeno de propagación de ondas sísmicas y, como consecuencia, hacer predicciones de valores no vistos por las redes neuronales con un buen grado de exactitud con base en valores previos a los predichos.

#### 1.3.1. Objetivos particulares

- Entrenar redes neuronales recurrentes apoyadas en redes neuronales convolucionales unidimensionales y mecanismos de atención que sean capaces de realizar predicciones de lecturas de ondas sísmicas contenidas en sismogramas dentro de un rango de error aceptable. Las métricas con las cuales se decidirá si las predicciones hechas por las redes neuronales están dentro de un rango aceptable se plantearán en el capítulo tres de esta tesis.
- Realizar pruebas con los modelos de redes recurrentes del tipo LSTM (*Long Short-Term Memory*) y GRU (*Gated Recurrent Unit*); estos son los modelos más ampliamente usados por ser más poderosos en capacidad de retención de información de corto, pero sobre todo de largo plazo y, además, ser más resistentes al desvanecimiento por gradiente descendente <sup>1</sup> que una red neuronal recurrente simple.

### 1.4. Hipótesis

La hipótesis que se plantea es que las redes neuronales recurrentes son capaces de establecer una relación que modele el comportamiento que exhiben los valores consecutivos en las sucesiones de lecturas contenidos en los sismogramas, de tal forma que el resultado de entregar una sucesión de lecturas consecutivas extraídas de un sismograma a la red neuronal recurrente sea una lectura o una sucesión de lecturas inmediatamente posteriores a la sucesión de entrada.

---

<sup>1</sup>Este tema se tratará en el próximo capítulo.

## 1.5. Planteamiento del problema

El problema se dividirá en varias secciones, de tal forma que se vayan cumpliendo objetivos, y a su vez, obteniendo resultados sobre los cuales ir construyendo uno sobre otro y que guíen paulatinamente a los resultados esperados.

Los pasos principales que se llevarán a cabo son los siguientes:

1. Acceder a bases de datos de sismogramas y extraer varios de ellos relativos a algún evento sísmico significativo en magnitud.
2. De los sismogramas descargados extraer sus series de tiempo e información relevante (frecuencia de muestreo del sensor, factor de calibración, cantidad total de lecturas contenidas en el sismograma, etc.) de los mismos.
3. Preparar o acondicionar las series de tiempo obtenidas de los sismogramas de tal forma que la información presentada a las redes neuronales sea más fácil de procesar.
4. Para cada serie de tiempo obtener subconjuntos de entrenamiento, validación y prueba. El método que se empleará para crear subconjuntos que contengan los vectores a la entrada de la red neuronal y los vectores a la salida de la misma de cada subconjunto será el de: *Sliding Window*.
5. Con los subconjuntos de entrenamiento, validación y prueba, llevar a cabo pruebas y evaluar el desempeño de los dos modelos de redes neuronales recurrentes que se pondrán a prueba. Probar el desempeño de las siguientes configuraciones de redes neuronales recurrentes: *Vanilla-RNN*, *Stacked-RNN*, *BiRNN*, *Conv1D-RNN*, *Conv1D-BiRNN*, *BiRNN-Attention*.
6. Refinar tanto como sea posible los hiperparámetros<sup>2</sup> de las redes neuronales mediante métodos de optimización (*Manual Search*, *Grid Search*, *Random Search*, *Bayesian Optimization* o *Algoritmos Genéticos*).
7. Evaluar el desempeño de las redes neuronales con sismogramas pertenecientes al mismo evento sísmico pero de estaciones sísmicas y componentes espaciales diferentes que con el cual fue entrenada la red neuronal.
8. Buscar en las bases de datos de sismogramas eventos sísmicos que difieran en fecha y ubicación al evento sísmico con el cual fueron entrenadas las redes neuronales y evaluar el desempeño de las mismas.

---

<sup>2</sup>Hiperparámetro es un parámetro que se usa para controlar un proceso de aprendizaje en un algoritmo de Aprendizaje Máquina. Son parámetros que no son aprendidos en el proceso de entrenamiento, son asignados al inicio del mismo.

### 1.6. Metodología

La metodología que se seguirá es la siguiente:

- Primeramente, documentarse a fondo en sismología, principalmente, en teoría y aplicaciones de propagación de ondas sísmicas. Paralelamente a esto, hacer un estudio del estado del arte de técnicas de aprendizaje profundo, en concreto, redes neuronales recurrentes, redes neuronales convolucionales unidimensionales y mecanismos de atención, y sus diferentes configuraciones o arquitecturas dirigidas a regresión o predicción.
- Generar los algoritmos para el procesamiento y tratamiento de las lecturas contenidas en los sismogramas.
- Realizar pruebas con las series de tiempo obtenidas de las lecturas contenidas en los sismogramas con los dos modelos de redes neuronales recurrentes más usados y conocidos (LSTM y GRU) y con las diferentes configuraciones de redes neuronales recurrentes con apoyo de redes convolucionales unidimensionales y mecanismos de atención. Lo anterior con el fin de potenciar los resultados en cuanto a las predicciones realizadas por las redes neuronales.
- Optimizar hiperparámetros de todas las configuraciones propuestas por medio de las técnicas de optimización mencionadas anteriormente.
- Analizar los resultados obtenidos en los puntos anteriores y determinar el modelo de red neuronal recurrente (LSTM o GRU), hiperparámetros y configuración de red neuronal que presente mejor desempeño.
- Llegar a una conclusión del significado de los resultados obtenidos.

### 1.7. Contribuciones

Al dar el enfoque de predicción mediante aprendizaje supervisado haciendo uso de sucesiones consecutivas de lecturas contenidas en los sismogramas se está tomando una ruta alterna a la solución de la ecuación de propagación de ondas sísmicas; obtenemos predicciones de lecturas contenidas en sismogramas con base a lecturas anteriores sin hacer ningún tipo de suposiciones previas acerca de los medios de transmisión o de la interacción de las ondas sísmicas con los medios de transmisión o entre ellas mismas ya que los valores son tomados directamente de los sensores que detectan el movimiento del terreno. Lo que se está realizando es fundamentar la predicción de las siguientes lecturas contenidas en sismogramas con base a lecturas pasadas y no en las variables independientes que habitualmente

se tendrían (coeficiente de elasticidad, densidad y fuerza; todas ellas variables dependientes de la posición y el tiempo) al resolver la ecuación diferencial.

Se espera obtener resultados favorables en cuanto a la generalidad y desempeño de la relación aprendida por las redes neuronales, es decir, que las redes neuronales sean entrenadas en una sola ocasión, con un único sismograma de una sola estación sísmica y en una sola componente espacial para un evento sísmico en particular y que con ese único entrenamiento las redes neuronales entreguen resultados favorables en sus predicciones con sismogramas de cualquier otro evento sísmico, independientemente de fecha, lugar de ocurrencia y magnitud del mismo.

## 1.8. Estructura de la tesis

Este trabajo de tesis se organiza en cinco capítulos de la siguiente manera:

- *Capítulo 1.* Se da un panorama general del tema de investigación y los detalles a considerar del mismo. En el capítulo se incluyen las siguientes secciones: motivación, objetivo, hipótesis, planteamiento del problema y metodología. Secciones que clarifican aún más el desarrollo y alcance de la tesis.
- *Capítulo 2.* Se abordan tres principales temas sobre los que se apoya este trabajo de tesis, los cuales son: sismología, series de tiempo y redes neuronales recurrentes, convolucionales y mecanismos de atención.
  - En sismología se da una descripción de los fenómenos que dan origen a las ondas sísmicas, se abordan detalladamente temas como la propagación de ondas sísmicas, sus interacciones con los medios de transmisión y el impacto que tiene el estudio de esta ciencia en la sociedad.
  - En series de tiempo se da un panorama general de conceptos y términos relevantes para el desarrollo de este trabajo de tesis.
  - En lo referente a redes neuronales recurrentes, redes neuronales convolucionales unidimensionales y mecanismos de atención se desarrollarán conceptos y definiciones sobre los cuales se fundamentan estos algoritmos de aprendizaje profundo. Se hará una revisión del estado del arte de redes neuronales recurrentes y convolucionales unidimensionales aplicadas a predicción. Además, se agregarán mecanismos de atención a algunas de las diferentes configuraciones que se emplearán en este trabajo de tesis, esperando mejorar los resultados obtenidos en las métricas propuestas.
- *Capítulo 3.* Se desarrollan y comentan los algoritmos y métodos que se involucrarán en el manejo de las series de tiempo contenidas en los sismogramas y

## Capítulo 1. Introducción

---

los diferentes modelos de redes neuronales detalladamente. El objetivo en este capítulo es hallar configuraciones que involucren redes neuronales recurrentes, redes neuronales convolucionales unidimensionales, mecanismos de atención, hiperparámetros y formas de preparación o acondicionamiento de datos que nos permitan tener los mejores resultados posibles en predicción de lecturas contenidas en sismogramas.

Se entrenarán diferentes configuraciones de redes neuronales, todas ellas con el mismo sismograma (sismograma extraído de la estación sísmica IGIG en la componente espacial Este-Oeste para el terremoto ocurrido el 19 de septiembre de 2017 en Puebla, México) y en una sola ocasión, es decir, el entrenamiento no se actualizará con cada sismograma que se analice, todas las predicciones que se realicen de los diferentes sismogramas que se presenten en el capítulo cuatro se deberán a este único entrenamiento. Se analizarán sismogramas de eventos sísmicos alrededor del mundo, sismogramas pertenecientes a eventos sísmicos diferentes que con el cual fueron entrenadas las redes neuronales. Se desea observar que la relación aprendida por las redes neuronales sea generalizable, y si lo es, evaluar el desempeño de la misma.

- *Capítulo 4.* Se analizan los resultados obtenidos en el capítulo anterior. Se elige el modelo de red neuronal recurrente y configuración que mejor resultados arroje en general.
- *Capítulo 5.* Se elabora una conclusión respecto a los resultados obtenidos en el capítulo anterior y se realizan propuestas de trabajo a futuro para ampliar y mejorar este trabajo de tesis.

## Capítulo 2

---

# Marco teórico

### 2.1. Introducción

En este capítulo se presentan definiciones, conceptos y teoría sobre la cual se respalda y desarrolla este trabajo de tesis. Se introducirán elementos importantes de sismología, series de tiempo y aprendizaje profundo. Especialmente, se abordarán redes neuronales recurrentes, redes neuronales convolucionales unidimensionales y mecanismos de atención.

### 2.2. Sismología

#### 2.2.1. Ecuación de ondas sísmicas

La ecuación de propagación de ondas sísmicas es desarrollada usando la teoría de esfuerzos y deformaciones en medios de transmisión elásticos. Existen dos tipos de soluciones, correspondientes a las ondas de compresión o ondas  $P$  (Primarias) y a las ondas de corte o ondas  $S$  (Secundarias). En ambas soluciones de dichas ecuaciones se busca una expresión que describa la velocidad de propagación de las ondas sísmicas  $P$  y  $S$ . La velocidad instantánea de las ondas sísmicas está determinada por las propiedades físicas de los medios por los cuales se propagan, específicamente, densidad y propiedades elásticas del medio [28].

La ecuación es la siguiente:

$$\rho \frac{\partial^2 u_i}{\partial t^2} = \partial_j \tau_{ij} + f_i \quad (2.1)$$

Esta es la ecuación fundamental de la sismología, y es conocida como la *ecuación de momentum* o *ecuación de movimiento*. Cada uno de los términos  $u_i$ ,  $\tau_{ij}$  y  $f_i$  en la ecuación 2.1 son funciones dependientes de la posición y el tiempo.

## Capítulo 2. Marco teórico

---

Donde  $\tau_{ij}$  representa las tres componentes espaciales de las deformaciones elásticas en los tres diferentes planos en el espacio, por otra parte,  $f_i$  representa las componentes espaciales de las fuerzas de cuerpo,  $t$  el tiempo,  $u_i$  las componentes espaciales de la función de desplazamiento y  $\rho$  la densidad del medio de propagación.

El término *fuerzas de cuerpo* es un vector que se suele descomponer en una componente gravitacional  $f_g$  y una componente de origen  $f_s$ , las cuales suelen ser ignoradas con el objetivo de simplificar la solución de la ecuación 2.1. Al eliminar las componentes de fuerzas de cuerpo se tiene la siguiente ecuación:

$$\rho \frac{\partial^2 u_i}{\partial t^2} = \partial \tau_{ij} \quad (2.2)$$

La ecuación 2.2 se conoce en sismología como la *ecuación homogénea de movimiento*, esta ecuación gobierna la propagación de las ondas sísmicas. La ecuación aún se tiene que desarrollar y realizar un par de suposiciones más para llegar a expresiones que permitan calcular las velocidades de las ondas sísmicas P y S con mayor facilidad.

Para poder resolver la ecuación 2.2 se necesita una relación entre las fuerzas y las deformaciones de tal forma que sea posible expresar  $\tau_{ij}$  en términos del desplazamiento  $u_i$ . La relación que se necesita es la siguiente:

$$\tau_{ij} = \lambda \delta_{ij} \partial_k u_k + \mu (\partial_i u_j + \partial_j u_i) \quad (2.3)$$

Al sustituir la relación 2.3 en la ecuación 2.2 y desarrollarla, se obtiene la siguiente ecuación:

$$\begin{aligned} \rho \frac{\partial^2 u_i}{\partial t^2} &= \partial_j [\lambda \delta_{ij} \partial_k u_k + \mu (\partial_i u_j + \partial_j u_i)] \\ &= \partial_i \lambda \partial_k u_k + \lambda \partial_i \partial_k u_k + \partial_j \mu (\partial_i u_j + \partial_j u_i) + \mu \partial_j \partial_i u_j + \mu \partial_j \partial_j u_i \\ &= \partial_i \lambda \partial_k u_k + \partial_j \mu (\partial_i u_j + \partial_j u_i) + \lambda \partial_i \partial_k u_k + \mu \partial_i \partial_j u_j + \mu \partial_j \partial_j u_i \end{aligned} \quad (2.4)$$

La ecuación 2.4 se puede reescribir con notación vectorial para mayor simplicidad de la siguiente forma:

$$\rho \ddot{\mathbf{u}} = \nabla \lambda (\nabla \cdot \mathbf{u}) + \nabla \mu \cdot [\nabla \mathbf{u} + (\nabla \mathbf{u})^T] + (\lambda + \mu) \nabla \nabla \cdot \mathbf{u} + \mu \nabla^2 \mathbf{u} \quad (2.5)$$

Donde se tiene que:

$$\ddot{\mathbf{u}} \equiv \frac{\partial^2 \mathbf{u}}{\partial t^2} \quad (2.6)$$

Usando la siguiente identidad vectorial y desarrollándola se obtiene lo siguiente:

$$\begin{aligned}\nabla \times \nabla \times \mathbf{u} &= \nabla \nabla \cdot \mathbf{u} - \nabla^2 \mathbf{u} \\ \nabla^2 \mathbf{u} &= \nabla \nabla \cdot \mathbf{u} - \nabla \times \nabla \times \mathbf{u}\end{aligned}\tag{2.7}$$

Sustituyendo la identidad vectorial 2.7 en la ecuación 2.5, se obtiene la siguiente expresión:

$$\rho \ddot{\mathbf{u}} = \nabla \lambda (\nabla \cdot \mathbf{u}) + \nabla \mu \cdot [\nabla \mathbf{u} + (\nabla \mathbf{u})^T] + (\lambda + 2\mu) \nabla \nabla \cdot \mathbf{u} - \mu \nabla \times \nabla \times \mathbf{u}\tag{2.8}$$

La ecuación 2.8 es una forma de expresar la *ecuación de onda sísmica*. Los dos primeros términos de esta ecuación en el lado derecho de la igualdad son no nulos cuando el material por el cual se propagan las ondas sísmicas es no homogéneo, lo cual sucede en la gran mayoría de los casos. El efecto de estos dos términos hacen que la solución de la ecuación 2.8 se complique demasiado, por tanto, y por simplicidad, suelen ser ignorados suponiendo que los medios por los cuales se propagan las ondas sísmicas son homogéneos. La ecuación resultante es la siguiente:

$$\rho \ddot{\mathbf{u}} = (\lambda + 2\mu) \nabla \nabla \cdot \mathbf{u} - \mu \nabla \times \nabla \times \mathbf{u}\tag{2.9}$$

La ecuación 2.9 es la forma estándar de la *ecuación de ondas sísmicas para medios homogéneos*. Para llegar a esta forma de la ecuación sísmica se ha ignorado la influencia de la fuerza gravitacional, de la fuerza de origen, y los gradientes de velocidad relativos a la propagación en medios no homogéneos, en otras palabras, se ha hecho la suposición de que las ondas se propagan por un material homogéneo e isotrópico, justo lo que se había mencionado en párrafos anteriores que se tendría que hacer para llegar a soluciones accesibles de la ecuación de ondas sísmicas [28].

Al tomar la divergencia de la ecuación 2.9 se obtiene la solución para la velocidad instantánea de las ondas sísmicas P y al tomar el rotacional de la misma ecuación, se obtiene la velocidad instantánea de las ondas sísmicas S.

### 2.2.2. Ondas sísmicas de cuerpo y superficiales

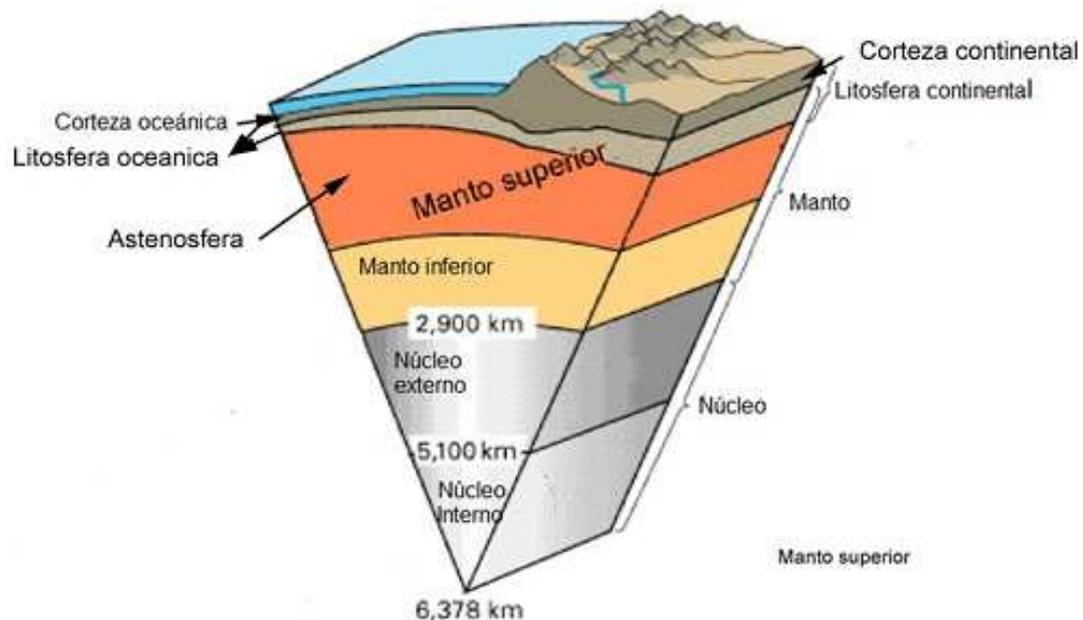
Durante la ocurrencia de un evento sísmico dos principales grupos de ondas sísmicas son generadas: *ondas de cuerpo* y *ondas superficiales*. Las ondas sísmicas primarias forman parte del grupo de ondas sísmicas conocidas como *ondas de cuerpo*, entre las cuales también se encuentran las ondas sísmicas secundarias, y se refieren a ellas como ondas de cuerpo porque viajan a través del interior de la Tierra en sus diferentes capas internas: litosfera, astenosfera, manto y núcleo (ver figura 2.1 en la página 14). Las ondas de cuerpo son más veloces que el grupo de las *ondas superficiales* (ondas sísmicas Love y Rayleigh) las cuales se propagan tal como su nombre lo dice, en la interfaz entre la corteza terrestre y la atmósfera.

## Capítulo 2. Marco teórico

---

Las ondas superficiales surgen de la interacción de las ondas elásticas con la superficie de la tierra, y están compuestas de una combinación lineal de ondas P y ondas S. Otra característica adicional es que el grupo de las ondas de cuerpo tienen una mayor frecuencia que las ondas superficiales. Las ondas superficiales decrecen en magnitud dependiendo de la profundidad del hipocentro o foco del evento sísmico.

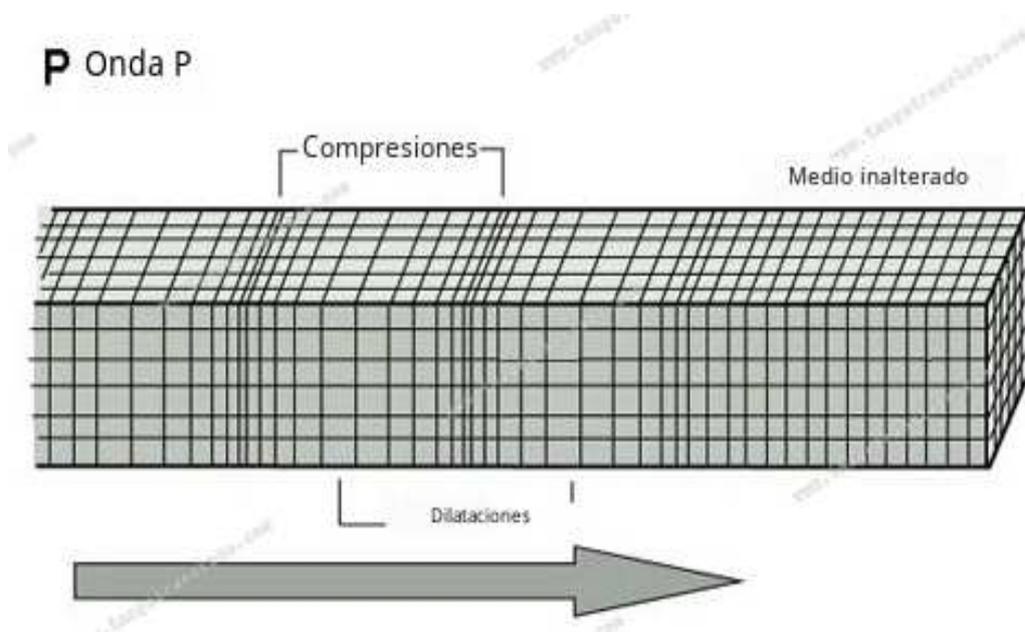
Al viajar a través de las diferentes capas de la Tierra las ondas de cuerpo son las primeras ondas detectadas por las estaciones sísmicas, en particular, entre las ondas de cuerpo, las ondas primarias se desplazan a mayor velocidad que las ondas secundarias, esta diferencia de velocidades de desplazamiento, y por tanto, en tiempo de llegada a una determinada estación sísmica permite a los sismólogos calcular la distancia existente entre la estación sísmica y foco del terremoto. Por otro lado, las ondas secundarias tienen una mayor amplitud que las ondas primarias y esto tiene como consecuencia que las ondas secundarias sean más perceptibles que las ondas primarias. Estas características hacen que sean fácilmente distinguibles ambos tipos de ondas sísmicas en el gráfico de un sismograma. Las ondas superficiales son las responsables de la mayor parte de los daños producidos durante un terremoto.



**Figura 2.1:** *Capas internas de la Tierra.* Imagen del usuario Jeremy Kemp de Wikipedia (CC BY-SA 3.0), recuperada de <https://commons.wikimedia.org/wiki/File:Earth-crust-cutaway-spanish.svg>

### 2.2.3. Ondas P

Se le nombró onda sísmica P como una forma de abreviación a dos nombres con los cuales se le conocía, ya que en la literatura de sismología en inglés se hacía referencia a ellas como: ondas de compresión o presión (*Pressure Waves*) u ondas primarias (*Primary Waves*). La onda sísmica P es la más veloz entre todas las ondas sísmicas y por tanto la primera en ser detectada en las estaciones sísmicas, por esta cualidad fueron nombradas como ondas primarias. La onda sísmica P se desplaza a través de sólidos, gases y líquidos, por lo cual, la onda P es capaz de atravesar mares, océanos y las capas líquidas de la Tierra. El movimiento de la onda primaria al propagarse es de contracción y expansión, (ver figura 2.2) justo como lo hacen las ondas sonoras que desplazan y contraen el aire a su paso, y esta es la razón de que también se les conozca como *ondas de presión*, *ondas de compresión* o *ondas elásticas*, por el movimiento de contracción y expansión descrito anteriormente. El movimiento que se produce en el suelo debido al desplazamiento de las ondas primarias es paralelo a la dirección de desplazamiento de la onda primaria, es decir, si suponemos que las ondas primarias se mueven en el plano  $xy$ , entonces el movimiento que se detectará por el sismómetro o acelerómetro en la estación sísmica será en ese mismo plano, no habrá movimiento en el eje  $z$ .



**Figura 2.2:** *Movimiento de las ondas primarias.* Adaptado de *Passage of P waves through the earth's crust*, traducida al idioma español, imagen por Giuseppe Olivadoti, 2001, Analog Dialogue (<https://www.analog.com/jp/analog-dialogue/articles/sensing-analyzing-and-acting-during-an-earthquake.html#>)

## Capítulo 2. Marco teórico

---

Al tomar la divergencia de la ecuación 2.9 y hacer uso de la siguiente identidad vectorial:  $\nabla \cdot (\nabla \times \Psi) = 0$ , se obtiene la siguiente expresión:

$$\frac{\partial^2(\nabla \cdot \mathbf{u})}{\partial t^2} = \frac{\lambda + 2\mu}{\rho} \nabla^2(\nabla \cdot \mathbf{u}) \quad (2.10)$$

La misma ecuación se puede reescribir de la siguiente forma:

$$\nabla^2(\nabla \cdot \mathbf{u}) - \frac{1}{\alpha^2} \frac{\partial^2(\nabla \cdot \mathbf{u})}{\partial t^2} = 0 \quad (2.11)$$

De las dos ecuaciones anteriores se llega a la siguiente expresión que define la velocidad,  $\alpha$ , de la onda primaria:

$$\alpha^2 = \frac{\lambda + 2\mu}{\rho} \quad (2.12)$$

De la ecuación 2.12 se tiene que  $\lambda$  representa el primer parámetro de Lamé y  $\mu$  el segundo parámetro de Lamé o módulo de rigidez en un medio elástico y  $\rho$  la densidad del medio. Los dos parámetros de Lamé son una parametrización del módulo elástico de un material homogéneo e isotrópico y ambos parámetros de Lamé se relacionan con el módulo de compresibilidad por medio de la siguiente expresión:  $K = \lambda + \frac{2}{3}\mu$ , por tanto, es posible expresar la velocidad de onda P de la siguiente forma:

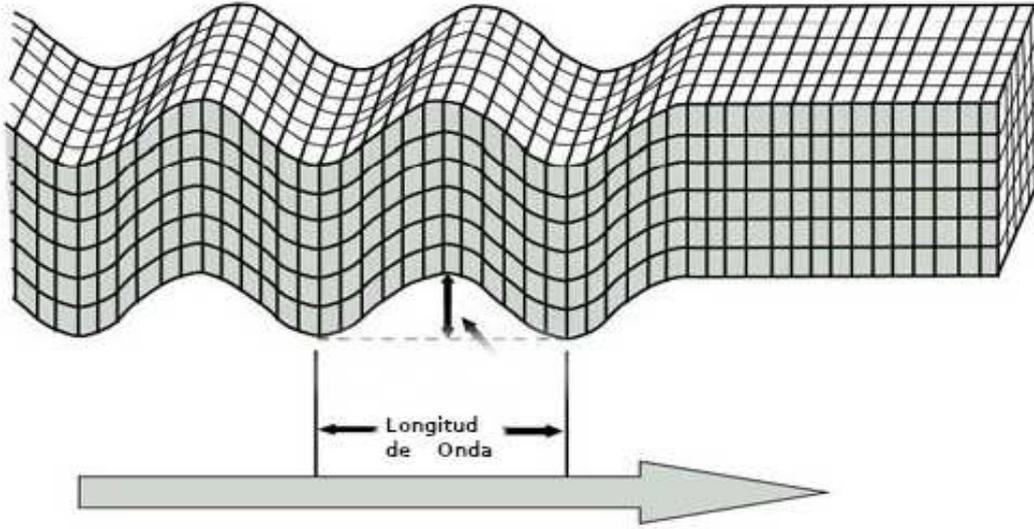
$$\alpha^2 = \frac{K + \frac{4}{3}\mu}{\rho} \quad (2.13)$$

### 2.2.4. Ondas S

Las ondas sísmicas S o secundarias, son conocidas en la literatura de sismología en inglés como: ondas secundarias (*Secondary Waves*) u ondas de deformación o corte (*Shear Waves*), esa es la razón de que se abrevien y se conozcan más habitualmente como ondas S. Las ondas sísmicas S son las segundas en ser detectadas por una estación sísmica ante la ocurrencia de un terremoto, de ahí que se conozcan como ondas secundarias. A diferencia de las ondas P, las ondas S no pueden desplazarse a través de medios líquidos y esto es debido a que la principal fuerza de restablecimiento proviene de las fuerzas de deformación, de ahí que también sean conocidas como *Shear Waves*. Es esta propiedad la que hizo concluir a los científicos que la capa externa del núcleo de la Tierra es líquida. El movimiento que las ondas sísmicas S provocan en el suelo por el cual se propagan es perpendicular a su desplazamiento, es decir, si las ondas S se propagan en el plano  $xy$  el movimiento del suelo se sentirá en el eje  $z$ , justo como sucede cuando

se agita por un extremo una cuerda atada en el extremo opuesto, la onda que se forma se desplaza de extremo a extremo moviéndose de arriba abajo (ver figura 2.3).

### Onda S



**Figura 2.3:** *Movimiento de las ondas secundarias.* Adaptado de *Passage of S waves through the earth's crust.*, traducida al idioma español, imagen por Giuseppe Olivadoti, 2001, Analog Dialogue (<https://www.analog.com/jp/analog-dialogue/articles/sensing-analyzing-and-acting-during-an-earthquake.html#>)

Al tomar el rotacional de la ecuación 2.9 y usar la siguiente identidad vectorial:  $\nabla \times (\nabla \phi) = 0$ , se obtiene la siguiente ecuación:

$$\frac{\partial^2(\nabla \times \mathbf{u})}{\partial t^2} = -\frac{\mu}{\rho} \nabla \times \nabla \times (\nabla \times \mathbf{u}) \quad (2.14)$$

Ahora, si se usa la identidad vectorial  $\nabla \times \nabla \times \mathbf{u} = \nabla \nabla \cdot \mathbf{u} - \nabla^2 \mathbf{u}$  y también  $\nabla \cdot (\nabla \times \mathbf{u}) = 0$ , se obtiene la siguiente expresión:

$$\frac{\partial^2(\nabla \times \mathbf{u})}{\partial t^2} = \frac{\mu}{\rho} \nabla^2(\nabla \times \mathbf{u}) \quad (2.15)$$

La misma ecuación se puede describir de la siguiente forma:

$$\nabla^2(\nabla \times \mathbf{u}) - \frac{1}{\beta^2} \frac{\partial^2(\nabla \times \mathbf{u})}{\partial t^2} \quad (2.16)$$

## Capítulo 2. Marco teórico

---

De las dos ecuaciones anteriores se llega a la siguiente expresión que define la velocidad,  $\beta$ , de la onda secundaria:

$$\beta^2 = \frac{\mu}{\rho} \quad (2.17)$$

Donde  $\mu$  es el segundo parámetro de Lamé o el módulo de rigidez del medio elástico y  $\rho$  es la densidad del medio.

De la ecuaciones 2.13 y 2.17 es fácil observar que  $\alpha > \beta \quad \forall \quad K > -\frac{1}{3}\mu$  (Ver figura 2.4).

Material	P wave velocity (m/s)	S wave velocity (m/s)
Steel	6100	3500
Concrete	3600	2000
Granite	5500–5900	2800–3000
Basalt	6400	3200
Sandstone	1400–4300	700–2800
Limestone	5900–6100	2800–3000
Sand (unsaturated)	200–1000	80–400
Sand (saturated)	800–2200	320–880

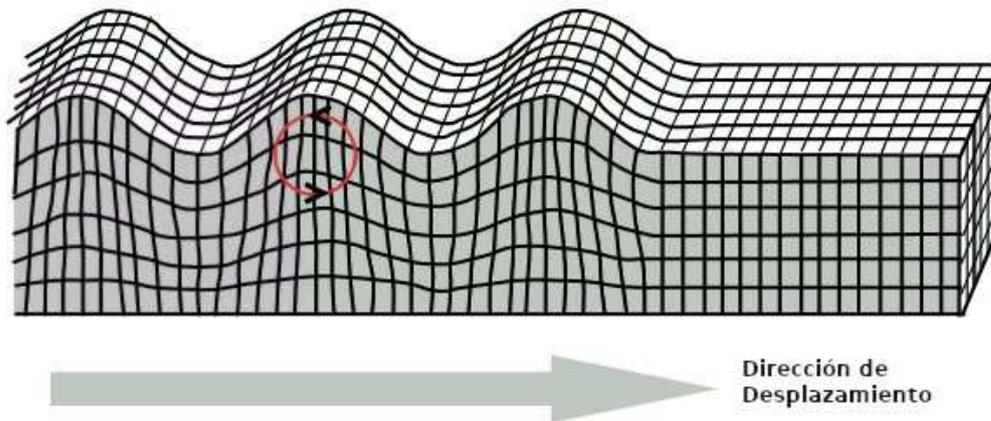
**Figura 2.4:** *Velocidades de las ondas P y S en diferentes materiales [30].*

### 2.2.5. Ondas Rayleigh

Las ondas sísmicas Rayleigh son nombradas en honor de John William Strutt, Lord Rayleigh, ya que él predijo la existencia de este tipo de ondas matemáticamente en 1885. Las ondas sísmicas Rayleigh producen un movimiento tipo elíptico en las partículas de suelo contrario a la dirección de desplazamiento, justo como lo hacen las olas en el mar. El movimiento es lado a lado y de arriba abajo en la misma dirección en la que se propagan (ver figura 2.5), por tanto, estas ondas sísmicas se captan en las tres dimensiones espaciales de movimiento del sismograma (Norte-Sur, Este-Oeste y Vertical). Gran parte del movimiento del suelo que se siente en un evento sísmico es debido a las ondas sísmicas de Rayleigh, ya que su amplitud de

onda y periodo suelen ser más grande que la de las demás ondas sísmicas. Estas ondas sísmicas son más lentas que las ondas de cuerpo S, aproximadamente en un 10% en medios elásticos homogéneos. En un medio compuesto de diferentes capas se tendrá que la velocidad de las ondas de Rayleigh depende de su frecuencia y longitud de onda.

### Onda Rayleigh

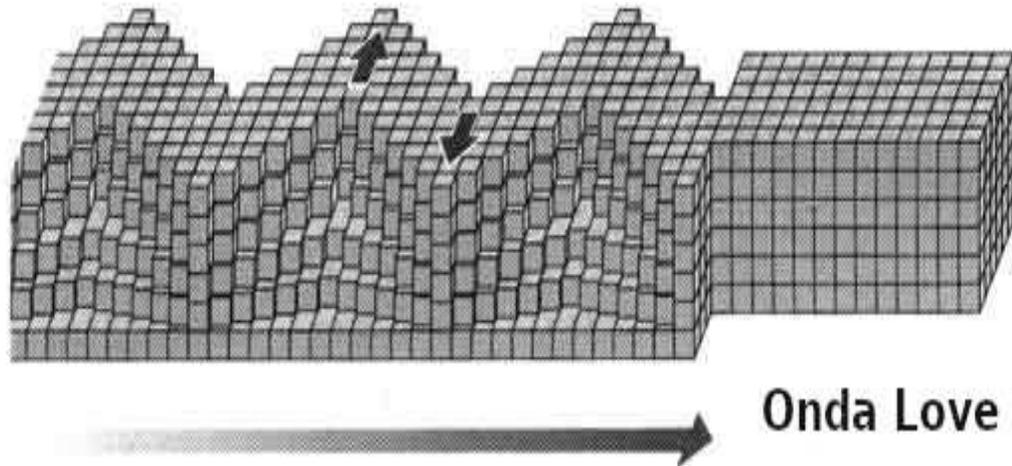


**Figura 2.5:** *Movimiento de las ondas Rayleigh.* Adaptado de *Passage of Rayleigh waves through earth's crust.*, traducida al español, imagen por Giuseppe Olivadoti, 2001, Analog Dialogue (<https://www.analog.com/jp/analog-dialogue/articles/sensing-analyzing-and-acting-during-an-earthquake.html#>)

#### 2.2.6. Ondas Love

Las ondas Love, son nombradas en honor de Augustus Edward Hough Love, ya que él predijo su existencia matemáticamente en 1911. Estas ondas sísmicas producen un movimiento ondulatorio perpendicular a la dirección desplazamiento de la onda sísmica, similar al movimiento que describen las serpientes al desplazarse por el suelo (ver figura 2.6). Las ondas sísmicas Love se desplazan a una velocidad menor que las ondas sísmicas primarias y secundarias, pero a su vez, mayor que las ondas sísmicas Rayleigh. La amplitud de estas ondas sísmicas decrecen con la profundidad del foco del terremoto. La amplitud decrece en un factor de  $\frac{1}{\sqrt{r}}$ , donde  $r$  es la distancia que ha viajado la onda sísmica. Un terremoto lo suficientemente fuerte puede generar ondas sísmicas Love que pueden recorrer la Tierra varias veces antes de desvanecerse; las ondas superficiales se desvanecen más lentamente que

las ondas de cuerpo. Las ondas Love son las más destructivas en las inmediaciones del epicentro de un evento sísmico.



**Figura 2.6:** *Movimiento de las ondas Love.* Adaptado de *Passage of Love waves through earth's crust.*, traducida al español, imagen por Giuseppe Olivadoti, 2001, Analog Dialogue (<https://www.analog.com/jp/analog-dialogue/articles/sensing-analyzing-and-acting-during-an-earthquake.html#>)

### 2.2.7. Sismógrafos

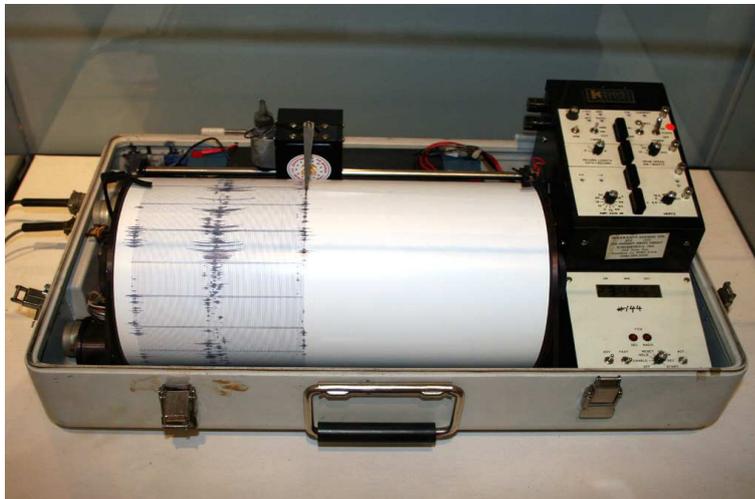
Los sismógrafos son instrumentos que miden el movimiento del suelo provocado por terremotos, erupciones volcánicas, explosiones, etc. Consisten de un dispositivo que detecta el movimiento del suelo, llamado sismómetro, el cual se acopla a un sistema de monitoreo. Los sismógrafos basan su funcionamiento en el principio de inercia.

El primer paso que se dio hacia la invención del sismógrafo fue el *sismoscopio*, antiguo dispositivo que fue inventado por el filósofo chino Chang Heng en el año 132 después de Cristo, sin embargo, este dispositivo no grababa lecturas de las ondas sísmicas, tan solo indicaba si un terremoto estaba ocurriendo. Se reporta que este dispositivo detectó un terremoto que ocurrió a unos 644 km de distancia, evento que no fue sentido en el lugar donde estaba ubicado el sismoscopio.

El primer sismógrafo fue inventado en el año 1890. El sismógrafo para su funcionamiento se monta en una superficie fija de tal forma que cuando el suelo se sacuda, el dispositivo entero se sacuda también, excepto por una masa adherida

a un resorte y a la masa va adherida un bolígrafo, la masa permanece fija en el mismo sitio. Al ocurrir un terremoto el dispositivo se mueve debajo de la masa dando lugar a que el movimiento relativo entre la masa y el resto del dispositivo sea registrado en un rollo de papel (ver figura 2.7). En la actualidad los sismógrafos basan su funcionamiento en señales eléctricas producidas por el movimiento del suelo con respecto de la masa, dichas señales eléctricas son almacenadas en una computadora. Al hacer modificaciones en cuanto a la masa, el resorte, bolígrafo y la estructura del sismógrafo se puede lograr que los sismógrafos sean capaces de grabar el movimiento del suelo en las tres componentes espaciales de un plano tridimensional. Los sismógrafos actuales son tan sensibles que pueden detectar movimientos tan pequeños como  $\frac{1}{10^7}$  cm en sitios donde no exista ruido en la señal.

Algunas estaciones sísmicas también incluyen un dispositivo de medición llamado acelerómetro, los cuales son diseñados para medir movimientos sísmicos de amplitudes y frecuencias grandes. Es común que en estaciones sísmicas cercanas a zonas propensas a terremotos se integren acelerómetros a sus equipos de medición puesto que los sismómetros al ser dispositivos de medición muy sensibles suelen saturarse. Las lecturas recolectadas por los acelerómetros, tal como el nombre del instrumento lo indica, son aceleraciones del terreno, lecturas que posteriormente pueden ser transformadas a lecturas de velocidad o movimiento.



**Figura 2.7:** *Sismógrafo*. Imagen del usuario Yamaguchi de Wikipedia (CC BY-SA 3.0), recuperada de [https://commons.wikimedia.org/wiki/File:Kinematics\\_seismograph.jpg](https://commons.wikimedia.org/wiki/File:Kinematics_seismograph.jpg)

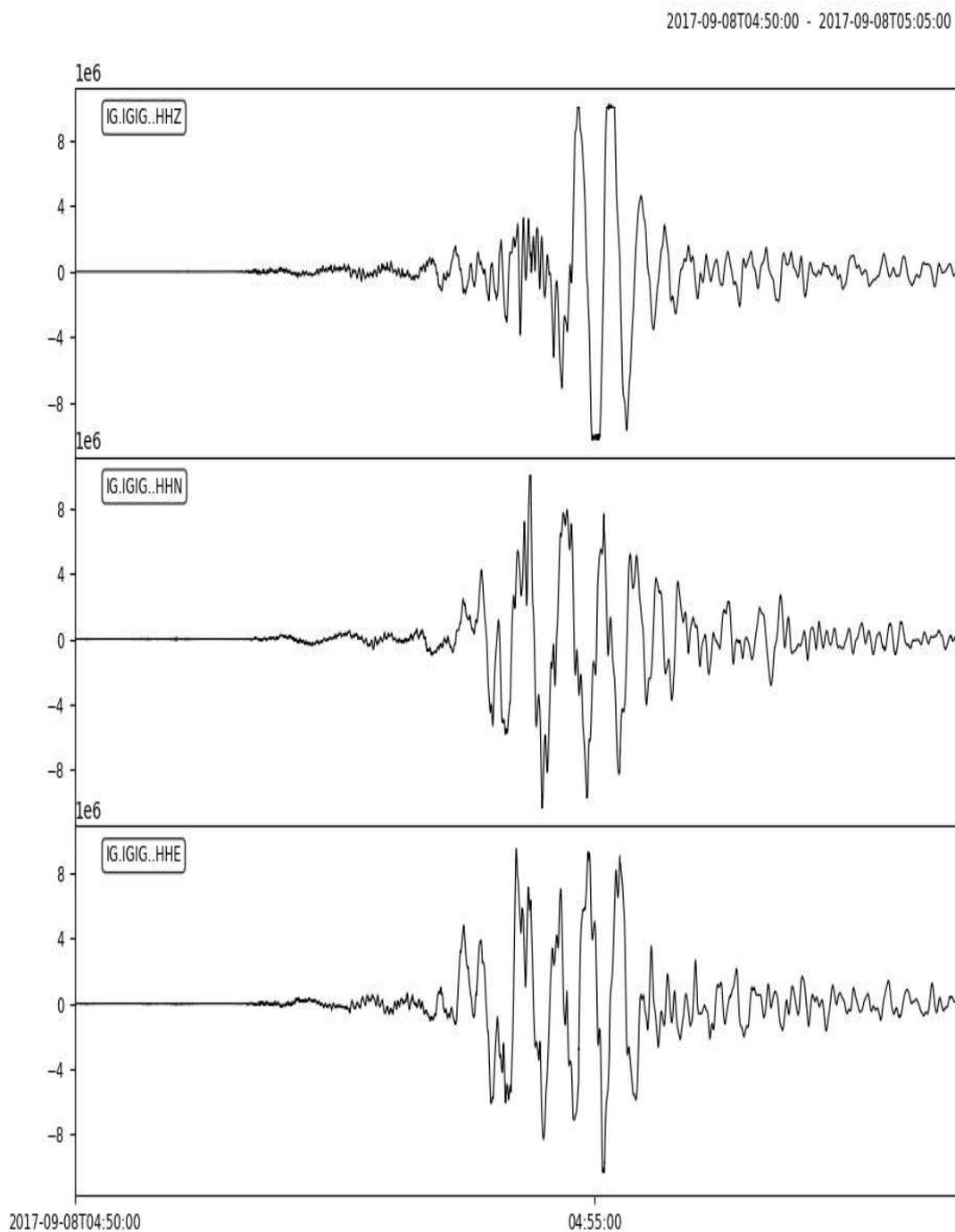
### 2.2.8. Sismogramas

En un sismograma se almacenan las lecturas del movimiento del suelo medidas por sismómetros o acelerómetros a través del tiempo. Los primeros sismógrafos recopilaban sus lecturas con papel y tinta, actualmente las lecturas se almacenan en archivos codificados en diferentes formatos digitales. De los diferentes formatos digitales existentes para almacenamiento de lecturas sismológicas, en este proyecto de tesis, se descargaron todos los archivos en formato *.sac* (*seismic analysis code*). Para cada terremoto las estaciones sísmicas obtienen sismogramas en las tres componentes espaciales. En sismología cada sismograma se guarda con una nomenclatura que da información de la fecha y hora del evento sísmico, tipo del sensor usado, e información de la estación sísmica, en particular, para distinguir las componentes espaciales se codifican con la siguiente nomenclatura: Z ó 3 (eje vertical), E ó 1 (Este-Oeste) y N ó 2 (Norte-Sur), los cuales nos dan información completa del movimiento del suelo en el plano tridimensional, información que es necesaria puesto que algunas ondas sísmicas no tienen presencia en las tres componentes espaciales. Estos archivos digitales codificados, además de las lecturas de los sensores contienen datos relativos al tipo de sensor que se está usando para la recopilación de las lecturas, ubicación de la estación sísmica, la red de sismógrafos a la cual pertenece, nombre de la estación sísmica y otros datos más relativos a la estación sísmica y sus sensores.

En los sismogramas es posible (para sismólogos experimentados) ubicar los diferentes tipos de ondas sísmicas y determinar la magnitud de los eventos sísmicos, epicentros e hipocentros de los mismos. Para determinar la ubicación de un evento sísmico se requiere de tres sismogramas de tres estaciones sísmicas diferentes de tal forma que por medio de triangulación se ubica el lugar donde ocurrió el evento sísmico haciendo uso de la diferencia en tiempos de llegada de las ondas primarias y secundarias. El cálculo de la magnitud del terremoto se basa en la amplitud de las ondas superficiales ya que éstas son las de mayor amplitud y las que causan el mayor daño en un terremoto. Para leer los archivos con extensión *.sac* se dispone de varios programas, entre ellos Matlab, Python, SAC, etc.

Para la realización de este trabajo de tesis se utilizará una librería de Python llamada *Obspy*, la cual nos permite acceder a todos los datos de los archivos extensión *.sac*. La librería también ofrece muchas funcionalidades más que permiten la manipulación de los datos de los sismogramas por medio de funciones y métodos útiles para el análisis y visualización de gráficos de sismogramas.

La figura 2.8 en la página 23 fue generada con la librería *Obspy* de Python, en la figura se puede apreciar en la parte superior izquierda la codificación con la cual se guarda cada sismograma. Las dos primeras letras identifican la red de



**Figura 2.8:** Sismogramas obtenidos de la estación sísmica IGIG en las tres componentes espaciales (Vertical, Norte-Sur y Este-Oeste) para el terremoto del 19 de septiembre de 2017.

estaciones sísmicas, las siguientes cuatro letras se refieren a la estación sísmica, las siguientes dos letras hacen referencia al tipo de instrumento que fue utilizado y la última letra a la componente espacial. En la parte superior de la gráfica se aprecia la fecha y hora de inicio y la fecha y hora final del sismograma graficado. En el eje del tiempo se observa la hora con minutos y segundos.

### 2.3. Series de Tiempo

Una serie de tiempo es una colección de datos obtenidos a través de la observación y medición de fenómenos bien definidos, las mediciones se obtienen con intervalos de tiempo predeterminados y regulares, por ejemplo, la medición de la temperatura ambiental cada 30 minutos durante todo el día es una serie de tiempo. Otro ejemplo es la colección de lecturas recopiladas en un sismograma durante un evento sísmico con frecuencia de muestreo de 100 Hz, es decir, 100 lecturas cada segundo. La frecuencia de muestreo en un sismógrafo depende del instrumento que se use, el dato se especifica en los archivos con terminación *.sac*.

Una serie de tiempo puede ser analizada y descompuesta en tres componentes distintivas principales:

- *Tendencia*. Dirección que asumen los datos de la serie de tiempo a largo plazo. La tendencia puede ser lineal o no lineal. La tendencia puede asumir tres estados: positiva, negativa o estacionaria.
- *Estacionalidad*. Comportamientos repetitivos y sistemáticos que se presentan cada cierto periodo de tiempo en la serie de tiempo.
- *Aleatoriedad*. Pequeñas fluctuaciones en los datos de la serie de tiempo de origen indeterminado.

Existen varios modelos para analizar y descomponer una serie de tiempo en sus diferentes componentes o atributos, entre los más usados están:

- *Modelos Aditivos*
- *Modelos Multiplicativos*, y
- *Modelos Híbridos de los dos anteriores*.

Supongamos que se tiene una serie de tiempo  $O_t$  y se desea descomponer usando el modelo aditivo, entonces la serie de tiempo se representa de la siguiente manera:

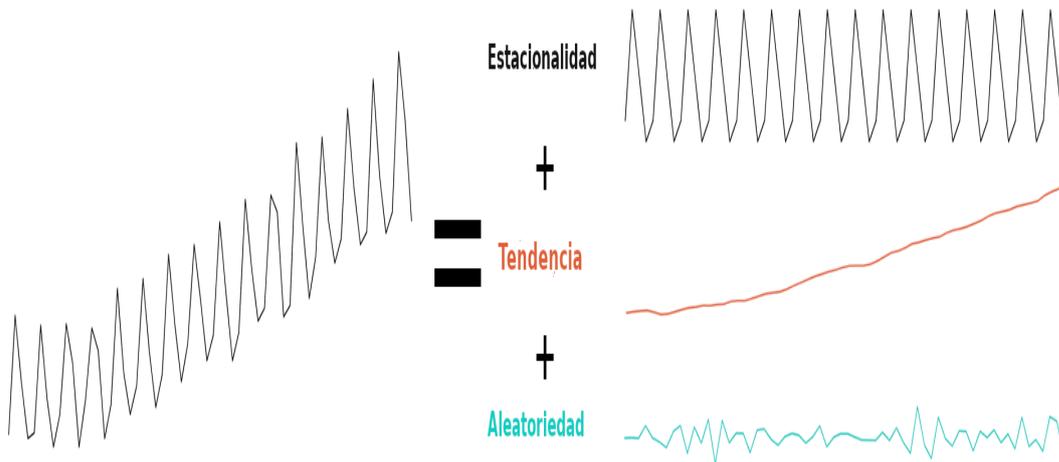
$$O_t = T_t + S_t + R_t \quad (2.18)$$

Si se quisiera descomponer usando el modelo multiplicativo se tendría lo siguiente:

$$O_t = T_t \cdot S_t \cdot R_t \quad (2.19)$$

Donde  $T_t$  es la tendencia,  $S_t$  la estacionalidad y  $R_t$  la aleatoriedad.

En la figura 2.9 se puede observar claramente las componentes que constituyen una serie de tiempo, la cual fue descompuesta usando el modelo aditivo. Hay que recalcar que no siempre es posible descomponer una serie de tiempo en sus respectivas componentes, tal es el caso de las series de tiempo que se obtienen de las lecturas de los sismogramas, en las cuales no es posible apreciar una tendencia o una estacionalidad bien definidas, lo cual se puede observar en la figura 2.8 de la página 23.



**Figura 2.9:** Componentes de una serie de tiempo: Tendencia, estacionalidad y aleatoriedad.

## 2.4. Redes Neuronales

### 2.4.1. Introducción

En esta sección se desarrollarán conceptos y definiciones que prepararán el camino para presentar los temas objetivos de esta sección que son las redes neuronales recurrentes, redes neuronales convolucionales unidimensionales y mecanismos de atención, y, su aplicación a secuencias de datos, tales como: series de tiempo y sucesiones. Se presentará el modelo del perceptrón y las redes neuronales artificiales,

se desarrollaran los correspondientes temas y definiciones hasta llegar a las redes neuronales recurrentes, en particular, redes neuronales recurrentes LSTM y GRU, las cuales se aplicarán a las series de tiempo obtenidas de los sismogramas con el objetivo de realizar predicciones. Como apoyo a las redes neuronales recurrentes se hará uso de redes neuronales convolucionales unidimensionales y mecanismos de atención.

### 2.4.2. Aprendizaje Profundo (*Deep Learning*)

El Aprendizaje Profundo (*Deep Learning*) es una rama del Aprendizaje Máquina (*Machine Learning*) que a su vez es una rama de la Inteligencia Artificial (*Artificial Intelligence*), así que se empezará por definir cada una de estas ramas de la Inteligencia Artificial.

La Inteligencia Artificial, a su vez, es una rama de las Ciencias de la Computación. El objetivo de la Inteligencia Artificial es desarrollar máquinas ‘inteligentes’ que sean capaces de elaborar o realizar tareas que hasta ese momento se creía solo los humanos podían llevar a cabo. En 1950 Alan Turing con su artículo: ‘*Computing Machinery and Intelligence*’ [22] y su ‘*Test de Turing*’ revolucionaron las Ciencias de la Computación con una simple pregunta: ‘¿*Pueden las máquinas ser capaces de pensar?*’. El trabajo de Alan Turing fue tan revolucionario que las metas y objetivos fundamentales de la Inteligencia Artificial aún descansan en ese trabajo. Las demandas y expectativas tan amplias que se han postrado sobre esta rama de las Ciencias de la Computación, además de la evolución constante de la misma han hecho que no se tenga una única definición universalmente aceptada de lo que es la Inteligencia Artificial.

El Aprendizaje Máquina surge como una alternativa a la Inteligencia Artificial Simbólica, de la cual se desprenden programas que impactaron al mundo, tal como aquellos que eran capaces de sostener un juego de ajedrez y de vencer a maestros mundiales del ajedrez, este tipo de programas especializados en una sola tarea se conocen como: *Sistemas Expertos*.

El Aprendizaje Máquina tiene como objetivo lograr que los sistemas o programas aprendan y mejoren en el desempeño de sus tareas o actividades a través de la experiencia, sin haber sido programados explícitamente para realizar dichas tareas, contrario a lo que sucede con los sistemas expertos, los cuales son explícitamente programados con conjuntos de reglas específicas para una tarea en particular (ver figura 2.10).

Lo que se busca en Aprendizaje Máquina es que los programas aprendan por sí mismos al ser expuestos a datos de los cuales extraerán la información y experiencias necesarias para realizar con éxito su actividad.

El objetivo principal es permitir que las máquinas aprendan automáticamente sin intervención humana. Un sistema basado en Aprendizaje Máquina es entrenado en lugar de ser explícitamente programado.



**Figura 2.10:** *Inteligencia Artificial Simbólica y Aprendizaje Máquina.*

Aprendizaje Profundo es una rama de Aprendizaje Máquina, una nueva aproximación a la forma de aprender representaciones la cual se enfoca en aprender capa por capa representaciones más refinadas de la tarea en cuestión. Una analogía sería que en esta nueva aproximación al aprendizaje se va refinando la información aprendida capa a capa o nivel por nivel, de tal forma que en cada nueva capa que se ha superado la información se va refinando más y más con el objetivo de obtener el aprendizaje y resultados deseados. En Aprendizaje Profundo las capas o niveles de los cuales se habla se refieren a modelos llamados: redes neuronales. Las redes neuronales se apilan una sobre otras y de ahí el porqué se usa el término de ‘profundo’. El Aprendizaje Profundo no necesariamente se tiene que basar en el modelo de redes neuronales, sin embargo, ha sido el más exitoso y ampliamente usado por la comunidad científica y estudiantil.

### 2.4.3. Técnicas de aprendizaje automático

Tres técnicas de aprendizaje automático son las más comunes en el campo del Aprendizaje Máquina: *Aprendizaje Supervisado*, *Aprendizaje no Supervisado* y *Aprendizaje por Reforzamiento*, se describirá las principales características de estas tres técnicas.

### Aprendizaje Supervisado

El término ‘supervisado’ se refiere a que del conjunto de datos de entrenamiento se formaran dos subconjuntos, el subconjunto de datos que se usa a la entrada del algoritmo y el subconjunto de datos a la salida del algoritmo que son los resultados que se espera obtener del algoritmo. El subconjunto de datos a la salida del algoritmo está plenamente definido y se sabe lo que se espera obtener de los datos del subconjunto de entrada de antemano, de ahí el nombre de ‘supervisado’. A través del entrenamiento se espera que el algoritmo aprenda una representación útil que caracterice y modele una función o relación entre los datos de entrada y los datos de salida [24].

Los problemas que entran en la categoría de aprendizaje supervisado pueden ser divididos en dos grupos:

- a) **Clasificación.** Un problema de aprendizaje supervisado concerniente a clasificación se presenta cuando la salida del algoritmo es una categoría, es decir, una variable discreta o valores que se interpretan como pertenecientes a algún grupo en específico.
- b) **Regresión o Predicción.** Un problema de aprendizaje supervisado referente a regresión se presenta cuando a la salida del algoritmo se esperan números reales continuos.

### Aprendizaje no Supervisado

Aprendizaje no supervisado se presenta cuando se tienen datos de entrada al algoritmo pero no se cuenta con los respectivos datos de salida, a diferencia del aprendizaje supervisado donde se sabe de antemano lo que se espera obtener a la salida del algoritmo. La meta del aprendizaje no supervisado es aprender la estructura subyacente o la distribución de los datos de entrada. Se espera que después del entrenamiento el algoritmo haya aprendido características y estructuras representativas de los datos de entrada.

Los algoritmos de aprendizaje no supervisado se pueden subdividir en dos grupos:

- a) **Agrupamiento (*Clustering*).** El objetivo que se tiene es lograr agrupar datos por sus características, por ejemplo, agrupar perros por su raza.
- b) **Asociación (*Association*).** El objetivo es obtener reglas que caractericen a grandes porciones de datos en el conjunto de entrenamiento, por ejemplo, agrupar perros por su raza, pero también por su color de pelaje.

### Aprendizaje por Reforzamiento

En aprendizaje por reforzamiento el objetivo es desarrollar un sistema (agente) que mejore su desempeño basado en interacciones con su ambiente. De las interacciones del sistema con el ambiente se premia ó se penaliza por medio de una señal de recompensa, de tal forma que el sistema trata siempre de maximizar la recompensa obtenida de la interacción con su ambiente, y por tanto, mejorar su desempeño. La forma de interactuar del sistema con el ambiente puede ser a prueba y error ó se puede tomar una aproximación con una orientación premeditada de la interacción sistema-ambiente.

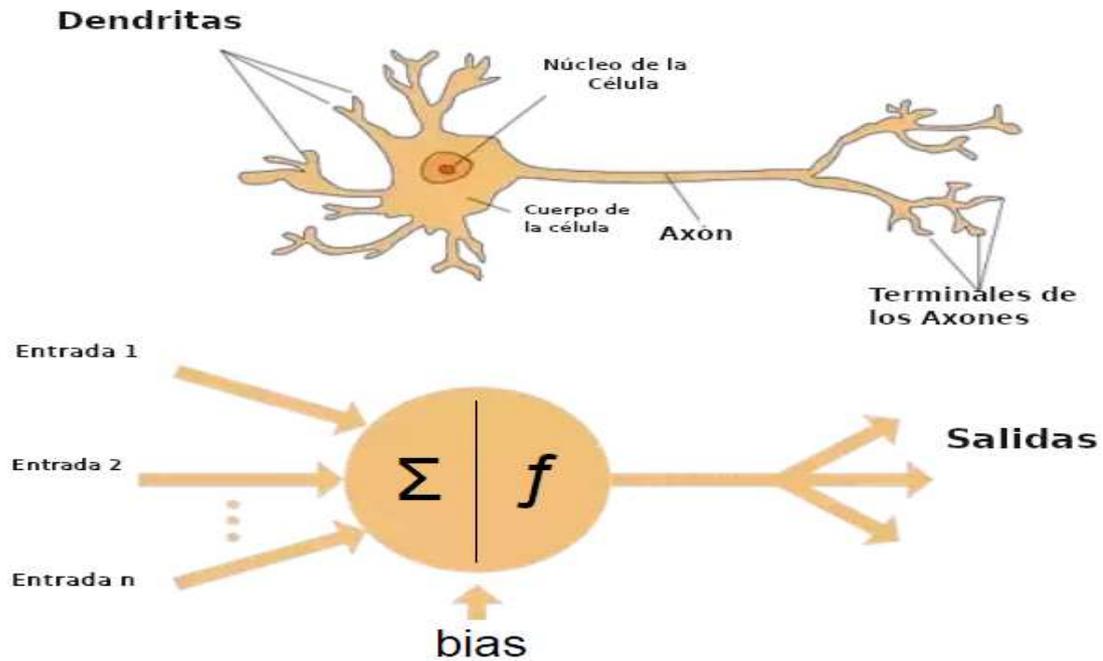
Un ejemplo de aprendizaje por reforzamiento es un juego de ajedrez entre un programa de computadora y un humano, en este ejemplo el programa decide basado en una serie de movimientos predeterminados dependientes del estado de las piezas del tablero, y la recompensa puede basarse en la efectividad de cada movimiento o en si el programa resulta vencedor o perdedor.

#### 2.4.4. Neurona biológica

El modelo de las redes neuronales artificiales toma su inspiración en el funcionamiento de la neurona biológica y de su interacción con otras neuronas cercanas a ella. Antes de presentar el modelo de redes neuronales artificiales se presentará la fuente de inspiración de dicho modelo.

Las neuronas biológicas son las unidades básicas del sistema nervioso (ver figura 2.11 en página 30), principalmente se componen de tres partes: dendrita, el cuerpo o soma y terminales del axón. Las dendritas reciben señales de neuronas circundantes, donde cada dendrita se conecta a una neurona. Soma es el cuerpo de la célula donde se procesan las señales de entrada a la neurona y se decide si la neurona debe mandar una señal de salida como respuesta a los estímulos de entrada o no. Axón es responsable de obtener las señales procesadas en el soma de la neurona. La sinapsis es el proceso de conexión entre axones de una neurona y dendritas de otras neuronas circundantes. Las señales de entrada que recibe la neurona pueden ser excitatorias (hacen que la neurona se active, que genere un impulso eléctrico que se transmite a través del axón) o inhibitorias (mantienen a la neurona inactiva).

La mayoría de las neuronas reciben muchas señales de entrada a través de sus dendritas. El que una neurona se active y mande una señal de excitación o dispare depende de la suma de todas las señales que recibe y si dichas señales son excitatorias o inhibitorias. Para que la neurona se excite y dispare se necesita que la suma de las señales excitatorias e inhibitorias rebase cierto umbral de disparo en un corto periodo de tiempo.



**Figura 2.11:** *Neurona biológica y neurona artificial.* Reproducido de “What is the differences between artificial neural network (computer science) and biological neural network?”, por Abhi S, 2018, (<https://www.quora.com/What-is-the-differences-between-artificial-neural-network-computer-science-and-biological-neural-network>). Obra de Dominio Publico.

### 2.4.5. Perceptrón

La neurona artificial de Rosenblatt [6] (también conocida como perceptrón) es la unidad básica de una red neuronal artificial. La neurona artificial de Rosenblatt que actualmente se conoce es una versión mejorada de la versión de McCulloch-Pitts [33]. La neurona artificial es modelada en base a las propiedades y funciones básicas de una neurona biológica. Esta unidad de procesamiento se desempeña en dos etapas: primero realizando una sumatoria priorizada a través de pesos sinápticos de sus entradas y de un término llamado ‘bias’, y posteriormente aplicando una función no lineal al resultado de la sumatoria de entradas (ver figura 2.11). La fórmula que resume lo explicado anteriormente es la siguiente:

$$y_j = f_j \left( \sum_{j=1}^n w_{ij} x_j + b_j \right) \quad (2.20)$$

Donde:

- $x_j$  son las entradas a la neurona  $j$ -ésima.
- $w_{ij}$  son los pesos sinápticos.
- $b_j$  parámetro conocido como *bias*.
- $y_j$  la salida de la neurona  $j$ -ésima.
- $f_j$  la función de activación de la  $j$ -ésima neurona.

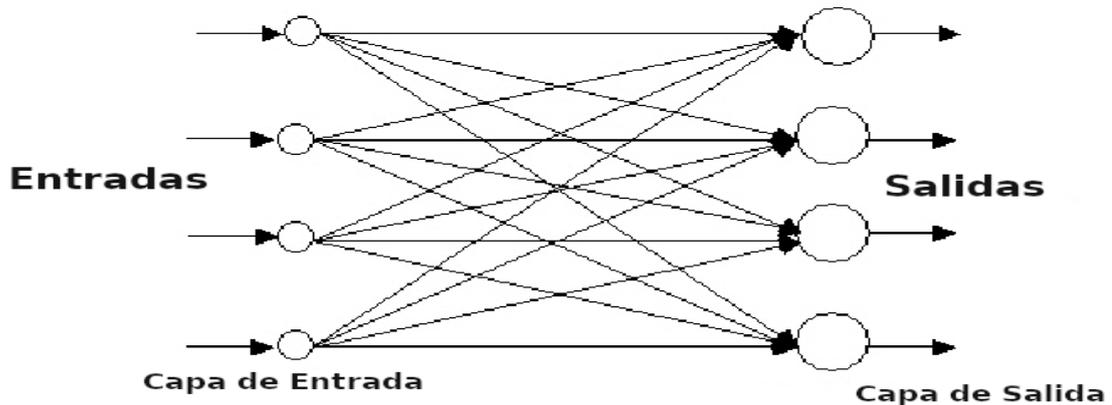
A diferencia de los sistemas computacionales convencionales, los cuales tienen instrucciones específicas para realizar alguna tarea, las neuronas artificiales necesitan ser entrenadas para ajustar sus parámetros iniciales (pesos sinápticos y *bias*) de tal forma que se cumpla con la función esperada de ellas.

### 2.4.6. Redes neuronales artificiales (RNA)

Las redes neuronales artificiales han sido aplicadas exitosamente en varios y diversos campos de la ciencia. Clasificación, reconocimiento, procesamiento de señales e imágenes, regresión, control de sistemas y predicción en la bolsa de valores [27], son solo algunas de las aplicaciones que se les ha dado a las redes neuronales artificiales.

Al conectar múltiples neuronas artificiales surge el verdadero poder computacional de las redes neuronales artificiales. La arquitectura más común para conectar múltiples neuronas artificiales es hacerlo por capas verticales. Cada neurona en una capa vertical se conecta con el resto de las neuronas de la siguiente capa neuronal, a este tipo de arquitectura se le conoce como totalmente conectada (ver figura 2.12 en página 32). A la capa donde van conectadas los datos de entrada a la red neuronal se le conoce como capa de entrada y no se realiza ningún tipo de operación en ella, tan solo distribuye las entradas a la siguiente capa. La siguiente capa en la figura 2.12 es la capa de salida, de la capa de entrada a la capa de salida para cada conexión de las neuronas se tienen pesos sinápticos únicos que van a determinar la injerencia que tengan las neuronas de la capa de entrada en las neuronas de la capa de salida.

Si se desea conseguir un mayor nivel de poder computacional solo basta con agregar más capas de neuronas artificiales entre la capa de entrada y la capa de salida (ver figura 2.13 en página 33), a estas capas intermedias de neuronas se les conoce como *capas escondidas*. A esta nueva arquitectura de redes neuronales artificiales se le conoce como *redes neuronales multicapas alimentadas hacia adelante* (*Multi-Layer Feed-Forward Neural Network*).



**Figura 2.12:** Red neuronal artificial de una sola capa totalmente conectada.

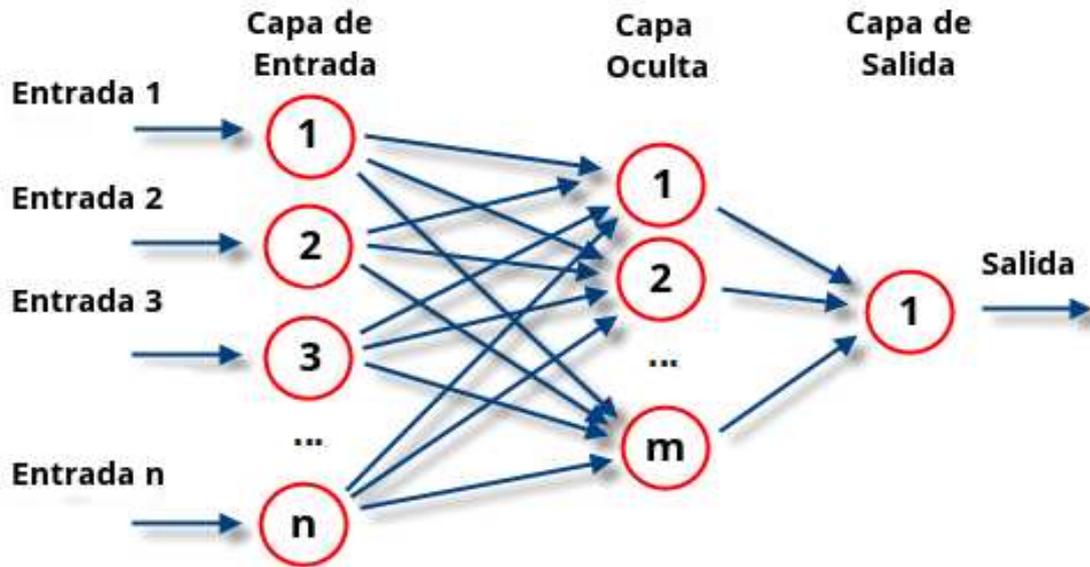
Es posible hacer varias distinciones entre las redes neuronales artificiales para clasificarlas de acuerdo a sus características:

1] Por su flujo de información entre neuronas:

- a) *Redes neuronales alimentadas hacia adelante (Feed-Forward Neural Networks)*. Como el nombre lo indica, la información fluye de la capa de entrada a la capa de salida en un solo sentido sin pasar por una neurona dos o más veces en una iteración. Estas redes neuronales alimentadas hacia adelante pueden ser separadas en dos grupos de acuerdo a su número de capas:
  - i) *Redes neuronales de capa única (single layer neural network)*. Estas redes neuronales solo constan de la capa de entrada y la capa de salida.
  - ii) *Redes neuronales de capa múltiple (multi-layer neural network)*. Estas redes neuronales constan de la capa de entrada, una o múltiples capas escondidas y la capa de salida.
- b) *Redes neuronales recurrentes (Recurrent neural networks)*. En este tipo de redes neuronales se realimenta la salida de una neurona o un grupo de neuronas a sí misma o a otras neuronas diferentes, es decir, se genera un lazo cerrado entre una o algunas neuronas en el flujo de información.

2] Por su tipo de conexión con neuronas de capas posteriores.

- a) *Totalmente conectadas*. Las neuronas de una capa anterior (a excepción de la capa de entrada) están, cada una de ellas, conectadas a cada una de las neuronas de la capa posterior.



**Figura 2.13:** Red neuronal multicapa alimentada hacia adelante totalmente conectada. Imagen del usuario Gengiskanhg de Wikipedia (CC BY-SA 3.0), recuperada de <https://commons.wikimedia.org/wiki/File:RedNeuronalArtificial.png>

- b) *Parcialmente conectadas.* No todas las neuronas de una capa anterior están conectadas a cada una de las neuronas de la capa posterior.

### 2.4.7. Entrenamiento de una red neuronal artificial

Una vez que se tiene una base de datos disponible para el entrenamiento de la red neuronal artificial se procede a dividir ese conjunto de entrenamiento en tres subconjuntos: uno para entrenamiento como tal, otro para validación del entrenamiento y uno más para prueba de la eficacia del entrenamiento, además, cada subconjunto se divide en entradas y salidas de la red neuronal, con lo cual se establece un problema de aprendizaje supervisado. Al tener los tres subconjuntos listos ahora se procede con el entrenamiento de la red neuronal, se elige por cuantas épocas <sup>1</sup> se va a repetir el entrenamiento, en otras palabras, cuantas veces la red neuronal procesará el conjunto de entrenamiento con el fin de aprender de la mejor manera posible las características que la red neuronal tiene que aprender.

Al ser una técnica perteneciente al Aprendizaje Profundo, las redes neuronales artificiales *aprenden* pesos sinápticos, o dicho de otra forma, ajustan tales parámetros al ser expuestas a ejemplos pertenecientes a un conjunto de datos por medio de

<sup>1</sup>Una época es un recorrido completo por el conjunto de entrenamiento.

entrenamiento. Las redes neuronales artificiales procesan los ejemplos a los que son expuestas y los resultados que entregan se comparan con los resultados esperados (por tal motivo el algoritmo de propagación hacia atrás es una técnica de aprendizaje perteneciente al aprendizaje supervisado), se calcula el error entre resultados esperados y resultados entregados por la red neuronal artificial, se selecciona una función de pérdida y por medio del algoritmo conocido como propagación hacia atrás (*Back-Propagation*) se actualizan los parámetros respectivos.

### 2.4.8. Propagación hacia atrás (*Back-Propagation*)

Existen dos variantes del algoritmo de propagación hacia atrás y de ellos el modo estocástico es el que se usa con mayor regularidad.

- a) *Modo Secuencial o Estocástico*. En el modo estocástico, las actualizaciones de los parámetros de entrenamiento de la red neuronal artificial se hacen conforme se va procesando cada elemento del conjunto de entrenamiento.
- b) *Modo de Porciones o Lotes*. En el modo de lotes, las actualizaciones de los parámetros de entrenamiento se realizan al finalizar una época o lote.

Hay que mencionar que el algoritmo de propagación hacia atrás no es el único algoritmo de aprendizaje que existe [21], sin embargo, sí es el más utilizado y más popular que existe para entrenar redes neuronales alimentadas hacia adelante.

El procedimiento que se lleva a cabo al aplicar el algoritmo de propagación hacia atrás es, en esencia, calcular derivadas parciales de la función de pérdida, la cual a su vez es una función que depende de los pesos sinápticos, y de esta forma buscar el mínimo error posible de la función de pérdida por medio de la actualización de los parámetros de los cuales depende la función de pérdida, sucesivamente, aplicando la regla de la cadena se va retrocediendo en las capas de la red neuronal calculando la contribución de cada neurona de capas anteriores y haciendo las respectivas actualizaciones de los pesos sinápticos [26]. Plasmando la explicación anterior en términos matemáticos se tendría lo siguiente:

Primero se elige una neurona en la capa de salida, cualquiera, se obtiene la diferencia entre el valor obtenido por la red neuronal y el valor esperado o real de una muestra del conjunto de entrenamiento:

$$e_j(n) = d_j - y_j(n) \quad (2.21)$$

Donde  $e_j(n)$  es la diferencia de un valor real  $n$ -ésimo del conjunto de entrenamiento y el calculado por la red neuronal para una neurona  $j$ -ésima en la capa de

salida,  $d_j$  es el valor real y  $y_j(n)$  es el valor calculado por la neurona usando sus pesos sinápticos actuales. Ahora, si dentro de las funciones de pérdida disponibles se elige al error medio cuadrático, se tiene lo siguiente:

$$\varepsilon_j(n) = \frac{1}{2}e_j^2(n) \quad (2.22)$$

Si se quisiera obtener el error cuadrático medio en todas las neuronas de la capa de salida de la red neuronal para calcular el error cuadrático total, entonces la fórmula es la siguiente:

$$\varepsilon(n) = \frac{1}{2} \sum_{j=1}^m e_j^2 \quad (2.23)$$

Donde  $m$  es la cantidad de neuronas en la capa de salida de la red neuronal. Si se supone que hay  $N$  elementos en el conjunto de entrenamiento, entonces el error promedio cuadrático de la red neuronal en una época es:

$$\varepsilon_{average} = \frac{1}{N} \sum_{n=1}^N \varepsilon(n) \quad (2.24)$$

Por otra parte, si se expresa la salida de la neurona  $j$ -ésima en la capa de salida en términos de sus parámetros independientes se tiene lo siguiente:

$$y_j(n) = f\left(\sum_{i=0}^k w_{li}(n) \cdot y_i(n)\right) \quad (2.25)$$

Donde  $k$  es el número total de entradas o conexiones que llegan a la neurona  $j$ -ésima de la capa anterior y  $f$  es la función de activación de la neurona  $j$ -ésima,  $w_{li}$  son los pesos sinápticos que multiplican a cada una de las salidas de las neuronas  $y_i$  de la capa anterior, es decir, el subíndice  $l$  da información acerca de la capa neuronal en la que se encuentra y el subíndice  $i$  informa acerca de la neurona en cuestión en dicha capa neuronal.

Ahora, la actualización de los pesos sinápticos que experimentará la neurona  $j$ -ésima es proporcional a la derivada parcial del error cuadrático medio de dicha neurona con respecto a sus pesos sinápticos:

$$\frac{\partial \varepsilon(n)}{\partial w_{li}(n)} \quad (2.26)$$

Finalmente, la fórmula para la actualización de los pesos sinápticos es la siguiente:

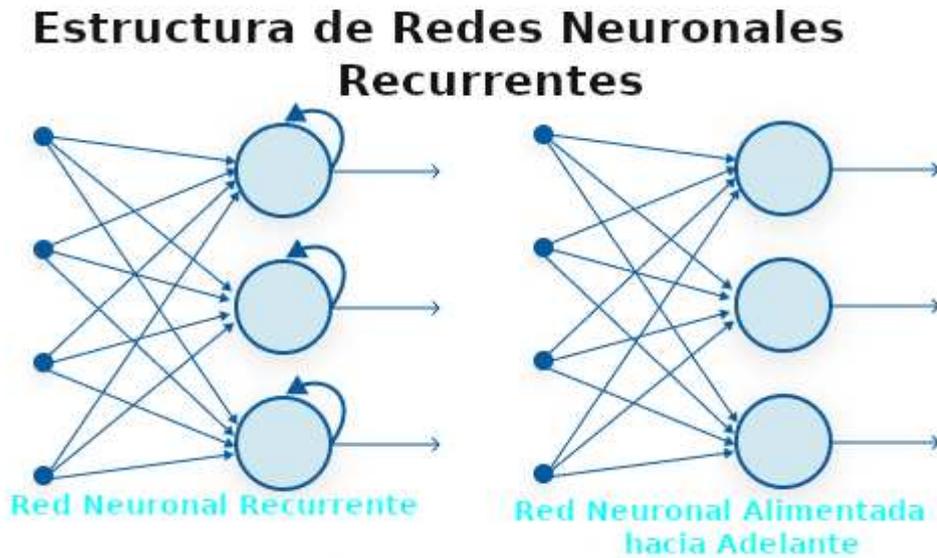
$$w_{li}(n+1) = w_{li}(n) - \eta \frac{\partial \varepsilon(n)}{\partial w_{li}(n)} \quad (2.27)$$

Donde  $\eta$  es una variable llamada coeficiente de aprendizaje (*learning rate*), la cual es ajustada, usualmente, entre un valor mayor que cero pero menor que uno. El signo menos en la ecuación 2.27 es para asegurarse de que en cada iteración del algoritmo la función de pérdida se estará acercando paulatinamente a un mínimo local o global (según la función de pérdida que se elija). El coeficiente de aprendizaje dicta con qué rapidez se acercará a dicho valor mínimo. Idealmente, el objetivo es acercarse a un mínimo global de la función de pérdida, es por eso que la función de pérdida debe seleccionarse de tal forma que sea una función continua, derivable y que además tenga un único mínimo. La función de error cuadrático medio cumple con las condiciones anteriores.

### 2.4.9. Redes Neuronales Recurrentes Simples

Las redes neuronales recurrentes son algoritmos pertenecientes a la familia del Aprendizaje Profundo que mejor resultados proporcionan en problemas que involucren secuencias de datos. Es un algoritmo diseñado para ‘recordar’ sus entradas anteriores por medio de la realimentación de datos pasados junto con datos actuales y esto se logra gracias a que la estructura de las neuronas recurrentes cuentan con una ‘memoria’ (*hidden state*) (ver figura 2.14) que retiene información de los estados inmediatamente anteriores. La memoria interna con la que cuentan les permite ‘recordar’ características o rasgos importantes en las secuencias de datos con las cuales se entrenan y es por eso que las redes neuronales recurrentes se han vuelto los algoritmos preferidos para problemas que involucren secuencias de datos tales como: series de tiempo, problemas relativos al lenguaje hablado y escrito, datos financieros, audio, video, etc. Estos algoritmos están detrás de grandes avances tecnológicos, entre ellos está el reconocimiento de voz, traducción de un idioma a otro, traducción de lenguaje hablado a lenguaje escrito, *chatbots*, y otros más.

Las redes neuronales recurrentes fueron creadas para superar las deficiencias que presentan las redes neuronales artificiales alimentadas hacia adelante, ya que estas redes neuronales no consideran entradas pasadas, no tienen forma de hacerlo, y por tanto, no son aptas para secuencias de datos. Las redes neuronales recurrentes fueron concebidas en la década de 1980 [12], pero solo hasta tiempos recientes donde el incremento de las bases de datos digitales, de la capacidad computacional, junto con la concepción de nuevos algoritmos han potenciado el uso y éxito de las redes neuronales recurrentes [5]. Entre los nuevos algoritmos que han enriquecido el desempeño de las redes neuronales recurrentes se encuentran: *Propagación hacia atrás en el tiempo* (*Back-Propagation Through Time*), *Redes neuronales recurrentes LSTM* [8] y *Redes neuronales recurrentes GRU* [31].



**Figura 2.14:** Comparación de la estructura de una red neuronal recurrente y una alimentada hacia adelante.

Observando la figura 2.15 en la página 38, la nomenclatura de las variables contenidas en la figura significan lo siguiente:

- El vector  $\vec{x}$  que consta de  $t+1$  elementos representa la secuencia de entrada a la red neuronal recurrente y  $\mathbf{U}$  su correspondiente matriz de pesos sinápticos.
- El vector  $\vec{o}$  representa la salida de la red neuronal recurrente y  $\mathbf{V}$  su correspondiente matriz de pesos sinápticos.
- El vector  $\vec{s}$  representa la memoria interna (*hidden state*) de la red neuronal recurrente y  $\mathbf{W}$  su correspondiente matriz de pesos sinápticos.

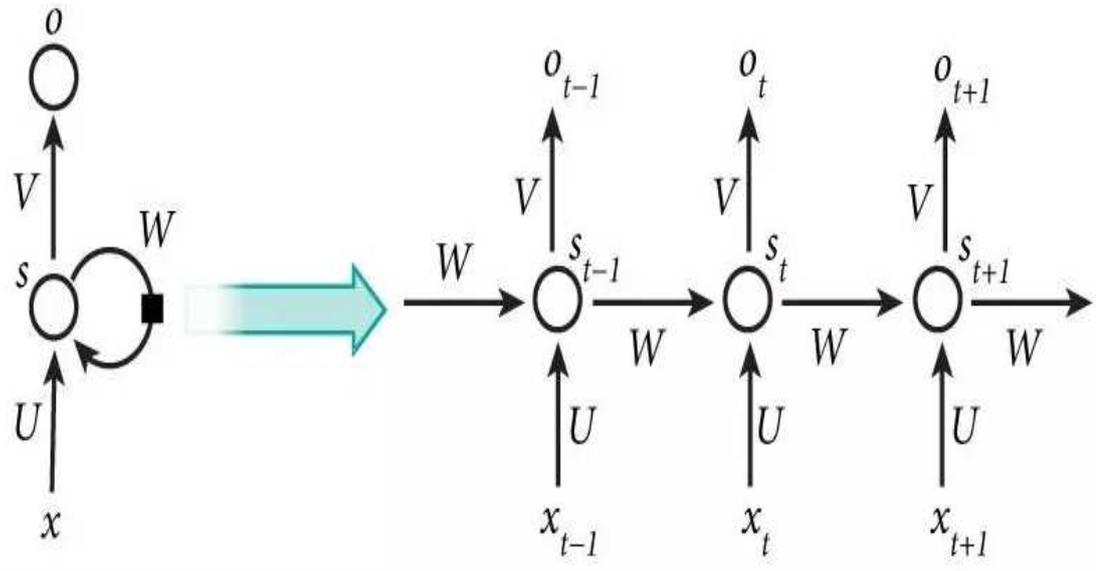
Internamente, con cada elemento  $t$  de los vectores  $\vec{x}$  y  $\vec{s}$  sucede lo siguiente:

$$s_t = f(Ux_t + Ws_{t-1}) \quad (2.28)$$

$$o_t = f(Vs_t) \quad (2.29)$$

Donde  $f$  es la función de activación de la red neuronal recurrente.

Se puede apreciar claramente como la ‘memoria’ de la red neuronal recurrente depende de la entrada actual y la entrada inmediatamente anterior  $t-1$ , y la información de la ‘memoria’ de la red se va actualizando y almacenando en la matriz de



**Figura 2.15:** Red neuronal recurrente extendida en el tiempo. Imagen del usuario MingxianLin de Wikipedia (CC BY-SA 4.0), recuperada de <https://commons.wikimedia.org/wiki/File:RNN.png>

pesos sinápticos  $W$  y en el vector  $\vec{s}$ . También se puede observar que el vector de salida  $\vec{o}$ , igualmente, está influenciado por el valor actual del vector  $\vec{x}$  y del vector  $\vec{s}$ .

Si bien estas redes neuronales recurrentes simples han probado ser eficientes en muchas áreas, no son aptas para secuencias de datos que presentan dependencias de tiempo a largo plazo puesto que estas dependencias terminan por desvanecerse poco a poco o ser olvidadas por completo. Este comportamiento es conocido como: *Desvanecimiento del Gradiente* (*Vanishing Gradient*) [34]. El problema del desvanecimiento de gradiente se debe a la longitud de la secuencia de datos que se está analizando y a la longitud de los vectores de entrada o fragmentos de la secuencia de datos con los cuales se está alimentando a la red neuronal recurrente, este problema se presenta al aplicar el algoritmo de propagación hacia atrás a través del tiempo, y lo que sucede es que en las capas más profundas de la red neuronal recurrente ya no se actualizan los pesos sinápticos de la misma forma que en las capas más superficiales provocando que el aprendizaje de la red neuronal recurrente se vea seriamente afectado. Para superar esta dificultad llegaron las redes neuronales recurrentes LSTM (*Long Short Term Memory*) las cuales resuelven este problema por medio de una arquitectura interna más elaborada y robusta.

### 2.4.10. *Long Short-Term Memory (LSTM)*

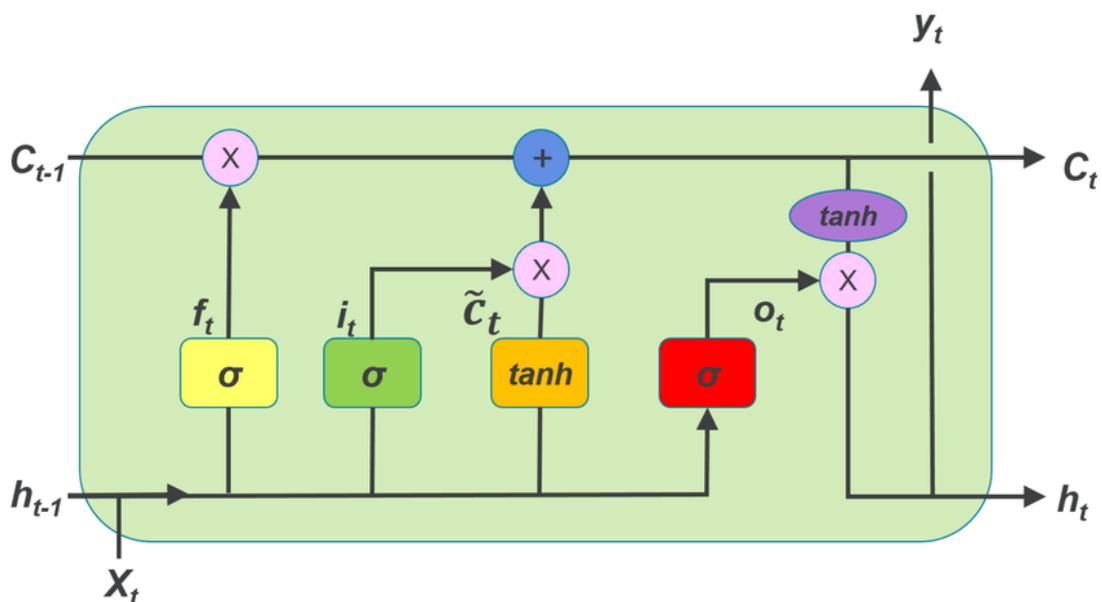
Existen múltiples mejoras de las redes neuronales recurrentes simples diseñadas para resolver el problema de desvanecimiento del gradiente, entre esas variantes están las redes neuronales recurrentes LSTM. Asimismo, existen diferentes variantes entre las mismas redes neuronales recurrentes LSTM, variantes que intentan simplificar o mejorar aspectos muy concretos en la arquitectura de una red neuronal LSTM estándar [8], en la figura 2.16 de la página 40 se tiene una de las arquitecturas más conocidas de las redes LSTM. Las redes neuronales LSTM son actualmente las más populares entre las redes neuronales recurrentes junto con las redes neuronales GRU.

Al igual que las redes recurrentes simples, las redes LSTM cuentan con conexiones recurrentes, una de ellas conocida como *memoria de trabajo* (*working memory* o *hidden state*) y otra de ellas es el *estado de la célula* (*cell state*). El estado de la célula es un agregado con el cual no cuenta el modelo de las redes recurrentes simples, y este nuevo agregado es el elemento que permitirá que la red neuronal recurrente LSTM cuente con una memoria a largo plazo. La memoria de trabajo juega el rol de ser la memoria a corto plazo. En el estado de la célula se puede agregar o remover información de largo plazo según se considere necesario por medio de compuertas (*gates*), este concepto se definirá en los siguientes párrafos.

En la figura 2.16 se puede observar las siguientes variables:

- $c_{t-1}$  es la célula de estado en el tiempo  $t - 1$ .
- $h_{t-1}$  es la memoria de trabajo en el tiempo  $t - 1$ .
- $x_t$  es la entrada en el tiempo actual  $t$ .
- $c_t$  es la célula de estado en el tiempo actual  $t$ .
- $h_t$  es la memoria de trabajo en el tiempo actual  $t$ .

Los vectores  $h_{t-1}$ ,  $x_t$  y  $c_{t-1}$  son las señales de entrada a la red neuronal recurrente LSTM, y los vectores  $h_t$  y  $c_t$  son las señales de salida. De la figura 2.16 también se puede apreciar cuatro pequeños recuadros en color amarillo, verde, naranja y rojo que simbolizan pequeñas redes neuronales que toman como entradas la concatenación de los vectores  $h_{t-1}$  y  $x_t$  y aplican funciones de activación a la salida, en este caso: sigmoidea ( $\sigma$ ) y tangente hiperbólica ( $\tanh$ ). En la figura 2.16 también se observan los pequeños círculos en color rosa y azul que representan puntos donde se sumarán o multiplicarán las señales de entrada. Con las flechas en color negra se observa el flujo y distribución de las diferentes señales. Si después de una función de activación sigmoidea se tiene una multiplicación de vectores a esta estructura o



**Figura 2.16:** Arquitectura Estándar de la Red Neuronal Recurrente Long Short-Term Memory [11].

combinación se le llamará *compuerta (gate)*, de la figura se observa que se tienen tres compuertas, cada una de ellas tiene su respectivo nombre, de izquierda a derecha: compuerta de olvido (*forget gate*), compuerta de entrada (*input gate*) y compuerta de salida (*output gate*). Por medio de las compuertas se controla el flujo y actualización de la información de la célula de estado y de la señal de salida de la red neuronal recurrente LSTM.

La idea principal y de mayor valor detrás de una red neuronal recurrente LSTM es la inclusión de la célula de estado y su arreglo de compuertas e interacción con los vectores de entrada. El agregar o remover información de la célula de estado  $c_{t-1}$  está regulado por la compuerta de olvido y la compuerta de entrada, al finalizar la actuación de estas dos compuertas se tendrá la célula de estado  $c_t$ , la cual pasará a interactuar con la siguiente muestra del conjunto de entradas  $x_{t+1}$ . A diferencia de las redes neuronales recurrentes simples en las cuales en cada iteración se reescribe el contenido de su memoria, en la red neuronal recurrente LSTM se decide que olvidar y que actualizar en su memoria de estado por medio de las compuertas de olvido y de entrada.

A continuación se definirá las compuertas de olvido y la compuerta de entrada y se verá como es que interactúan entre ellas para obtener  $c_t$ .

La compuerta de olvido se define de la siguiente manera:

$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f\right) \quad (2.30)$$

Donde  $W_f$  es la matriz de pesos sinápticos correspondiente a la compuerta de olvido. La compuerta  $f_t$  entregará un valor entre cero y uno, el valor que entregue determinará que tanto de la señal  $c_{t-1}$  se olvidará o recordará. Si se tuviera como resultado cero en la compuerta de olvido, claramente se estaría olvidando totalmente la señal  $c_{t-1}$ , lo contrario sucedería si tuviéramos como resultado uno.

Ahora se definirá la compuerta de entrada:

$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] + b_i\right) \quad (2.31)$$

Donde  $W_i$  es la matriz de pesos sinápticos correspondiente a la compuerta de entrada. La compuerta de entrada al tener una función de activación sigmoidea entregará un valor entre cero y uno. Por otra parte, de la figura 2.16 se observa que hay una red neuronal con función de activación tangente hiperbólica (recuadro naranja) y a su señal de salida se le identifica con el símbolo  $\tilde{c}_t$ , esta señal representa la célula de estado temporal, qué tanto de esa información temporal se agregará a la célula de estado  $c_t$  dependerá de la compuerta de entrada.

$\tilde{c}_t$  se define como:

$$\tilde{c}_t = \tanh\left(W_c \cdot [h_{t-1}, x_t] + b_c\right) \quad (2.32)$$

Donde  $W_c$  es la matriz de pesos sinápticos de la célula de estado temporal  $\tilde{c}_t$ . Ahora ya se tienen las dos funciones que ayudarán a determinar en que medida se agregará nueva información a la célula de estado  $c_t$ . Se tiene la compuerta de entrada  $i_t$  que regula la cantidad de información que pasará de la célula de estado temporal  $\tilde{c}_t$ , se cuenta con la compuerta de olvido  $f_t$  y la célula de estado temporal  $\tilde{c}_t$ , entonces, es posible definir la fórmula de actualización de la célula de estado  $c_t$ :

$$c_t = f_t c_{t-1} + i_t \tilde{c}_t \quad (2.33)$$

La fórmula indica el procedimiento que se lleva a cabo internamente, se multiplica el valor pasado de la célula de estado por la compuerta de olvido para determinar que es lo que se debe de olvidar de la célula de estado pasada, posteriormente, por medio de la compuerta de entrada se regula cual será la nueva información que se agregará a la célula de estado a partir de la célula de estado temporal.

Finalmente, se necesita una expresión para determinar que es lo que la red neuronal recurrente LSTM entregará como salida. La salida de la red neuronal

recurrente LSTM,  $h_t$ , será una señal filtrada obtenida a partir de la célula de estado recién calculada  $c_t$ . Primeramente, se define la compuerta de salida  $o_t$ , la cual filtrará la señal de salida de la red neuronal LSTM, posteriormente, se hará pasar a la célula de estado a través de una función de activación tangente hiperbólica (recuadro morado), y como resultado se entregará a la salida la multiplicación de dichas señales descritas.

$$o_t = \sigma\left(W_o[h_{t-1}, x_t] + b_o\right) \quad (2.34)$$

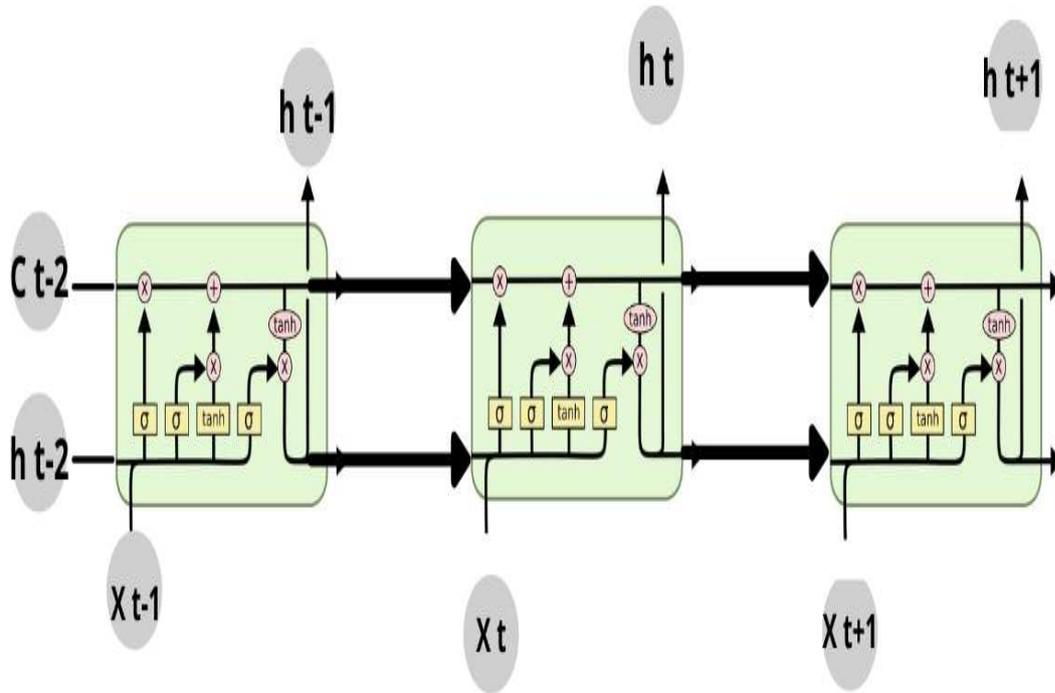
$$h_t = o_t \cdot \tanh(c_t) \quad (2.35)$$

Donde  $W_o$  es la matriz de pesos sinápticos de la compuerta de salida. Hasta este punto se han definido las variables y cómo es que se actualizan dichas variables en el interior de la estructura de una red neuronal recurrente LSTM. Las redes neuronales recurrentes LSTM a través del tiempo, al igual que las redes neuronales recurrentes simples, pueden ser visualizadas como múltiples copias de la misma estructura encadenadas unas tras otras (ver figura 2.17 en la página 43), donde cada estructura individual aceptará un valor del conjunto de entrada en su respectivo tiempo  $t_n$ .

### 2.4.11. *Gated Recurrent Unit (GRU)*

Las redes neuronales recurrentes GRU (*Gated Recurrent Unit*) son consideradas como una versión más simplificada de las redes neuronales LSTM [35], pero no por eso menos eficientes. Fueron presentadas en el año 2014 por Cho et. al. [31]. Las redes neuronales GRU al igual que las redes neuronales LSTM controlan el flujo de información por medio de compuertas. Las redes neuronales recurrentes GRU a diferencia de las redes neuronales recurrentes LSTM solo cuentan con dos compuertas: *compuerta de actualización (update gate)* y *compuerta de reinicio (reset gate)*, y tal como su nombre lo indica, la primera compuerta se encarga de filtrar qué tanta de la antigua información pasará al siguiente estado y la segunda compuerta se encarga de filtrar qué tanta información se eliminará. Las redes neuronales recurrentes GRU dejan de lado la variable que en redes neuronales recurrentes LSTM se conoce como ‘memoria de trabajo’, (*working memory* o *hidden state*) ocupando tan solo una unidad de memoria. A continuación se definen las variables de la figura 2.18 en la página 44.

- $h_{t-1}$  es la célula de estado (*‘hidden state’*) en el tiempo  $t - 1$ .
- $x_t$  es el vector de entrada actual en el tiempo  $t$ .
- $h_t$  es la célula de estado en el tiempo actual  $t$ .



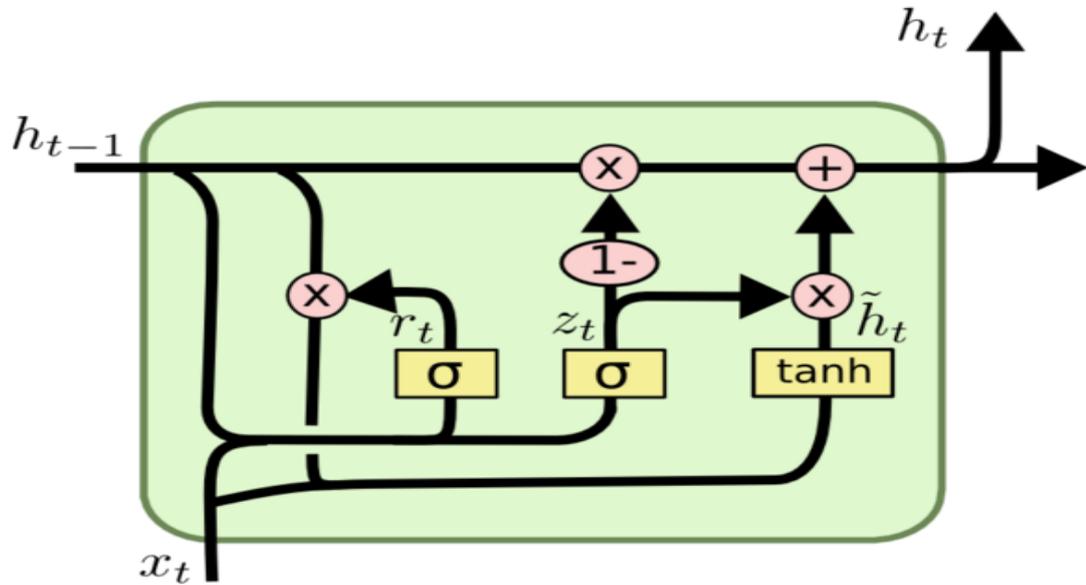
**Figura 2.17:** *Redes neuronales recurrentes LSTM extendidas en el tiempo.* Imagen del usuario MingxianLin de Wikipedia (CC BY-SA 4.0), recuperada de <https://commons.wikimedia.org/wiki/File:LSTM.jpg>

De la figura 2.18 se observa que se tienen dos vectores de entrada ( $h_{t-1}$  y  $x_t$ ) y uno a la salida ( $h_t$ ). También se observa la ubicación de las compuertas de actualización  $z_t$  y la compuerta de reinicio  $r_t$ , ambas determinadas por una función de activación sigmoidea. Igualmente, se puede apreciar la célula de estado temporal  $\tilde{h}_t$ . Con flechas en color negro se observa el flujo de los vectores de entrada y su paso por las pequeñas redes neuronales representadas en recuadros color amarillo, así como las estaciones de operaciones binarias representados con los círculos en color rosa.

Se define la compuerta de actualización:

$$z_t = \sigma(W_z x_t + U_z h_{t-1}) \quad (2.36)$$

Donde  $W_z$  y  $U_z$  son matrices de pesos sinápticos correspondientes a la compuerta de actualización.



**Figura 2.18:** Red neuronal recurrente GRU. Imagen del usuario MingxianLin de Wikipedia (CC BY-SA 4.0), recuperado de <https://commons.wikimedia.org/wiki/File:GRU.png>

Ahora, se define la compuerta de reinicio:

$$r_t = \sigma(W_r x_t + U_r h_{t-1}) \quad (2.37)$$

Donde  $W_r$  y  $U_r$  son matrices de pesos sinápticos correspondientes a la compuerta de reinicio.

Una vez que se han definido las dos compuertas es posible definir la célula de estado temporal  $\tilde{h}_t$  y la célula de estado actual  $h_t$ , primero se definirá la célula de estado temporal y después la actual:

$$\tilde{h}_t = \tanh(W x_t + r_t \cdot U h_{t-1}) \quad (2.38)$$

La fórmula anterior indica que la célula de estado temporal estará compuesta por el vector de entrada  $x_t$  y de la célula de estado anterior  $h_{t-1}$  que se multiplicará por la compuerta de reinicio para determinar que tanto de la información de la célula del estado anterior se removerá.

El contenido final de la información de la célula de estado está determinado por la siguiente fórmula:

$$h_t = z_t \cdot h_{t-1} + (1 - z_t) \cdot \tilde{h}_t \quad (2.39)$$

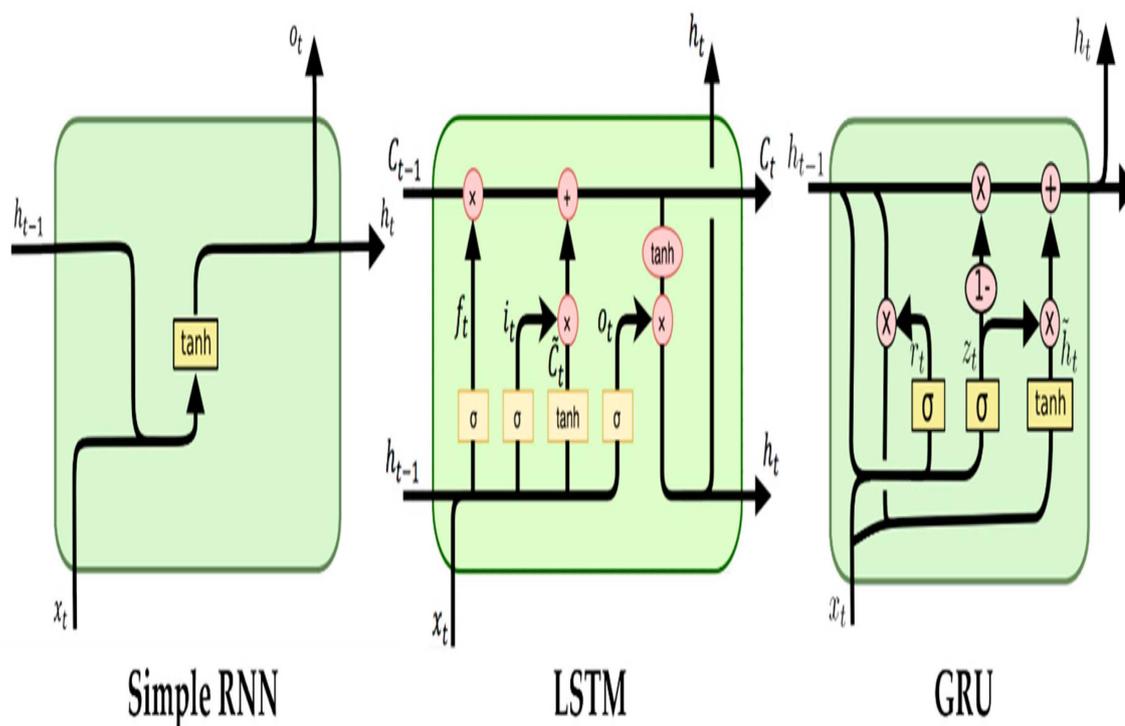
De la fórmula anterior se aprecia que la célula de estado pasada  $h_{t-1}$  y la célula de estado temporal  $\tilde{h}_t$  son multiplicadas por la compuerta de actualización para determinar que tanto de la información de cada célula de estado pasará a formar parte de la célula de estado actual  $h_t$ .

Hasta este punto se han podido observar las diferencias en arquitectura y de cómo interactúan las compuertas con sus respectivas entradas en las redes neuronales recurrentes LSTM y las GRU. Se ha visto que las redes neuronales recurrentes GRU, por decirlo de alguna forma, tienen una arquitectura más condensada, se deshacen de un vector de entrada en comparación con las redes neuronales recurrentes LSTM, y en la salida solo hay un vector. También usan una compuerta menos al actualizar y presentar su célula de estado, concretamente, no tienen la compuerta de salida que en las redes neuronales recurrentes LSTM regula la presentación que se dará del vector de salida de la red neuronal. Otra diferencia sumamente importante que contrasta a las redes neuronales recurrentes simples de las redes neuronales recurrentes LSTM y GRU es la actualización de su célula de estado, mientras que en el modelo de las redes recurrentes simples se reemplaza totalmente la información de una iteración a otra, en las redes neuronales recurrentes LSTM y GRU se recicla la información anterior por medio de las compuertas y se agrega nueva información, esta naturaleza aditiva en la célula de estado permite que los errores sean propagados hacia atrás en el tiempo sin que se desvanezcan los gradientes con facilidad [4].

Debido a que la red neuronal recurrente GRU tiene una arquitectura menos robusta que la red neuronal recurrente LSTM tiene un costo computacional menor en tiempo de entrenamiento. Aunque si hablamos de efectividad de una sobre la otra no hay una respuesta absoluta, unas se desempeñan mejor en cierto tipo de circunstancias que las otras y viceversa, lo que sí se puede afirmar es que las diferencias en desempeño no son muy amplias. Por tal motivo, en este trabajo de tesis se pondrán a prueba ambos modelos de redes neuronales recurrentes y se determinará qué modelo de red recurrente se ajusta mejor a los objetivos planteados.

En la figura 2.19 se pueden observar las diferencias que se mencionaban en el párrafo anterior en cuanto a la arquitectura de las redes neuronales recurrentes LSTM y GRU, también se observa la red neuronal recurrente simple y lo breve y sencilla de su arquitectura comparada con los otros dos modelos de redes neuronales recurrentes.

En la siguiente sección se abordará el tema de las redes neuronales convolucionales unidimensionales puesto que han probado tener gran desempeño en la predicción de valores contenidos en series de tiempo y en una sección posterior los



**Figura 2.19:** Arquitectura interna de las redes neuronales recurrentes simples, LSTM y GRU. Adaptado de “RNN, LSTM & GRU”, por dprogrammer, 2019 (<http://dprogrammer.org/rnn-lstm-gru>)

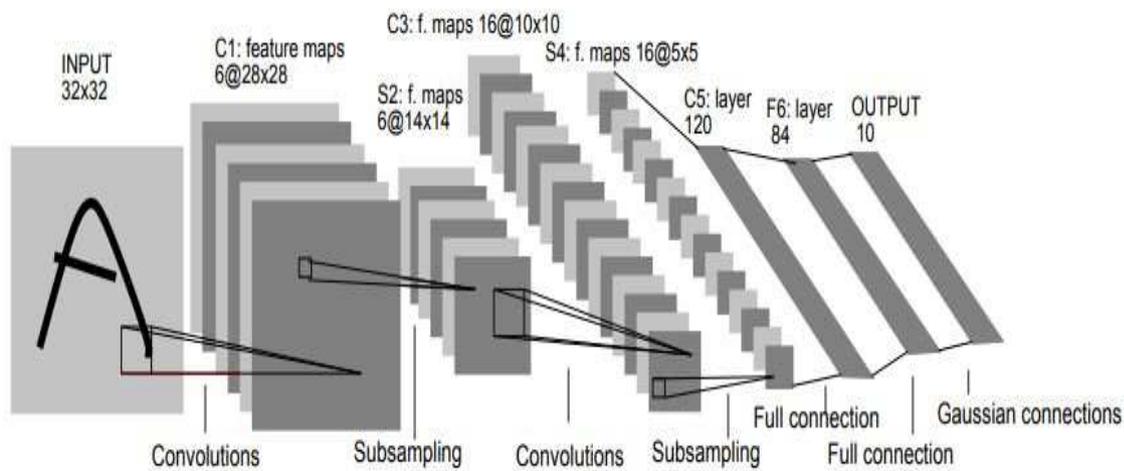
mecanismos de atención. La inclusión de mecanismos de atención han mejorado significativamente muchos algoritmos pertenecientes al área de NLP (*Natural Language Processing*) y CV (*Computer Vision*) hasta posicionarlos con la categoría de algoritmos a la vanguardia o *state of the art*. Se intentará incorporar mecanismos de atención a predicción, la idea será acoplar redes neuronales convolucionales unidimensionales, redes neuronales recurrentes, mecanismos de atención y redes neuronales densamente conectadas con el objetivo de obtener configuraciones de redes neuronales que entreguen predicciones aceptables para este trabajo de tesis.

#### 2.4.12. Redes neuronales convolucionales de una dimensión (*Conv1D*)

##### Introducción

En 1982, Fukushima y Miyake [7] presentaron el algoritmo que sería el predecesor de las redes neuronales convolucionales (*Convolutional Neural Networks*), conocido

como: *Neocognitron*. El neocognitron es una red neuronal que tiene la capacidad de reconocer patrones de estímulos basada en las diferencias de sus apariencias, es decir, en sus formas y contornos. Esta fue la primera red neuronal con la habilidad de clasificar correctamente objetos similares que solo variaban por sus posiciones o en algunas diferencias mínimas en su forma. Posteriormente, con el advenimiento del algoritmo de propagación hacia atrás [26] en 1986 fue posible que en el año 1990, LeCun *et al.* presentara en base al neocognitron y la propagación hacia atrás su red neuronal conocida como: *LeNet* [18], a la cual se le conoce como la primera red neuronal convolucional. La red neuronal convolucional propuesta en ese trabajo tuvo un gran éxito clasificando la base de datos del MNIST<sup>2</sup> (*Modified National Institute of Standards and Technology*) con lo cual se despertó nuevamente el interés en los algoritmos pertenecientes al Aprendizaje Profundo. La red *LeNet* (ver figura 2.20) constaba de dos etapas de convolución y dos de *Pooling* o *Down-Sampling* intercaladas una tras otra y tres capas densamente conectadas (dos capas escondidas y una capa de salida).



**Figura 2.20:** Arquitectura de la red neuronal convolucional conocida como ‘*LeNet*’ [19].

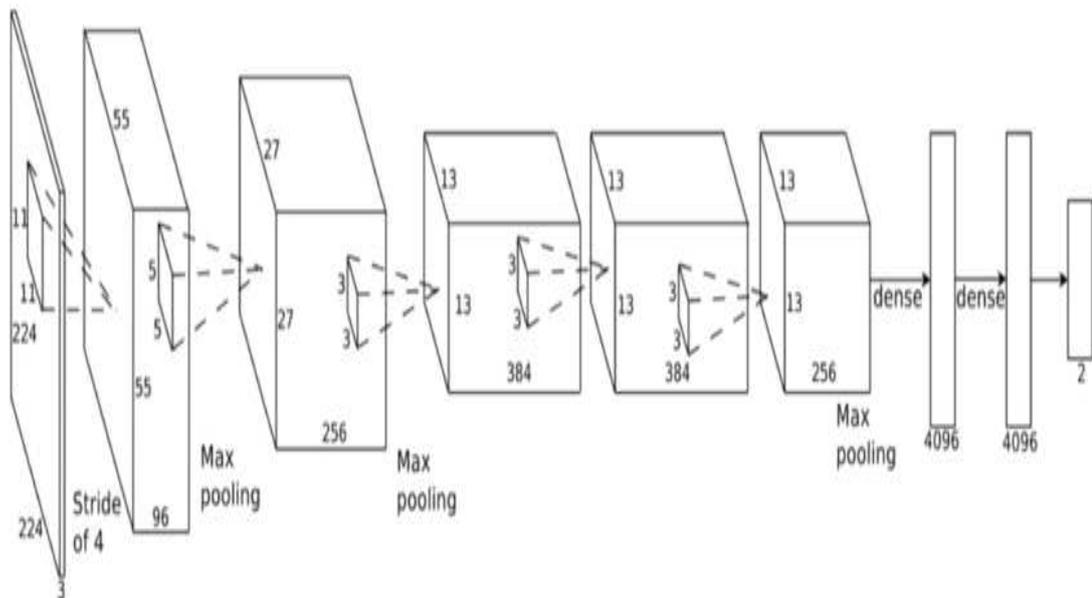
El siguiente gran avance en el desarrollo de las redes neuronales convolucionales sucedió en el año 2012 con la presentación de la red neuronal convolucional ‘profunda’ conocida como: *AlexNet* [17], la cual se puede considerar como una versión más profunda, amplia y mejorada de la red *LeNet*. En ese mismo trabajo también se presentó la función de activación *ReLU* (*Rectified Linear Unit*). El

<sup>2</sup>Imágenes de números en el rango de 0 a 9 escritos a mano de  $32 \times 32 \times 1$  píxeles.

## Capítulo 2. Marco teórico

mismo equipo de trabajo unos años antes de la presentación de *AlexNet* presentó un nuevo método para mitigar el sobre-entrenamiento en redes neuronales conocido como: *Dropout* [23]. *AlexNet* logró impresionantes resultados en la competición de clasificación de imágenes ILSVRC (*ImageNet Large Scale Visual Recognition Challenge*) en el año 2012, logrando tener tan solo un porcentaje de error de 16.4% clasificando imágenes pertenecientes a la base de datos *ImageNeT*,<sup>3</sup> consiguiendo superar por cerca de 10% al algoritmo más cercano basado en métodos tradicionales de Aprendizaje Máquina que obtuvo un porcentaje de error de 26.2%.

La red neuronal convolucional profunda *AlexNet* (ver figura 2.21) constaba de cinco capas convolucionales, tres capas *Pooling* o *Down-Sampling* y tres densamente conectadas (dos capas escondidas y una capa de salida). En los años posteriores a la presentación de *AlexNet* los algoritmos de redes neuronales convolucionales profundas se convirtieron en los algoritmos más usados para reconocimiento de imágenes, y desde ese momento los avances e innovaciones no han cesado, no solo en el ámbito de las imágenes sino que también se han usado las redes neuronales convoluciones en secuencias de datos.



**Figura 2.21:** Arquitectura de la red neuronal convolucional presentada por Alex Krizhevsky et al., conocida como ‘AlexNet’ [25].

<sup>3</sup>La base de datos *ImageNeT* contiene más de un millón de imágenes de  $224 \times 224 \times 3$  píxeles con mil categorías distintas.

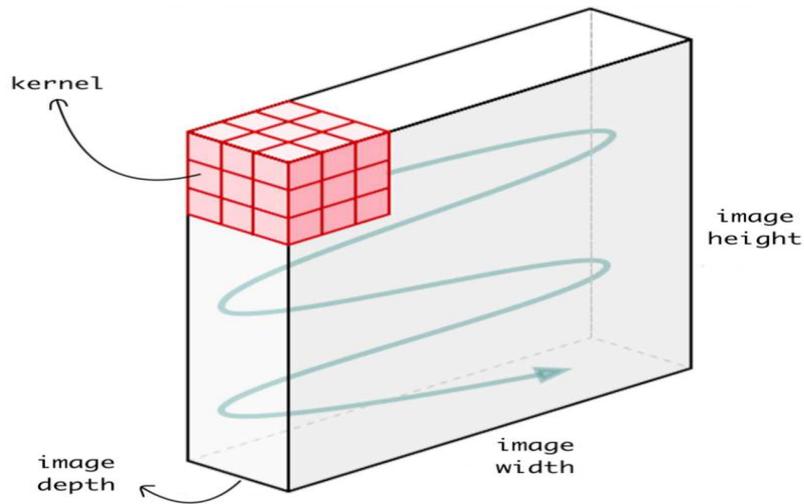
Las redes neuronales convolucionales fueron ideadas para reconocimiento o clasificación de imágenes por medio de la extracción de características de formas, colores y contornos; atributos únicos a determinados conjuntos de imágenes con mucho éxito. No obstante, no había redes neuronales convolucionales enfocadas totalmente a secuencias de datos unidimensionales; se hacían adaptaciones de secuencias de datos unidimensionales para poder presentarlos como si fuesen imágenes, aunque, si bien en ocasiones los resultados eran aceptables, en la mayoría de los casos los resultados no eran los deseados, así que se tuvo que idear un algoritmo de red neuronal convolucional que aceptara secuencias de datos unidimensionales. En el año 2015, Kiranyaz *et al.* [16] presentaron la primera red neuronal convolucional que aceptaba directamente datos unidimensionales. En muy poco tiempo las redes convoluciones unidimensionales han logrado popularidad y desempeño que está a la par de los resultados considerados como *estado del arte* (*State of the Art*) en procesamiento de señales unidimensionales.

### Redes neuronales convolucionales bidimensionales

Aunque han pasado más de 30 años desde que las redes neuronales convolucionales (RNC) se presentaran por primera vez, las arquitecturas modernas de RNC aún presentan las características más importantes que hicieron tan efectivas a las primeras RNC, tales como las capas convolucionales, las capas *pooling* o *down-sampling* y el algoritmo de actualización de parámetros en redes neuronales, propagación hacia atrás (*back-propagation*).

Entre algunas de las razones para que las RNC tuvieran el auge del que aún gozan se listan las siguientes:

- Las RNC pueden fusionar la extracción de características y la clasificación en un solo cuerpo integrado de red neuronal.
- Eficiencia computacional, ya que en la arquitectura de las RNC las neuronas no están densamente conectadas, lo que implica una reducción en la carga computacional.
- Las RNC son inmunes a pequeñas variaciones en los datos de entrada incluyendo desplazamientos, rotaciones, acercamientos y distorsiones.
- Las RNC se pueden adaptar a diferentes tamaños en las imágenes de entrada.

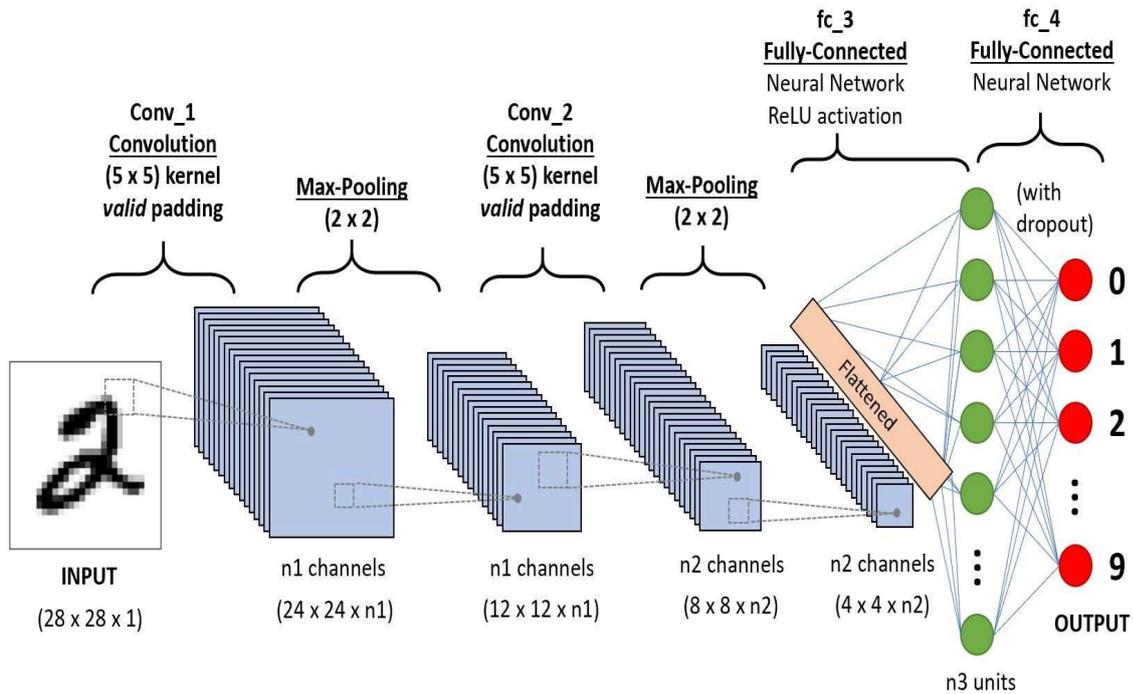


**Figura 2.22:** *Capa convolucional en una RNC.* Adaptado de *Kernel sliding over the Image* por Towards Data Science, 2019, recuperada de <https://towardsdatascience.com/understanding-1d-and-3d-convolution-neural-network-keras-9d8f76e29610>

En una red neuronal convencional (*Multiple Layer Perceptron*) en cada neurona los pesos sinápticos, entradas y salidas son de naturaleza escalar. Por el contrario, debido a que las entradas en una RNC son imágenes, las neuronas en una RNC cuentan con pesos sinápticos de naturaleza matricial, los cuales son conocidos como *kernel*, y las entradas y salidas son conocidas como mapa característico (*feature map*).

En la figura 2.22 se observa el proceso que se lleva a cabo en una capa convolucional y las variables que intervienen en el proceso. Primero, se observa que se está tratando con una imagen a color pues tenemos altura, amplitud y profundidad. Se observa el tamaño del *kernel* que es de  $3 \times 3 \times 3$ , y la flecha verde describe el movimiento de izquierda a derecha y de arriba abajo que se describe al realizar la *convolución*, abarcando cada uno de los datos contenidos en la imagen, como resultado de las capas convolucionales se obtienen los mapas característicos.

En la figura 2.23 se observa el efecto de las capas *pooling* o *down-sampling*, donde se especifica el factor de disminución que se tendrá en la altura y amplitud de los mapas característicos, en el ejemplo se tiene un factor de  $2 \times 2$ , lo cual implica que un *kernel* de  $2 \times 2$  realizará una convolución por los mapas característicos y en cada desplazamiento del *kernel* se tendrá como resultado un escalar, y los métodos más comunes para elegir el escalar que arrojará el *kernel* en cada desplazamiento



**Figura 2.23:** Ejemplo de RNC bidimensional. Adaptado de *A CNN sequence to classify handwritten digits* por Towards Data Science, 2018, recuperada de <https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53>

son: *Maximum Pooling* o *Average Pooling*. En *Maximum Pooling* se elegirá el valor máximo contenido en el *kernel* y en *Average Pooling* el resultado será el promedio de los valores contenidos en el *kernel*. Al pasar la imagen por las capas convolucionales y las capas *pooling* ya se han extraído las características más determinantes. Por último, se observan dos capas densamente conectadas, también conocidas como *capas probabilísticas*, ya que en estas capas es donde se realiza la clasificación de las imágenes.

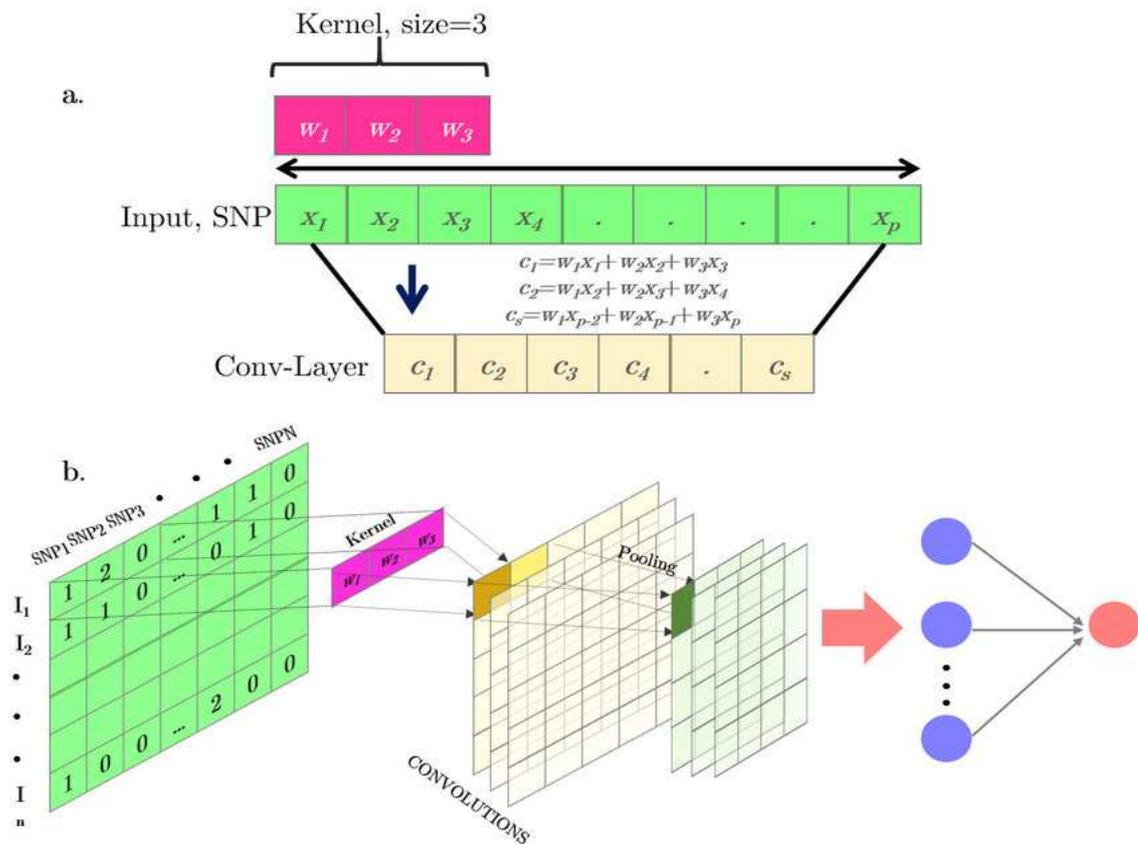
### Redes neuronales convolucionales unidimensionales

Con la introducción de las RNC unidimensionales se han observado las siguientes ventajas con respecto a sus contrapartes bidimensionales:

- La complejidad computacional de las RNC unidimensionales es menor que la de las RNC bidimensionales.

## Capítulo 2. Marco teórico

- Debido a sus bajos requerimientos computacionales las RNC unidimensionales tienen buen desempeño en aplicaciones en tiempo real.
- Comparadas con las RNC bidimensionales, las RNC unidimensionales entregan mejores resultados en problemas donde las bases de datos sean señales unidimensionales.



**Figura 2.24:** Ejemplo de RNC unidimensional [36]

La selección de hiperparámetros y principio de funcionamiento en las capas convolucionales y capas *pooling* para RNC unidimensionales son prácticamente las mismas que en las RNC bidimensionales, claro está, que en una RNC unidimensional el *kernel* se moverá en una sola dimensión. En la figura 2.24 se observa en color verde un arreglo de tamaño  $p$ , en color rosa observamos que el *kernel* es de tamaño  $3 \times 1$  y es claro que se mueve en una sola dimensión de izquierda a derecha. Como resultado de la capa convolucional se obtiene un elemento de un mapa característico que es el arreglo en color café y también observamos las

operaciones que se llevan a cabo entre el *kernel* y los respectivos segmentos del arreglo de entrada para poder obtener el mapa característico. En la figura 2.24 se observa la etapa de la capa *pooling* la cual tiene un *kernel* de tamaño  $2 \times 1$  que provoca una reducción en tamaño en los mapas característicos, y por último se observa la capa probabilística que consta de dos capas densamente conectadas.

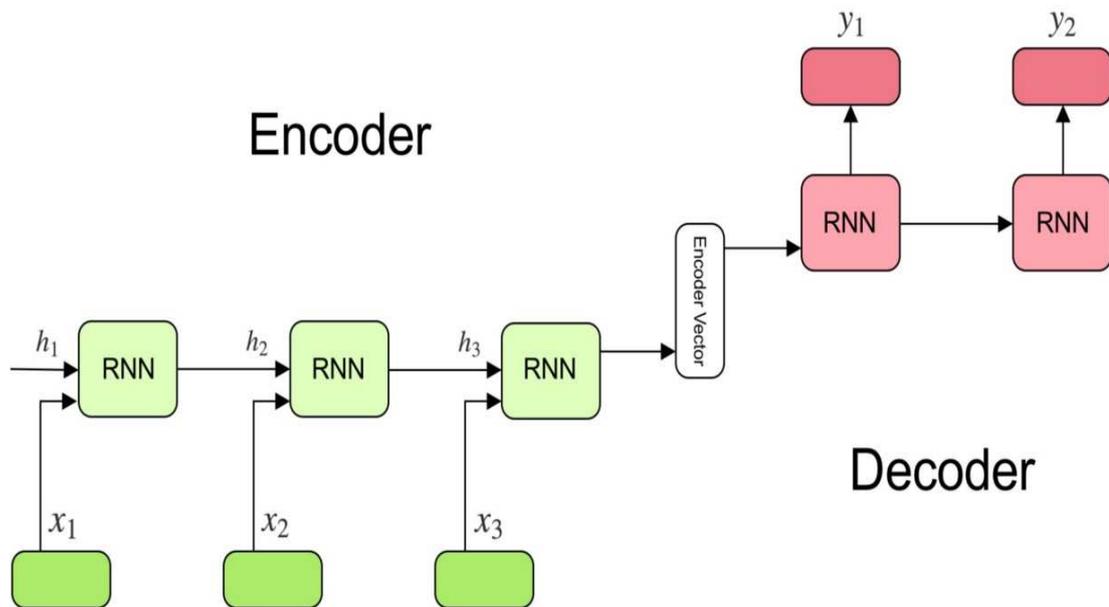
En ambas RNC, ya sea bidimensional o unidimensional, se ha mostrado que al final de la etapa de extracción de características (capas convolucionales y capas *pooling*) se agrega una capa probabilística, en el caso que se aborda en este trabajo de tesis, que es de predicción de lecturas contenidas en sismogramas, no se agregará una capa probabilística, el objetivo no es clasificación; inmediatamente después de las capas convolucionales se agregará una capa de redes neuronales recurrentes (o viceversa, redes recurrentes y después convolucionales) con el fin de dirigir los resultados a predicción. Detallando un poco más, se tomarán las características más distintivas extraídas por las capas convolucionales y se pasaran por redes neuronales recurrentes para que éstas capten las dependencias temporales que puedan existir en las características extraídas por las capas convolucionales, y por último, en la capa de salida habrá una capa densamente conectada de la cual obtendremos las predicciones deseadas. En base a lo explicado en este párrafo, y en general en esta sección, fue como se construyeron las redes *Conv1D-GRU* y *Conv1D-BiGRU* con las cuales se obtuvieron algunos de los resultados presentados en el próximo capítulo.

### 2.4.13. Mecanismos de Atención

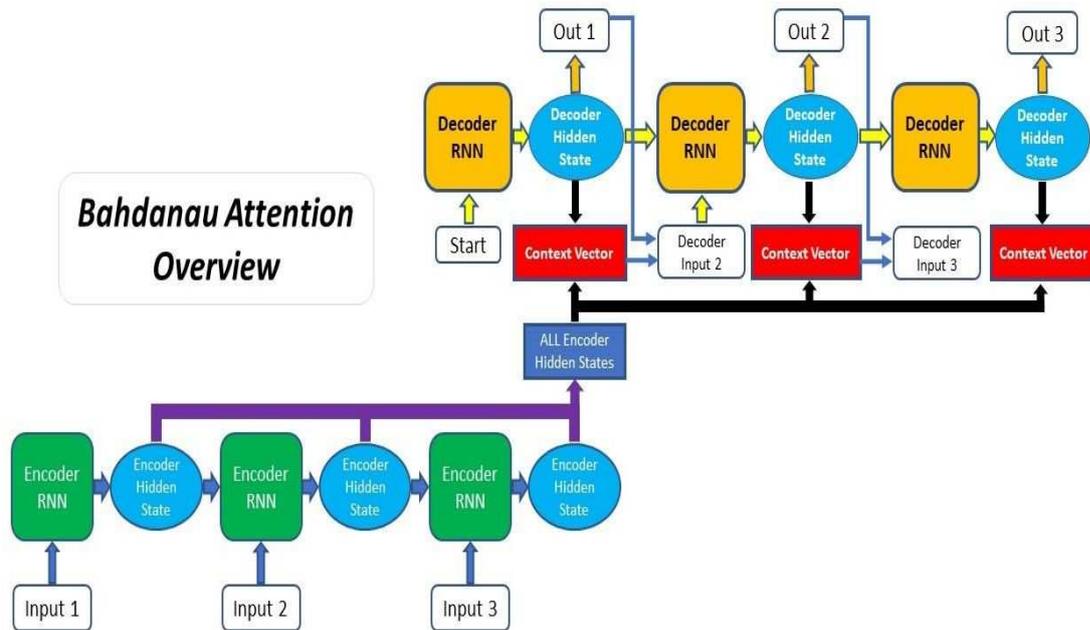
Los mecanismos de atención [1] surgieron en el año 2015 como un algoritmo de apoyo a las redes neuronales conocidas como: *Seq2Seq* [10]. Dichas redes *Seq2Seq* fueron presentadas en el año 2014. Las redes neuronales *Seq2Seq* son parte del área de estudio llamada Procesamiento Natural del Lenguaje (*Natural Language Processing*) la cual tiene como objetivo lograr que las computadoras o máquinas sean capaces de entender el lenguaje humano en cualquiera de sus modalidades posibles (hablado, escrito, señas, etc). Dentro del área del Procesamiento Natural del Lenguaje se encuentra una sub-área conocida como *Neural Machine Translation* (NMT). El modelo *Seq2Seq* pertenece al área NMT y su objetivo es traducir oraciones de un idioma a otro; esta es una de muchas aplicaciones más que se le ha dado a dicho modelo.

La arquitectura del modelo *Seq2Seq* está constituida por un codificador y un decodificador, ambos mediante redes neuronales recurrentes LSTM, en la figura 2.25 se puede apreciar como toda la información relevante de la oración de entrada en el codificador es almacenada en la última célula de estado (*hidden state*) en-

tregada por la red neuronal recurrente LSTM del codificador, esta característica del modelo Seq2Seq es el punto débil de la arquitectura, puesto que esta única célula de estado es de tamaño fijo una vez que se ha seleccionado la arquitectura de la red neuronal y, por tanto, estará limitada a poder almacenar cierta cantidad de información. Este fenómeno es conocido como *Information Bottleneck* ya que toda la información relevante de la oración de entrada no puede ser almacenada en un vector de longitud fija, especialmente cuando la longitud de la oración o texto de entrada es demasiado larga. La consecuencia de este fenómeno es que el decodificador no es capaz de traducir la oración adecuadamente puesto que carece de toda la información necesaria para realizar su tarea. Y es justamente por este motivo que surgieron los mecanismos de atención, los cuales no solo mejoraron drásticamente el desempeño de las redes neuronales Seq2Seq sino que han hecho camino en muchos otros campos del Aprendizaje Profundo con mucho éxito, a tal punto que surgieron modelos de redes neuronales que basan su funcionamiento enteramente en mecanismos de atención, desechando el uso de redes neuronales recurrentes. Tal es el caso de los famosos *Transformers* que fueron presentados en un artículo en el año 2017 titulado: *Attention is all you need* [32].



**Figura 2.25:** *Modelo Secuencia a Secuencia*. Adaptado de *Encoder-decoder sequence to sequence model* por Towards Data Science, 2019, recuperada de: <https://towardsdatascience.com/understanding-encoder-decoder-sequence-to-sequence-model-679e04af4346>



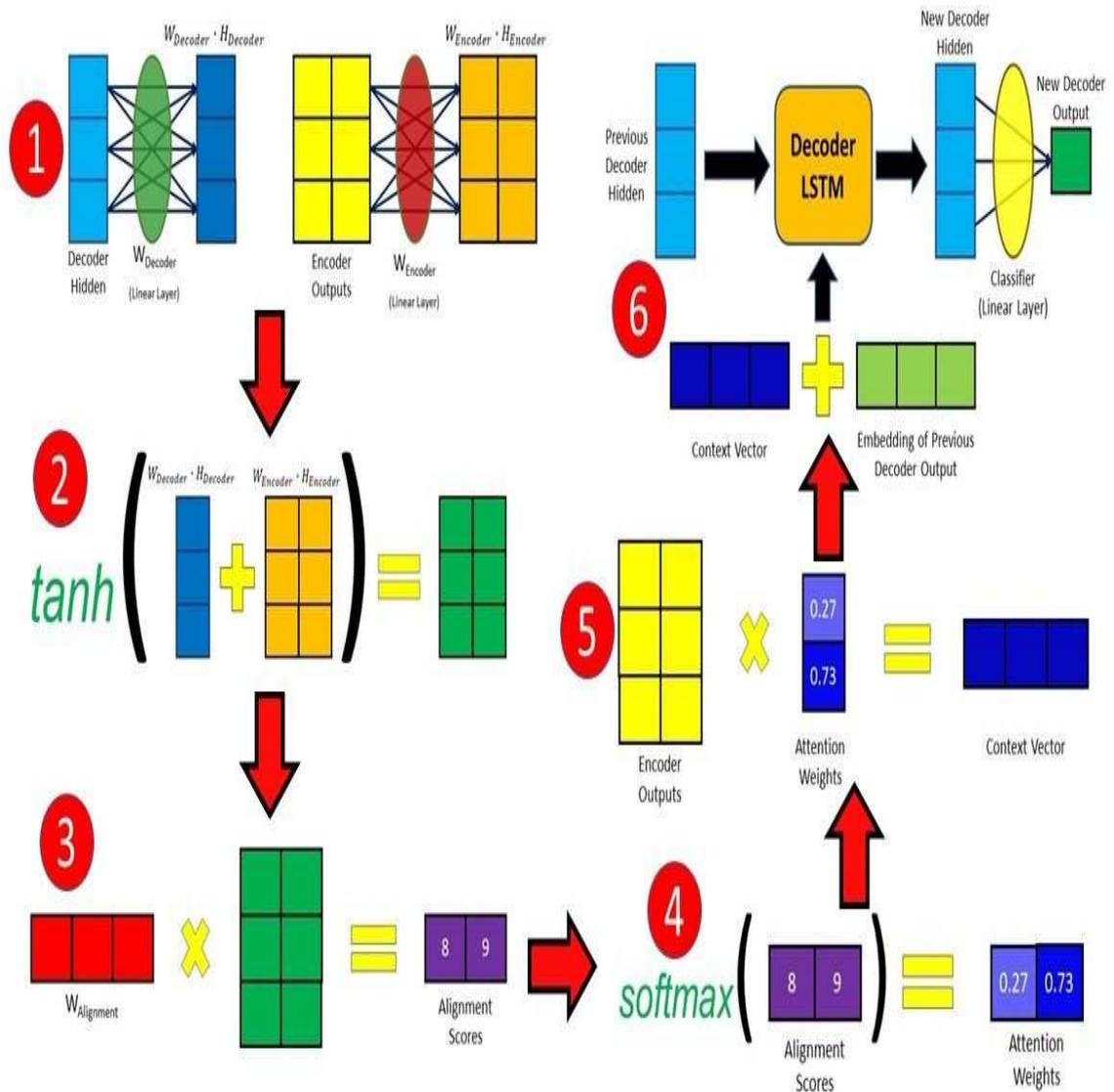
**Figura 2.26:** Mecanismo de Atención de Bahdanau. Adaptado de *Overall process for Bahdanau Attention seq2seq model* por FloydHub Blog, 2019, recuperada de: <https://blog.floydhub.com/attention-mechanism/#bahdanau-att-step2>

La idea fundamental de los mecanismos de atención es centrar los esfuerzos o recursos selectivamente en algunos elementos de la información de entrada, y a su vez ignorar aquellos elementos que se consideren irrelevantes para la tarea en cuestión. En la figura 2.26 se observa un panorama general de como es que se aplica el mecanismo de atención al modelo Seq2Seq. Se observa que ahora se utilizan todas las células de estado arrojadas por el codificador y a partir de ellas se forma un vector de contexto (*context vector*). Con el vector de contexto, la célula de estado en el tiempo  $t - 1$  y el vector de entrada se forma la nueva entrada al decodificador en el tiempo  $t$ .

El vector de contexto contiene información de qué entradas o palabras codificadas y representadas por medio de las células de estado son más relevantes para cada una de las entradas al decodificador, dicho en otras palabras, se le entrega información priorizada al decodificador de qué palabras o células de estado son más importantes para las palabras de entrada en el decodificador, de tal forma que el decodificador tenga más información relevante al momento de realizar la traducción. Estos valores priorizados asociados a cada uno de las células de estado para cada una de las entradas al decodificador reciben el nombre de *pesos de atención*. En la figura 2.27 podemos observar el proceso que se lleva a cabo para calcular los pesos de atención y posteriormente la entrada al decodificador.

## Capítulo 2. Marco teórico

La figura 2.27 en sí misma es bastante clara e ilustrativa en el proceso que se sigue al aplicar el mecanismo de atención de Bahdanau. Este proceso es el que se pretende aplicar a este trabajo de tesis, la idea es ocupar una red neuronal recurrente bidireccional y posteriormente aplicar un mecanismo de atención.



**Figura 2.27:** Desarrollo del mecanismo de atención. Adaptado de *Flow of calculating Attention weights in Bahdanau Attention* por FloydHub Blog, 2019, recuperada de: <https://blog.floydhub.com/attention-mechanism/#bahdanau-att-step2>

## Capítulo 3

---

# Redes neuronales recurrentes y convolucionales unidimensionales aplicadas a sismogramas para predicción

### 3.1. Introducción

En este capítulo se presenta el proceso de desarrollo que se llevó a cabo para realizar este trabajo de tesis. Las pruebas se realizaron con diferentes sismogramas recolectados de estaciones sísmicas alrededor del mundo y en años que van desde inicios de la década de 1990 hasta el año 2020 a través de la base de datos de *IRIS* y *SSN*. Como ya se había mencionado previamente, para realizar la lectura y manipulación de los archivos con extensión *.sac* se usará la librería *Obspy*<sup>1</sup> de Python y para la construcción de las redes neuronales se usará la librería *Keras*<sup>2</sup>.

Se explicará a detalle los pasos que se siguieron para la obtención de los resultados presentados en este capítulo. Se realizaron pruebas con diferentes configuraciones de redes recurrentes y con los modelos de redes neuronales recurrentes LSTM y GRU; las redes neuronales recurrentes simples fueron excluidas de las pruebas puesto que no eran capaces de arrojar resultados satisfactorios. También se agregaron redes neuronales convolucionales unidimensionales y mecanismos de atención a algunas de las configuraciones presentadas en esta sección. Se probaron diferentes configuraciones haciendo uso de combinaciones de redes neuronales recurrentes, redes convolucionales unidimensionales y mecanismos de atención. En general, los mejores resultados se obtuvieron usando el modelo de redes neuronales recurrentes GRU combinadas con mecanismos de atención.

---

<sup>1</sup><https://docs.obspy.org>

<sup>2</sup><https://keras.io>

## 3.2. Preparación de datos

Una vez que se seleccionó un evento sísmico, que se descargaron sismogramas de alguna de las dos bases de datos mencionadas anteriormente y que se logró obtener la serie de tiempo respectiva del sismograma se procedió a separar dicha serie de tiempo en tres subconjuntos: entrenamiento, validación y pruebas. Posteriormente, cada subconjunto es separado en subconjunto de entrada y subconjunto de salida; para formar los subconjuntos de entrada y salida se hace uso del algoritmo: *Sliding Window* [2]. Se forman subconjuntos de entrada y salida porque se está abordando un problema de aprendizaje supervisado enfocado a predicción. Posterior a formar los subconjuntos de entrada y salida se escalan los dos subconjuntos, de tal forma que los elementos de los subconjuntos quedan con valores contenidos en el intervalo  $[-1, 1]$ .

Es una práctica habitual en algoritmos pertenecientes a Aprendizaje Profundo normalizar o estandarizar los conjuntos de datos con los cuales se trabajará, no se recomienda tener algunos datos contenidos en un intervalo acotado por ciertos valores y algunos otros datos en intervalos con cotas más grandes o pequeñas. Se procura tener todos los datos con los cuales tendrá contacto la red neuronal en un intervalo único, ésto con el fin de evitar problemas en el algoritmo de propagación hacia atrás pues es posible que se presente el desvanecimiento de gradiente ó la explosión de gradiente.

### 3.2.1. Ventana deslizante (*Sliding Window*) y Normalización de datos

Para formar los subconjuntos de entrada y salida de los subconjuntos de entrenamiento, validación y pruebas se usa el algoritmo de la ventana deslizante, el procedimiento es el siguiente:

- Se elige la longitud de la ventana de entrada ( $i$ ) y de la ventana de salida ( $o$ ).
- Se desplazan las ventanas de entrada y salida (la primera atrás de la segunda) sobre el conjunto  $Q$  del cual se van a obtener los subconjuntos de entrada  $E$  y salida  $S$ , de tal forma que se seleccionan los primeros  $i$  elementos que abarca la longitud de la ventana de entrada y se agregan al subconjunto de entrada  $Q[0 : i] \subset E$ , posteriormente, se agregan los siguientes  $o$  elementos que cubre la ventana de salida al conjunto de salida a partir de la posición  $i$ , resultando en lo siguiente:  $Q[i : i + o] \subset S$ .

- En la siguiente iteración se desplazan las ventanas de entrada y salida un lugar hacia adelante y se repite el procedimiento descrito en el punto anterior, es decir,  $Q[1 : i + 1] \subset E$  y  $Q[i + 1 : i + 1 + o] \subset S$ .
- Se repite el paso anterior hasta cubrir la totalidad de los elementos del conjunto  $Q$ .

La selección de las variables  $i$  y  $o$  influirá en el diseño de la red neuronal recurrente puesto que para presentar los datos en la capa de entrada de la red neuronal recurrente se forman matrices con la forma:  $(i, f)$ , donde  $i$  también se conoce como *número de pasos en el tiempo* (*time-steps*) y  $f$  es el número de variables de la serie de tiempo (*features*). En el caso de las series de tiempo obtenidas de los sismogramas solo se tiene una variable que es la lectura del sensor usado para la recopilación de datos. La elección de los pasos en el tiempo es importante ya que una elección demasiado pequeña no le permitiría a la red neuronal recurrente captar el comportamiento de la serie de tiempo. La elección de  $o$  influirá en la predicción que arroje la red neuronal recurrente, ya que la red neuronal recurrente en su capa de salida dará una matriz con la forma:  $(o, f)$  como predicción; una elección muy grande de  $o$  limitaría la posibilidad de obtener buenos resultados pues se estaría forzando a la red neuronal recurrente a aprender una relación de mapeo entre entradas y salidas más complicada, con más variables que aprender o ajustar. En resumen, la elección de  $i$  y  $o$  es crucial para el desempeño de la red neuronal recurrente.

Finalmente, se normalizan los conjuntos de datos que tengan contacto con la red neuronal recurrente, la fórmula para normalizar es la siguiente:

$$a_i = \frac{a_i - \min(A)}{\max(A) - \min(A)} \quad (3.1)$$

Donde  $a_i$  es un elemento cualquiera del conjunto de datos  $A$ ,  $a_i \in A$ ,  $\max(A)$  es el elemento máximo del conjunto de datos  $A$  y  $\min(A)$  es el mínimo. Al realizar la normalización tendremos que  $a_i \in [-1, 1]$ ,  $\forall a_i \in A$ .

En este trabajo de tesis no solo se normalizarán las series de tiempo que se obtengan de los sismogramas, además, se asegurará de que posterior a la normalización se sume a cada elemento de la serie de tiempo el valor absoluto del mínimo valor encontrado en la serie de tiempo después de haber sido normalizada, esto con el objetivo de lograr que cada valor en la serie de tiempo sea no negativo. Se encontró experimentalmente que las configuraciones de redes neuronales recurrentes entrenadas para este trabajo de tesis entregaban mejores resultados cuando todos los valores de las series de tiempo eran no negativos.

## Capítulo 3. Redes neuronales recurrentes y convolucionales unidimensionales aplicadas a sismogramas para predicción

---

Para las series de tiempo que se obtuvieron de los sismogramas no se necesitó hacer limpieza de datos ya que las series de tiempo vienen sin lecturas faltantes y en el formato adecuado. Una vez que los datos han sido preparados o acondicionados se está en posición de definir la configuración, modelo de red neuronal recurrente y sus respectivos hiperparámetros.

### 3.3. Selección de las diferentes configuraciones de redes neuronales recurrentes

Se sabe que la mejor elección de red neuronal para un problema que involucre secuencias de datos o series de tiempo con dependencias temporales a largo plazo son las redes neuronales recurrentes, ya sea que se elija el modelo LSTM o GRU, según el problema que se esté atendiendo. Se hará uso de las diferentes configuraciones de redes neuronales recurrentes conocidas y en algunas de esas configuraciones se agregarán redes neuronales convolucionales unidimensionales y mecanismos de atención. Se probarán configuraciones de redes neuronales donde la primera capa sea una red neuronal convolucional unidimensional con el objetivo de extraer características diferenciadoras de las series de tiempo para posteriormente aplicar redes neuronales recurrentes que podrán captar las dependencias temporales de las características extraídas por las redes neuronales convolucionales unidimensionales, también se probarán configuraciones donde la primera capa sean redes neuronales recurrentes y posteriormente redes convolucionales unidimensionales ó mecanismos de atención.

Como primer acercamiento, y como resultados a vencer por configuraciones posteriores, se usará la configuración más simple de las redes neuronales recurrentes, la cual es conocida como: *Vanilla*, que es una simple red recurrente en solitario. Para cada configuración propuesta se usará los modelos de redes neuronales recurrentes LSTM y GRU con el objetivo de comparar el desempeño de ambos modelos. Las demás configuraciones que se usarán son las siguientes:

- *Stacked-RNN*.
- *BiRNN*.
- *Conv1D-RNN*.
- *Conv1D-BiRNN*.
- *BiRNN-Attention*.

Después de seleccionar la configuración de red neuronal recurrente se elegirán los diferentes hiperparámetros que podrían ser objeto de optimización, entre los más comunes, y solo por mencionar algunos, se tienen los siguientes:

- Número de capas escondidas en la red neuronal.
- Número de neuronas en las diferentes capas de la red neuronal.
- Función de Pérdida.
- Función de Optimización.
- Tipo de modelo de red neuronal recurrente (LSTM o GRU).
- Tamaño del Lote (*Batch Size*).
- Factor de aprendizaje (*Learning Rate*).

Hay que mencionar que se tienen muchos más hiperparámetros que pueden ser manipulados al construir una red neuronal en *Keras*, es difícil poder variar cada uno de los hiperparámetros por el tiempo que requeriría hacerlo, el *espacio de configuración*<sup>3</sup> sería muy extenso, por lo cual se suele seleccionar los hiperparámetros que se variarán en base a la experiencia y la intuición que se tenga en el uso del algoritmo en cuestión. Como bien se sabe, los algoritmos en Aprendizaje Profundo tienen una naturaleza heurística basada en lo empírico y para lograr que un algoritmo de Aprendizaje Profundo obtenga buenos resultados en gran medida se debe a la adecuada elección del modelo a usar, de los hiperparámetros y de la preparación de datos.

## 3.4. Algoritmos para la optimización de hiperparámetros

Para optimizar la selección de hiperparámetros se empleó en un principio el método de prueba y error (*Manual Search* o *Trial & Error*) como primera aproximación a los resultados que nos podría ofrecer la selección de las diferentes configuraciones de las redes neuronales recurrentes. Posteriormente, se abordaron los siguientes algoritmos de optimización de hiperparámetros más efectivos que el método de prueba y error:

- Búsqueda de malla (*Grid Search*).
- Búsqueda aleatoria (*Random Search*).

---

<sup>3</sup>Este concepto se define en los próximos párrafos.

## Capítulo 3. Redes neuronales recurrentes y convolucionales unidimensionales aplicadas a sismogramas para predicción

---

- Optimización Bayesiana (*Bayesian Optimization*).
- Algoritmos Genéticos (*Genetic Algorithms*)

Las librerías que se utilizaron para hacer optimización de hiperparámetros fueron: *skopt*<sup>4</sup>, *hyperopt*<sup>5</sup> y *geneticalgorithm*<sup>6</sup> de Python.

### Búsqueda manual

Este método es totalmente manual, tal como el nombre lo indica, la selección de qué hiperparámetros y cómo modificarlos se basa enteramente en la experiencia y la intuición del usuario. Este tipo de optimización de hiperparámetros es educacional y da cierto grado de información en lo que puede ó no puede funcionar, pero no es apto para trabajos profesionales donde el tiempo es un factor importante.

### Búsqueda de malla

En este método se hacen listas manuales ya sean de valores continuos o discretos de los hiperparámetros que se desean evaluar. El conjunto de las posibles combinaciones tomadas de las listas manuales es conocido como espacio de configuración. Contando con el espacio de configuración se automatiza un algoritmo para que evalúe cada una de las combinaciones posibles resultantes del espacio de configuración. Si el espacio de configuración tiene muchas dimensiones este método se torna ineficiente por el tiempo que requeriría completar todas las combinaciones.

### Búsqueda aleatoria

En este método se selecciona de antemano la cantidad de puntos del espacio de configuración que se evaluarán y del espacio de configuración se seleccionan aleatoriamente las combinaciones de hiperparámetros a evaluar. Este método suele superar en desempeño al método búsqueda de malla [13] (*Grid Search*), sobre todo cuando la dimensionalidad del espacio de configuración es grande, ya que es capaz de explorar con mayor libertad el espacio de configuración. Tanto la búsqueda de malla como la búsqueda aleatoria realizan corridas independientes, es decir, no guardan información de las corridas anteriores con la cual puedan guiar sus siguientes corridas hacia los puntos óptimos.

---

<sup>4</sup><https://scikit-optimize.github.io/stable/>

<sup>5</sup><http://hyperopt.github.io/hyperopt/>

<sup>6</sup><https://pypi.org/project/geneticalgorithm/>

### 3.5. Métricas para evaluar el desempeño de las redes neuronales recurrentes en la predicción de lecturas contenidas en sismogramas

---

#### Optimización Bayesiana

Este es un método de optimización global, está basado en un modelo probabilístico de la función de pérdida. El modelo evalúa la función de pérdida con valores del espacio de configuración que son prometedores y actualiza su información en base al resultado de la función de pérdida, en este método el objetivo es recopilar tanta información como sea posible de la función de pérdida y de la posible ubicación de los valores óptimos. El método está diseñado para explorar el espacio de configuración a la vez que intenta ubicar los valores óptimos para la función de pérdida [14, 15]. La optimización Bayesiana ha probado arrojar mejores resultados en menor cantidad de evaluaciones comparado con la búsqueda aleatoria (*Random Search*) y con la búsqueda de malla (*Grid Search*).

#### Algoritmos Genéticos

Los Algoritmos Genéticos (AG) son una técnica de búsqueda y optimización basada en los principios de la Genética y la Selección Natural. Esta técnica de optimización es un algoritmo perteneciente a una sub-área de la Computación Científica conocida como: *Computación Evolutiva*. Los AG se clasifican como una técnica de búsqueda y optimización aleatoria dirigida, fueron introducidos en el año 1975 por John Holland y algunos de sus estudiantes en su trabajo titulado: *Adaptation in natural and artificial systems* [9]. En los AG se tiene una población inicial (espacio de configuración) de posibles soluciones a un determinado problema. Estas posibles soluciones (o individuos) son sometidas a un proceso de recombinación y mutación, produciendo nuevos descendientes, y este proceso es llevado a cabo por varias generaciones. Cada individuo (candidato a posible solución) en la población es clasificado mediante su nivel de aptitud, y dependiendo de este nivel de aptitud se le asigna una probabilidad de dejar descendencia en la población venidera de tal forma que las generaciones futuras tendrán poblaciones más adaptadas o aptas para ser soluciones de un determinado problema. Los AG están diseñados para explorar el espacio de configuración y a su vez ubicar los valores óptimos del espacio explorado.

### 3.5. Métricas para evaluar el desempeño de las redes neuronales recurrentes en la predicción de lecturas contenidas en sismogramas

Las métricas que se usarán para evaluar el desempeño de las redes neuronales recurrentes en las series de tiempo de lecturas obtenidas de los sismogramas serán tres principalmente:

### Capítulo 3. Redes neuronales recurrentes y convolucionales unidimensionales aplicadas a sismogramas para predicción

---

- I) Porcentaje del promedio del valor absoluto de los errores en la totalidad de lecturas contenidas en una serie de tiempo, también conocido como *MAPE* (*Mean Absolute Percentage Error*) por sus siglas en inglés.
- II) Desviación estándar del porcentaje del promedio del valor absoluto de los errores en la totalidad de lecturas contenidas en una serie de tiempo (MAPE).
- III) Porcentaje del máximo error relativo de una lectura en la totalidad de las predicciones realizadas en la serie de tiempo.

La primera métrica permitirá evaluar el desempeño en general en las predicciones realizadas por la red neuronal recurrente en la totalidad de las lecturas de la serie de tiempo. La segunda métrica permitirá evaluar qué tan consistentes o qué tan volátiles son los errores en la totalidad de las lecturas de la serie de tiempo. Y la tercera métrica tan solo es para ver la magnitud del máximo error relativo de una predicción en la totalidad de las lecturas de la serie de tiempo. Obviamente, el objetivo es que las tres métricas sean tan pequeñas como sea posible, sobre todo las dos primeras que son las más representativas del desempeño de la red neuronal recurrente.

#### 3.6. Resultados obtenidos con la redes neuronales recurrentes

Para realizar una comparación lo más imparcial posible del desempeño de las diferentes configuraciones de redes neuronales recurrentes se eligió un evento sísmico y una estación sísmica, se obtuvo un sismograma en una sola componente espacial y su respectiva serie de tiempo con la cual se entrenó cada una de las diferentes configuraciones de redes neuronales recurrentes presentadas en este trabajo de tesis. Además, para evaluar el desempeño de las redes neuronales recurrentes después del entrenamiento se eligieron eventos sísmicos y estaciones sísmicas diferentes al evento sísmico con el cual se entrenaron las diferentes configuraciones de redes neuronales recurrentes, se extrajeron los sismogramas y sus respectivas series de tiempo y con dichas series de tiempo seleccionadas se evaluó por igual el desempeño de las diferentes configuraciones de redes neuronales recurrentes que se entrenaron.

La selección de hiperparámetros de las configuraciones de redes neuronales recurrentes fueron exactamente los mismos cuando lo que se deseaba era comparar el desempeño de los modelos de redes neuronales recurrentes LSTM y GRU.

Cuando se evaluaron los resultados de las diferentes configuraciones propuestas lo que se realizó fue optimizar tanto como fuera posible los hiperparámetros de

cada una de las configuraciones para posteriormente realizar la comparación entre configuraciones del desempeño de las mismas en las predicciones.

Se presentarán tablas y gráficas mostrando los resultados de las predicciones de las diferentes configuraciones de redes neuronales recurrentes que se entrenaron.

Como resultados base se utilizarán las métricas obtenidas por redes neuronales recurrentes tipo *Vanilla* usando los modelos de redes recurrentes LSTM y GRU, posteriormente se intentará superar estos resultados mediante la implementación de configuraciones más elaboradas de redes neuronales recurrentes que contarán con la ayuda de redes neuronales convolucionales unidimensionales y mecanismos de atención.

Todas las configuraciones de redes neuronales recurrentes que se presenten en este trabajo de tesis se entrenarán una única ocasión con el sismograma en la componente espacial Este-Oeste de la estación sísmica IGIG (ubicada en Irapuato, Guanajuato) para el sismo ocurrido el 19 de septiembre de 2017 en Puebla, México. El entrenamiento nunca se actualizará en ninguno de los resultados que se presenten a lo largo de este trabajo de tesis.

#### **3.6.1. Análisis de sismogramas en la componente espacial Este-Oeste para el terremoto del 19 de septiembre de 2017 en Puebla, México**

En la tabla 3.1 se muestran las métricas obtenidas por redes neuronales recurrentes Vanilla-LSTM y Vanilla-GRU. Ambas redes neuronales recurrentes comparten los mismos hiperparámetros. Los resultados son para el sismo del 19 de septiembre de 2017 en Puebla, México. Se presentan los resultados para siete diferentes estaciones sísmicas pertenecientes a la red de estaciones sísmicas del Servicio Sismológico Nacional (SSN).

En las figuras 3.1, 3.2 y 3.3 se observan las lecturas reales de los sismogramas y las lecturas predichas por la red Vanilla-GRU para las estaciones sísmicas AAIG, ZAIG y THIG. Tan solo se muestran las gráficas de estas tres estaciones sísmicas ya que las demás gráficas son muy parecidas en cuanto a que las predicciones hechas por las redes Vanilla-GRU y Vanilla-LSTM están muy próximas a las lecturas reales de los sismogramas. En cada una de las tres diferentes gráficas mostradas se realizó un total de 27 000 predicciones de lecturas por estación sísmica. En todas las gráficas presentadas el eje  $x$  corresponde al número de lectura en la serie de tiempo y el eje  $y$  corresponde al valor medido por el sismómetro en dicha lectura.

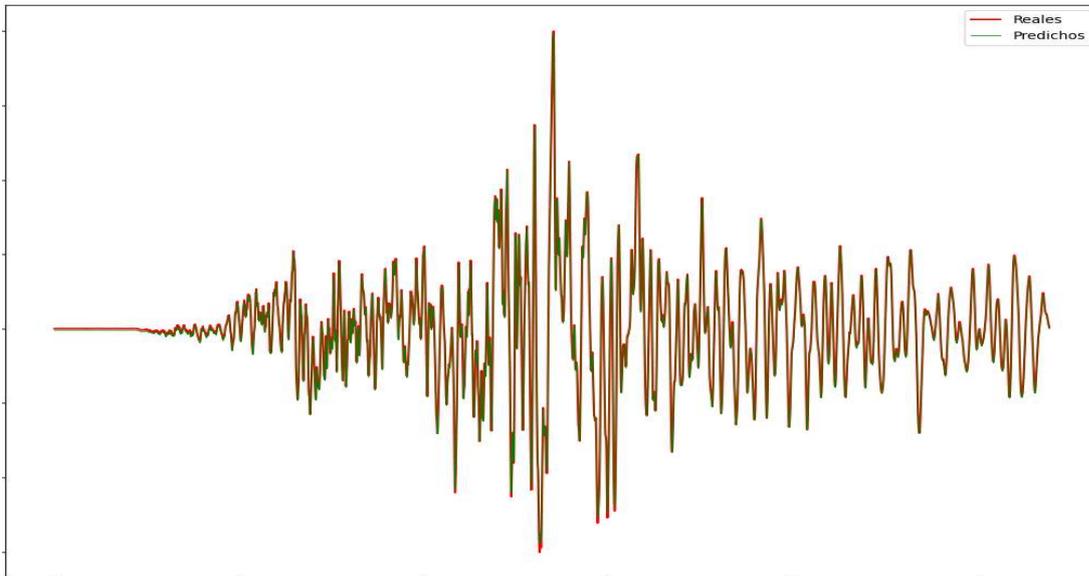
Capítulo 3. Redes neuronales recurrentes y convolucionales unidimensionales aplicadas a sismogramas para predicción

---

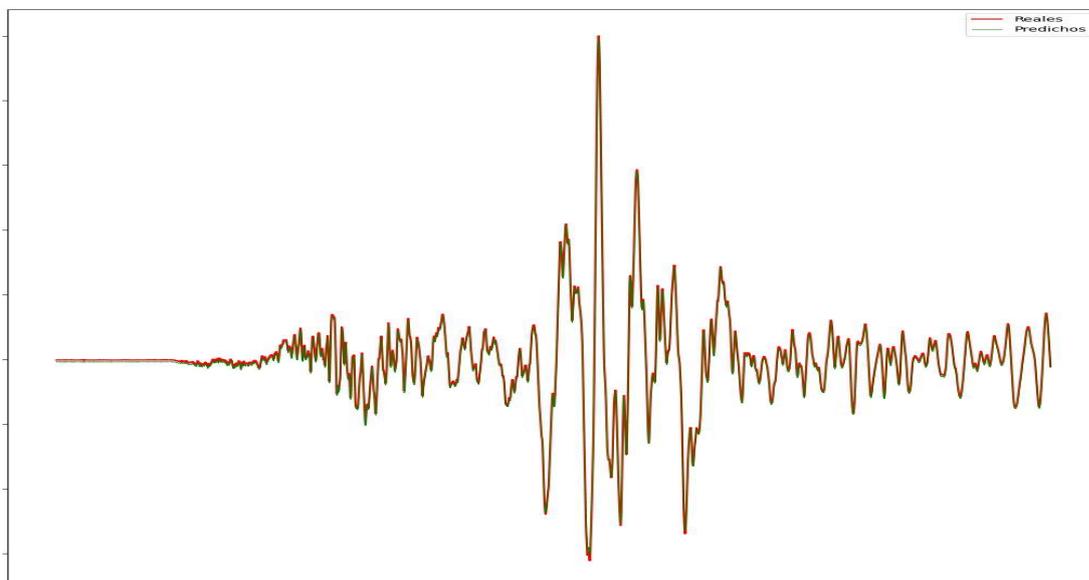
Estación Sísmica	Configuración y Modelo	Error Absoluto Promedio en la totalidad de las predicciones	Máximo Error Absoluto en una predicción	Desviación Estándar de los Errores Absolutos
AAIG	Vanilla LSTM	1.0794 %	15.0426 %	1.2461 %
	Vanilla GRU	<b>0.9158 %</b>	<b>14.2492 %</b>	<b>1.1293 %</b>
ZAIG	Vanilla LSTM	0.7852 %	12.6830 %	<b>0.9258 %</b>
	Vanilla GRU	<b>0.5733 %</b>	<b>12.4035 %</b>	0.9325 %
THIG	Vanilla LSTM	1.2747 %	23.6124 %	1.7540 %
	Vanilla GRU	<b>1.1208 %</b>	<b>21.5626 %</b>	<b>1.5502 %</b>
RPIG	Vanilla LSTM	0.7953 %	21.4459 %	1.1816 %
	Vanilla GRU	<b>0.7096 %</b>	<b>19.6051 %</b>	<b>1.0532 %</b>
ANIG	Vanilla LSTM	0.7777 %	13.7776 %	0.9100 %
	Vanilla GRU	<b>0.6842 %</b>	<b>12.9554 %</b>	<b>0.8500 %</b>
TJIG	Vanilla LSTM	<b>0.7884 %</b>	<b>17.4771 %</b>	<b>0.9291 %</b>
	Vanilla GRU	0.8188 %	18.3341 %	0.9641 %
MYIG	Vanilla LSTM	0.6423 %	21.9891 %	0.8110 %
	Vanilla GRU	<b>0.5982 %</b>	<b>20.0177 %</b>	<b>0.7173 %</b>

**Cuadro 3.1:** Comparación del desempeño de los modelos LSTM y GRU para el evento sísmico del 19 de septiembre de 2017 en Puebla, México. Se presentan los resultados de sismogramas de siete estaciones sísmicas ubicadas en la república mexicana, todas ellos en la componente espacial Este-Oeste y sus respectivas métricas. En letra negrita se resaltan los mejores resultados.

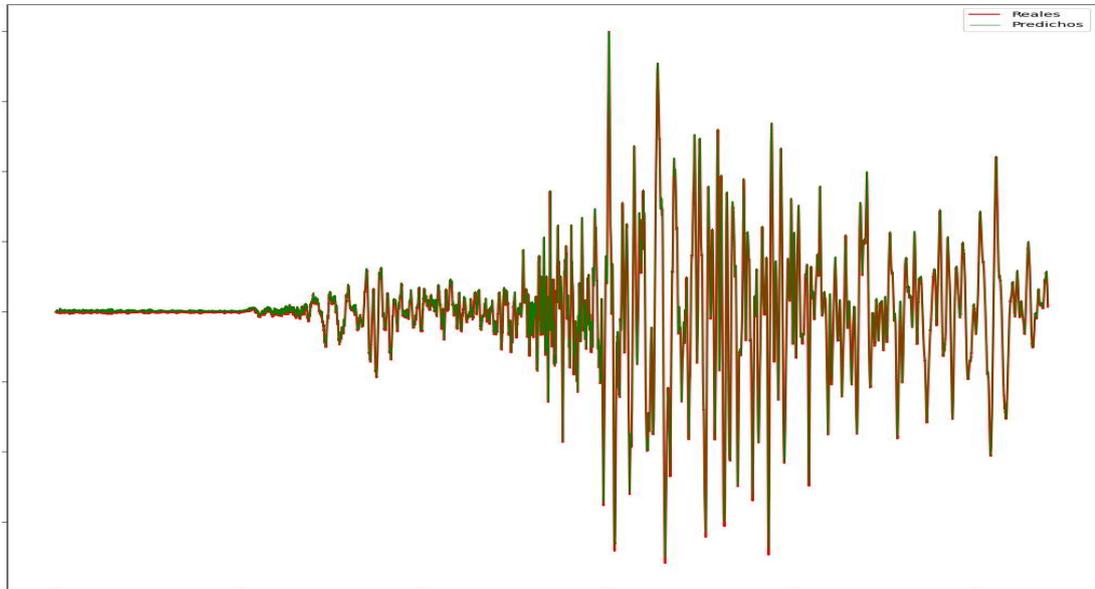
### 3.6. Resultados obtenidos con la redes neuronales recurrentes



**Figura 3.1:** Valores reales (color rojo) del sismograma de la estación AAIG (Aguascalientes, Aguascalientes) contra valores predichos (color verde) por la red neuronal Vanilla-GRU para el terremoto del 19 de septiembre de 2017 en Puebla, México.



**Figura 3.2:** Valores reales (color rojo) del sismograma de la estación ZAIG (Zacatecas, Zacatecas) contra valores predichos (color verde) por la red neuronal Vanilla-GRU para el terremoto del 19 de septiembre de 2017 en Puebla, México.



**Figura 3.3:** Valores reales (color rojo) del sismograma de la estación THIG (Tapachula, Chiapas) contra valores predichos (color verde) por la red neuronal Vanilla-GRU para el terremoto del 19 de septiembre de 2017 en Puebla, México.

### 3.6.2. Análisis de sismogramas en las componentes espaciales Norte-Sur y Vertical para el terremoto del 19 de septiembre de 2017 en Puebla, México

Se analizarán más sismogramas del mismo evento sísmico del 19 de septiembre de 2017 ocurrido en Puebla, México, pero, ahora se analizarán sismogramas en las componentes espaciales Norte-Sur y Vertical. Las redes neuronales recurrentes Vanilla-LSTM y Vanilla-GRU fueron entrenadas con un sismograma en la componente espacial Este-Oeste del mismo evento sísmico, así que se pondrá a prueba qué tanto afecta usar sismogramas en componentes espaciales distintas a la que se usó para entrenar las redes neuronales.

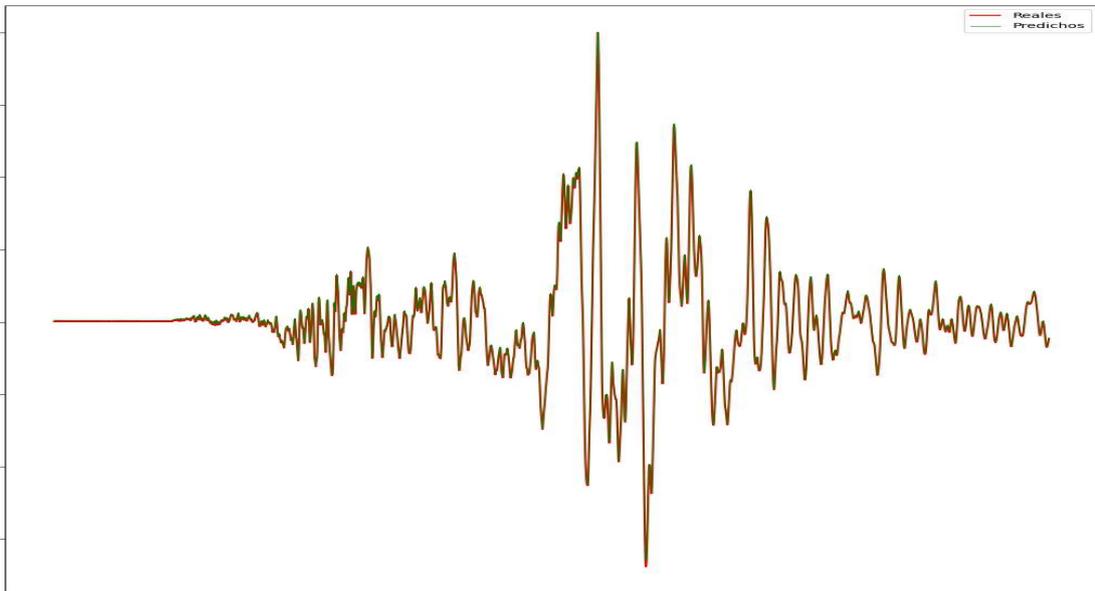
En la tabla 3.2 se observan las métricas obtenidas de analizar con las redes neuronales recurrentes Vanilla-LSTM y Vanilla-GRU ocho diferentes sismogramas de cuatro estaciones sísmicas ubicadas en territorio mexicano, cada una de ellas en las componentes espaciales Norte-Sur y Vertical.

En las figuras 3.4 y 3.5 se observan las lecturas reales de los sismogramas contra las lecturas predichas por la red neuronal recurrente Vanilla-GRU. Solo se presentan dos gráficos. En cada sismograma se realizaron 27 000 predicciones.

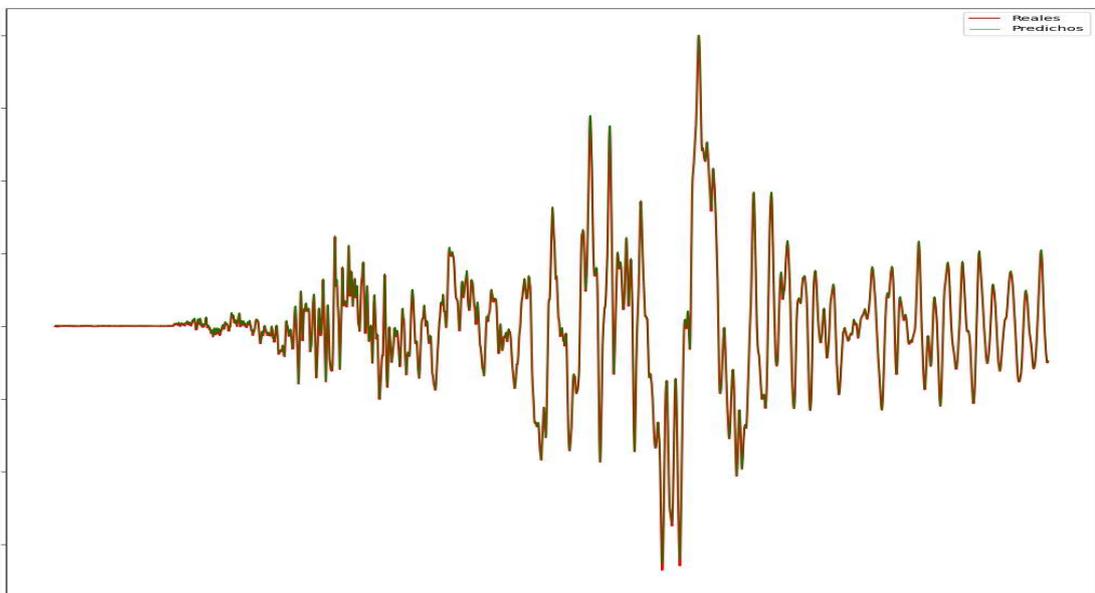
3.6. Resultados obtenidos con la redes neuronales recurrentes

Estación Sísmica	Configuración y Modelo	Error Absoluto Promedio en la totalidad de las predicciones	Máximo Error Absoluto en una predicción	Desviación Estándar de los Errores Absolutos
IGIG(Norte-Sur)	Vanilla LSTM	0.9817 %	19.5092 %	1.1936 %
	Vanilla GRU	<b>0.7725 %</b>	<b>17.9357 %</b>	<b>1.1204 %</b>
IGIG(Vertical)	Vanilla LSTM	1.1108 %	16.5302 %	1.3089 %
	Vanilla GRU	<b>0.9644 %</b>	<b>15.3870 %</b>	<b>1.1636 %</b>
AAIG(Norte-Sur)	Vanilla LSTM	0.8825 %	26.1478 %	1.1011 %
	Vanilla GRU	<b>0.8109 %</b>	<b>23.6118 %</b>	<b>0.9539 %</b>
AAIG(Vertical)	Vanilla LSTM	0.8275 %	15.0957 %	1.0111 %
	Vanilla GRU	<b>0.7442 %</b>	<b>14.0866 %</b>	<b>0.9236 %</b>
ZAIG(Norte-Sur)	Vanilla LSTM	0.6539 %	14.7377 %	0.7272 %
	Vanilla GRU	<b>0.5461 %</b>	<b>13.8762 %</b>	<b>0.6791 %</b>
ZAIG(Vertical)	Vanilla LSTM	0.7832 %	15.5429 %	0.9403 %
	Vanilla GRU	<b>0.6562 %</b>	<b>14.5042 %</b>	<b>0.8853 %</b>
THIG(Norte-Sur)	Vanilla LSTM	1.3837 %	25.5969 %	1.8585 %
	Vanilla GRU	<b>1.2298 %</b>	<b>23.0979 %</b>	<b>1.6294 %</b>
THIG(Vertical)	Vanilla LSTM	0.9724 %	25.9418 %	1.2318 %
	Vanilla GRU	<b>0.8719 %</b>	<b>23.5486 %</b>	<b>1.0817 %</b>

**Cuadro 3.2:** Desempeño de modelos LSTM y GRU para sismogramas en las componentes espaciales Norte-Sur y Vertical para el terremoto del 19 de septiembre en Puebla, México. Se presentan los resultados de sismogramas de cuatro estaciones sísmicas.



**Figura 3.4:** Sismograma de la estación ZAIG (Zacatecas, Zacatecas) componente Norte-Sur contra valores predichos (color verde) por la red neuronal Vanilla-GRU para el terremoto del 19 de septiembre de 2017 en Puebla, México.



**Figura 3.5:** Sismograma de la estación ZAIG (Zacatecas, Zacatecas) componente Vertical contra valores predichos (color verde) por la red neuronal Vanilla-GRU para el terremoto del 19 de septiembre de 2017 en Puebla, México.

Hasta este punto se ha observado que los resultados de las métricas no se vieron afectados al analizar sismogramas en componentes espaciales diferentes que con la cual fueron entrenadas las redes neuronales recurrentes. Las métricas obtenidas por las redes neuronales recurrentes son prácticamente tan buenas como lo son para sismogramas en la misma componente espacial que con la que fueron entrenadas las redes neuronales recurrentes.

#### **3.6.3. Análisis de sismogramas de eventos sísmicos ocurridos en México, pero, diferentes al que tomó lugar el 19 de septiembre de 2017 en Puebla, México**

En la sección anterior se observó que al analizar sismogramas en componentes espaciales diferentes a la que se utilizó para entrenar las redes neuronales no tuvo un impacto negativo en el desempeño de las redes neuronales recurrentes, por tanto, ahora se dará un paso más allá y se analizarán sismogramas de eventos sísmicos diferentes al sismo con el cual fueron entrenadas las redes neuronales recurrentes Vanilla-LSTM y Vanilla-GRU. Se observará si los resultados entregados por las redes neuronales recurrentes siguen siendo aceptables como lo fueron con sismogramas en las tres componentes espaciales pertenecientes al mismo evento sísmico con el cual fueron entrenadas.

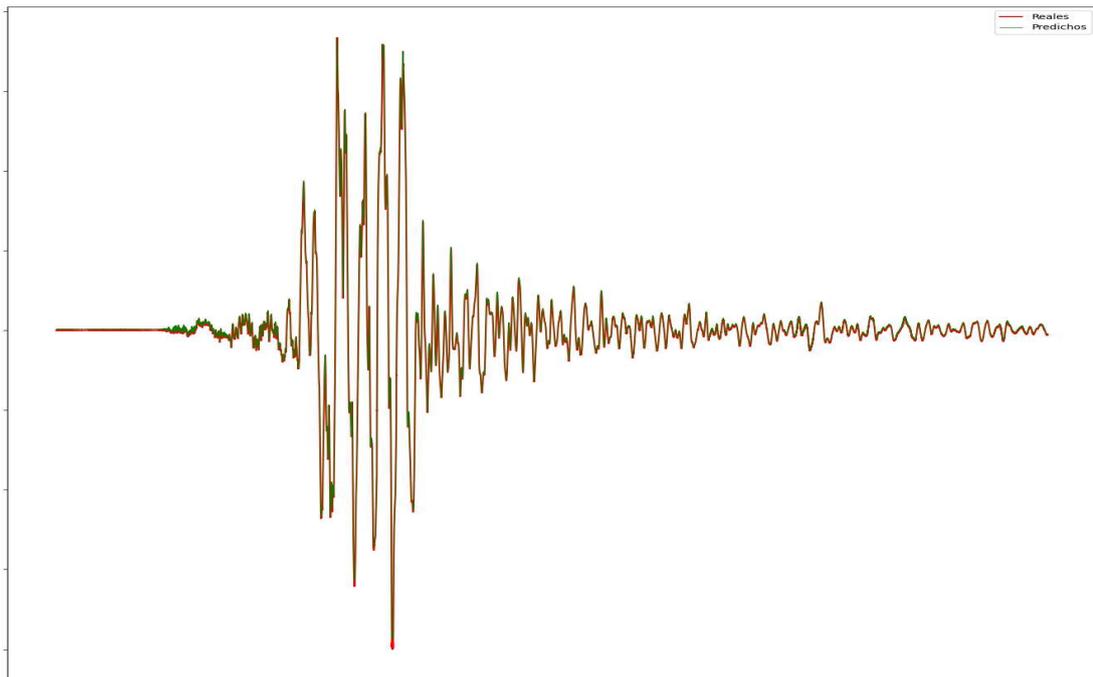
#### **Sismo en Chiapas el 8 de septiembre de 2017**

El 8 de septiembre de 2017 a las 04:49:17 (hora central) ocurrió un terremoto de magnitud 8.2, con epicentro 140 km al suroeste de Pijijiapan, Chiapas. En la tabla 3.3 se muestra el análisis de dos sismogramas en la componente espacial Este-Oeste con las redes neuronales Vanilla-LSTM y Vanilla-GRU. Los sismogramas de las estaciones sísmicas IGIG (Irapuato, Guanajuato) y ZIIG (Zacatecas, Zacatecas) contienen 90 000 lecturas cada una de ellas. En las figuras 3.6 y 3.7 se muestran las lecturas reales de los sismogramas contra las lecturas obtenidas con la red neuronal Vanilla-GRU.

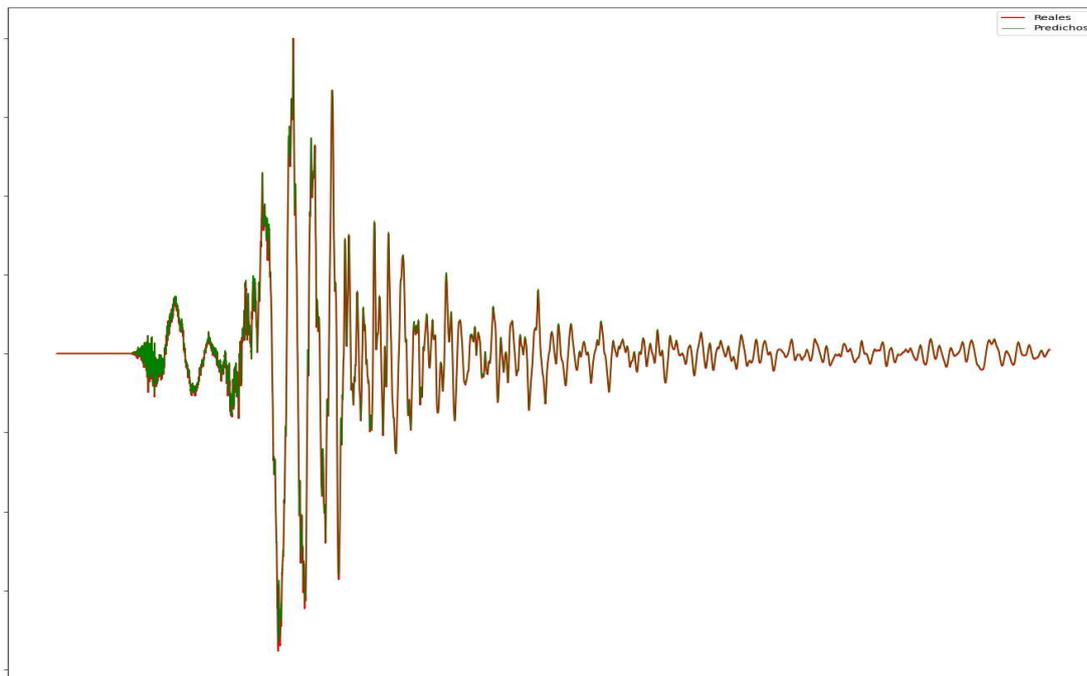
Capítulo 3. Redes neuronales recurrentes y convolucionales unidimensionales aplicadas a sismogramas para predicción

Estación Sísmica	Configuración y Modelo	Error Absoluto Promedio en la totalidad de las predicciones	Máximo Error Absoluto en una predicción	Desviación Estándar de los Errores Absolutos
IGIG	Vanilla LSTM	0.4493 %	24.6021 %	0.5163 %
	Vanilla GRU	<b>0.4485 %</b>	<b>23.0776 %</b>	<b>0.5021 %</b>
ZIIG	Vanilla LSTM	0.5417 %	13.3398 %	<b>0.6102 %</b>
	Vanilla GRU	<b>0.4969 %</b>	<b>12.5568 %</b>	0.6127 %

**Cuadro 3.3:** Desempeño de modelos LSTM y GRU para sismogramas del terremoto ocurrido el 8 de septiembre de 2017 en Pijijiapan, Chiapas. Se presentan los resultados de sismogramas de dos estaciones sísmicas.



**Figura 3.6:** Sismograma de la estación IGIG (Irapuato, Guanajuato) del terremoto del 8 de septiembre de 2017 en Pijijiapan, Chiapas, contra valores predichos (color verde) por la red neuronal Vanilla-GRU.



**Figura 3.7:** Sismograma de la estación ZIIG (Zacatecas, Zacatecas) del terremoto del 8 de septiembre de 2017 en Pijijiapan, Chiapas, contra valores predichos (color verde) por la red neuronal Vanilla-GRU.

Ahora se evaluará el desempeño de las redes neuronales recurrentes con otros dos eventos sísmicos ocurridos en territorio mexicano diferentes al del 8 y 19 de septiembre de 2017 en Puebla y Chiapas, respectivamente, y se observará si los resultados obtenidos por las redes neuronales recurrentes se siguen manteniendo estables.

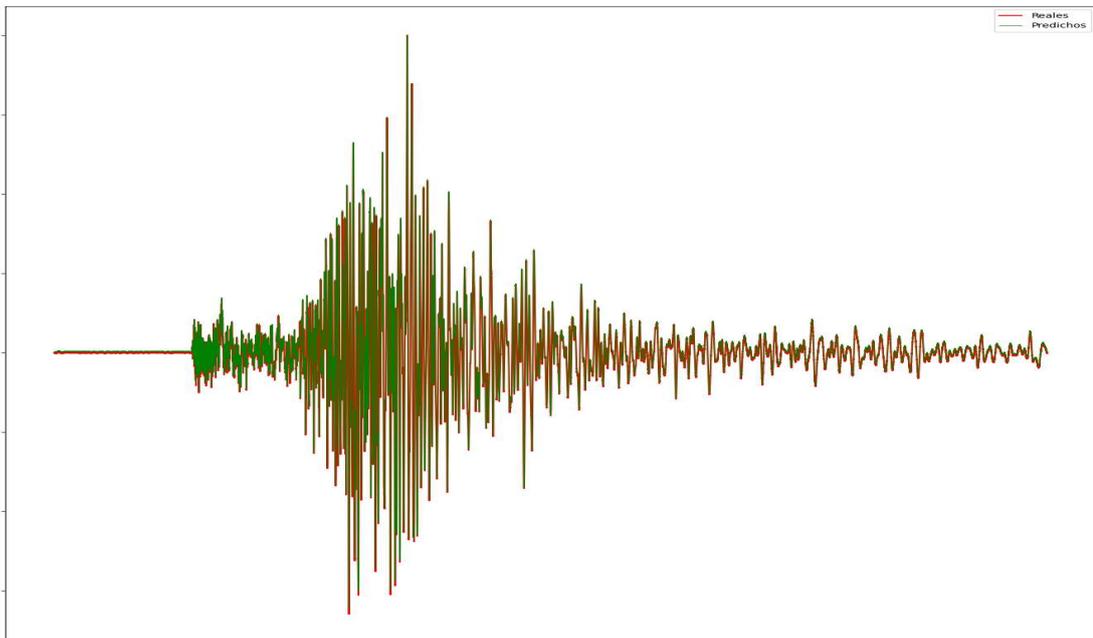
### Sismo en Chiapas el 1 de febrero de 2019

El 1 de febrero de 2019 ocurrió un terremoto de magnitud 6.5 a las 10:42:12 (hora local), con epicentro 37 km al suroeste de Ciudad Hidalgo, Chiapas. En la tabla 3.4 se muestra el análisis de dos sismogramas en la componente espacial Este-Oeste con las redes neuronales recurrentes Vanilla-LSTM y Vanilla-GRU. Los sismogramas de las estaciones sísmicas ZIIG (Zacatecas, Zacatecas) y PZIG (El Pozo, Ciudad de México) contienen 94 800 lecturas cada uno de ellos. En las figuras 3.8 y 3.9 se muestran las lecturas reales de los sismogramas contra las lecturas obtenidas con la red neuronal recurrente Vanilla-GRU.

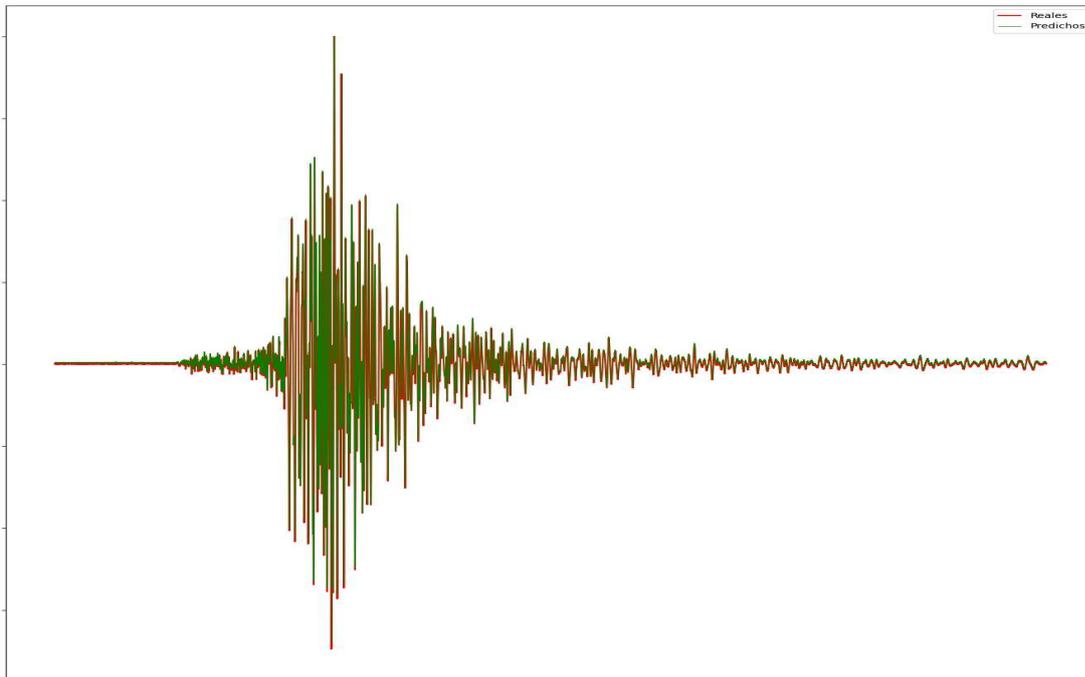
Capítulo 3. Redes neuronales recurrentes y convolucionales unidimensionales aplicadas a sismogramas para predicción

Estación Sísmica	Configuración y Modelo	Error Absoluto Promedio en la totalidad de las predicciones	Máximo Error Absoluto en una predicción	Desviación Estándar de los Errores Absolutos
ZIIG	Vanilla LSTM	0.7679 %	20.9610 %	0.9289 %
	Vanilla GRU	<b>0.6410 %</b>	<b>19.2834 %</b>	<b>0.8350 %</b>
PZIG	Vanilla LSTM	0.6439 %	22.8567 %	0.7646 %
	Vanilla GRU	<b>0.5502 %</b>	<b>20.7793 %</b>	<b>0.6782 %</b>

**Cuadro 3.4:** Desempeño de modelos LSTM y GRU para sismogramas del terremoto ocurrido el 1 de febrero de 2019 en Ciudad Hidalgo, Chiapas. Se presentan los resultados de sismogramas de dos estaciones sísmicas.



**Figura 3.8:** Sismograma de la estación ZIIG (Zacatecas, Zacatecas) del terremoto del 1 de febrero de 2019 en Ciudad Hidalgo, Chiapas, contra valores predichos (color verde) por la red neuronal Vanilla-GRU.



**Figura 3.9:** *Sismograma de la estación PZIG (EL Pozo, Ciudad de México) del terremoto del 1 febrero de 2019 en Ciudad Hidalgo, Chiapas, contra valores predichos (color verde) por la red neuronal Vanilla-GRU.*

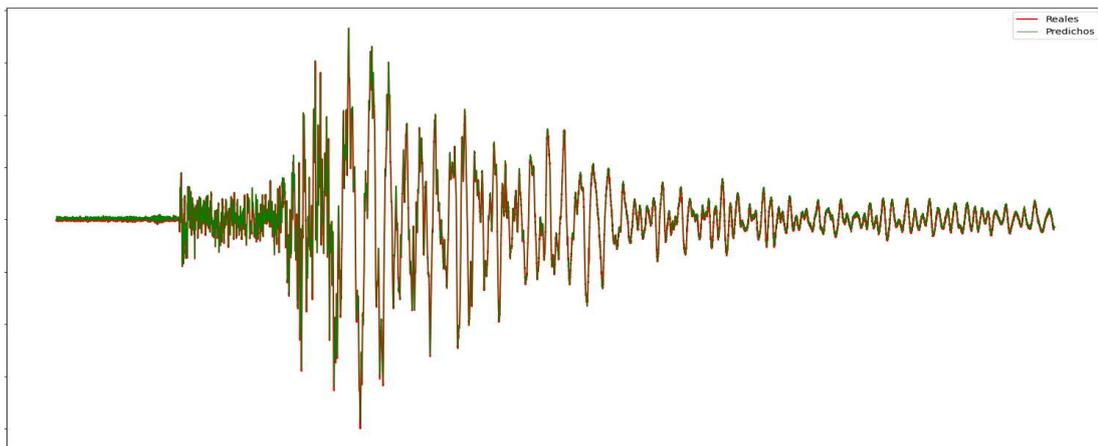
### Sismo en Chiapas el 19 de noviembre de 2019

El 19 de noviembre de 2019 ocurrió un sismo de magnitud 6.4 a las 22:27:06 (hora local), con epicentro 164 km al suroeste de Ciudad Hidalgo, Chiapas. En la tabla 3.5 se observa el análisis de tres sismogramas pertenecientes a la estación sísmica PZIG (El Pozo, Ciudad de México) en las componentes espaciales Este-Oeste, Norte-Sur y Vertical con las redes neuronales recurrentes Vanilla-LSTM y Vanilla-GRU. Cada uno de los sismogramas contiene 95 400 lecturas. En la figura 3.10 se muestra las lecturas reales de los sismogramas contra las lecturas obtenidas con la red neuronal recurrente Vanilla-GRU.

Capítulo 3. Redes neuronales recurrentes y convolucionales unidimensionales aplicadas a sismogramas para predicción

Estación Sísmica	Configuración y Modelo	Error Absoluto Promedio en la totalidad de las predicciones	Máximo Error Absoluto en una predicción	Desviación Estándar de los Errores Absolutos
PZIG (Este-Oeste)	Vanilla LSTM	0.6806 %	15.9734 %	0.7871 %
	Vanilla GRU	<b>0.6030 %</b>	<b>15.2415 %</b>	<b>0.7253 %</b>
PZIG (Norte-Sur)	Vanilla LSTM	0.5995 %	16.6416 %	0.6536 %
	Vanilla GRU	<b>0.5555 %</b>	<b>15.6694 %</b>	<b>0.5755 %</b>
PZIG (Vertical)	Vanilla LSTM	0.5725 %	16.3394 %	0.6151 %
	Vanilla GRU	<b>0.5437 %</b>	<b>15.8272 %</b>	<b>0.5608 %</b>

**Cuadro 3.5:** Desempeño de modelos LSTM y GRU para sismogramas del terremoto ocurrido el 19 de noviembre de 2019 en Ciudad Hidalgo, Chiapas. Se presentan los resultados de tres sismogramas.



**Figura 3.10:** Sismograma de la estación PZIG (El Pozo, Ciudad de México) en la componente espacial vertical para el terremoto del 19 noviembre de 2019 en Ciudad Hidalgo, Chiapas, contra valores predichos (color verde) por la red neuronal Vanilla-GRU.

Hasta este momento se han analizado sismogramas en la misma componente (Este-Oeste) del mismo evento sísmico (19 septiembre 2017 en Puebla, México) con el que fueron entrenadas las redes neuronales recurrentes Vanilla-LSTM y Vanilla-GRU. También se analizaron sismogramas del mismo evento sísmico con el cual fueron entrenadas las redes neuronales recurrentes en las componentes espaciales Norte-Sur y Vertical; componentes espaciales con las cuales no fueron entrenadas las redes neuronales recurrentes Vanilla. Posteriormente, se analizaron sismogramas de terremotos ocurridos en México en las tres componentes espaciales de sismogramas pertenecientes a eventos sísmicos diferentes al evento sísmico con el cual se entrenaron las redes recurrentes Vanilla-LSTM y Vanilla-GRU.

Ahora es momento de analizar sismogramas de eventos sísmicos ocurridos en países diferentes a México.

#### **3.6.4. Análisis de sismogramas de eventos sísmicos ocurridos fuera de México**

Se analizarán sismogramas de cinco terremotos ocurridos en los cinco continentes del mundo, un evento sísmico por continente y tres sismogramas por evento sísmico. Los sismogramas se obtendrán de estaciones sísmicas esparcidas en diferentes continentes del mundo. Como se mencionó en párrafos anteriores, el entrenamiento de las redes neuronales recurrentes que se ha venido utilizando para analizar sismogramas no se actualizará, el entrenamiento seguirá siendo con el sismograma en la componente Este-Oeste de la estación sísmica IGIG del terremoto ocurrido el 19 de septiembre del 2017 en Puebla, México.

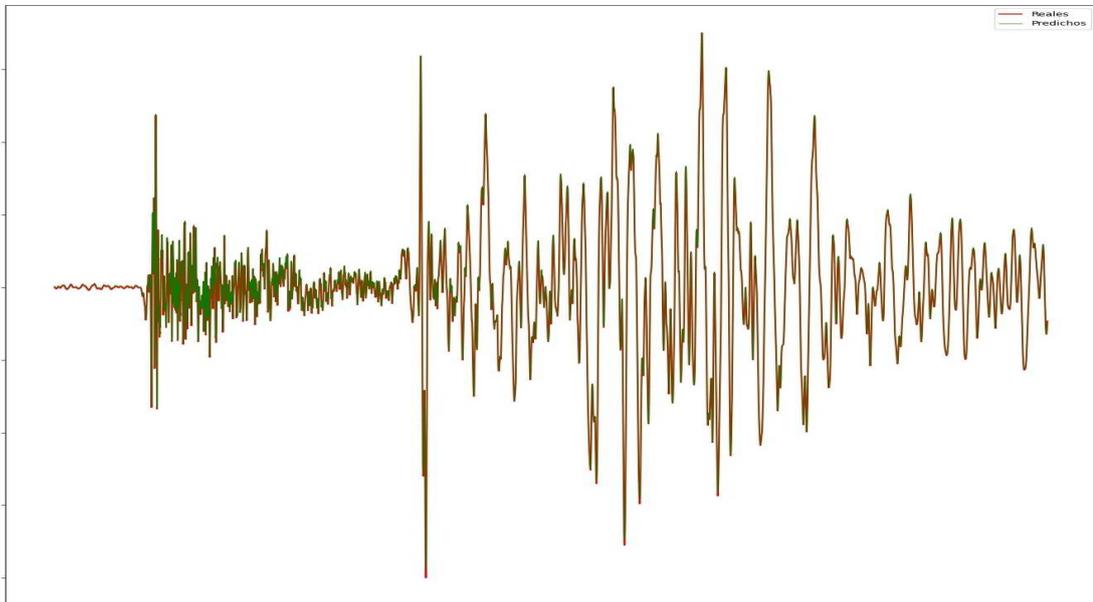
##### **Sismo al norte de Chile**

El 3 de junio de 2020 ocurrió un sismo de magnitud 6.8 a las 07:35:34 (hora del centro) con epicentro al norte de Chile. En la tabla 3.6 se observa el análisis de tres sismogramas en la componente espacial Este-Oeste con las redes neuronales recurrentes Vanilla-LSTM y Vanilla-GRU. Los sismogramas de las estaciones sísmicas TRQA (Tornsquist, Argentina), SBA (Scott Base, Antártica) y KIEV (Kiev, Ucrania) contienen 26 400, 26 400 y 13 200 lecturas, respectivamente. En las figuras 3.11 y 3.12 se muestran las lecturas reales de los sismogramas contra las lecturas obtenidas con la red neuronal Vanilla-GRU.

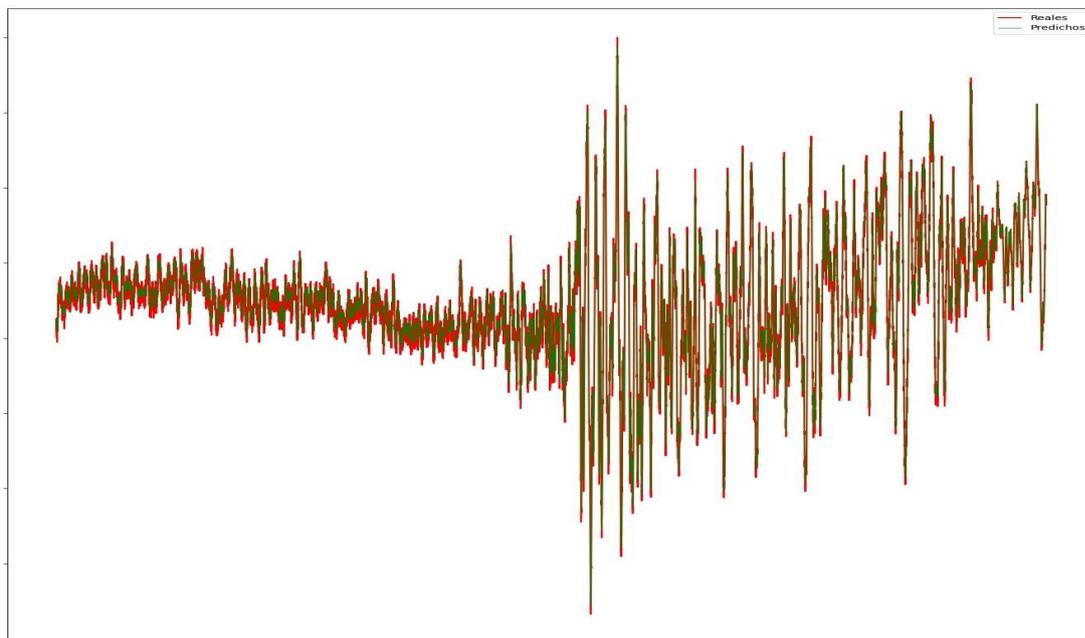
### Capítulo 3. Redes neuronales recurrentes y convolucionales unidimensionales aplicadas a sismogramas para predicción

Estación Sísmica	Configuración y Modelo	Error Absoluto Promedio en la totalidad de las predicciones	Máximo Error Absoluto en una predicción	Desviación Estándar de los Errores Absolutos
TRQA	Vanilla LSTM	1.0364 %	26.6252 %	1.3406 %
	Vanilla GRU	<b>0.9245 %</b>	<b>24.0225 %</b>	<b>1.1794 %</b>
SBA	Vanilla LSTM	0.9931 %	15.8834 %	1.2094 %
	Vanilla GRU	<b>0.9013 %</b>	<b>14.8487 %</b>	<b>1.0637 %</b>
KIEV	Vanilla LSTM	2.9651 %	68.5073 %	3.3124 %
	Vanilla GRU	<b>2.7618 %</b>	<b>63.3446 %</b>	<b>2.9161 %</b>

**Cuadro 3.6:** Desempeño de modelos LSTM y GRU para sismogramas en la componente Este-Oeste del terremoto ocurrido el 3 de junio de 2020 al norte de Chile. Se presentan los resultados de sismogramas de tres estaciones sísmicas.



**Figura 3.11:** Sismograma de la estación TRQA (Tornsquist, Argentina) para el terremoto del 3 de junio de 2020 al norte de Chile, contra valores predichos (color verde) por la red neuronal Vanilla-GRU.



**Figura 3.12:** Sismograma de la estación KIEV (Kiev, Ucrania) para el terremoto del 3 de junio de 2020 al norte de Chile, contra valores predichos (color verde) por la red neuronal Vanilla-GRU.

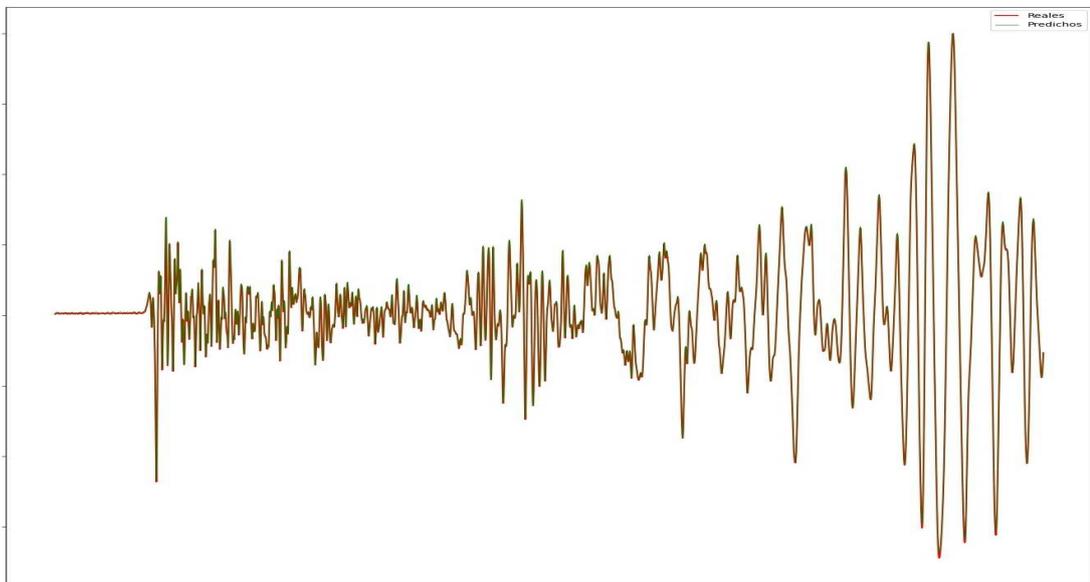
### Sismo en Creta, Grecia

El 2 de mayo de 2020 ocurrió un sismo de magnitud 6.6 a las 12:51:06 (hora del centro) con epicentro en Creta, Grecia. En la tabla 3.7 se observa el análisis de tres sismogramas en la componente espacial Este-Oeste con las redes neuronales recurrentes Vanilla-LSTM y Vanilla-GRU. Los sismogramas de las estaciones sísmicas GRFO (Grafenberg, Alemania), MBAR (Mbarara, Uganda) y WVT (Tennessee, USA) contienen 26 400 lecturas cada uno de ellos. En las figuras 3.13 y 3.14 se muestran las lecturas reales de los sismogramas contra las lecturas obtenidas con la red neuronal recurrente Vanilla-GRU.

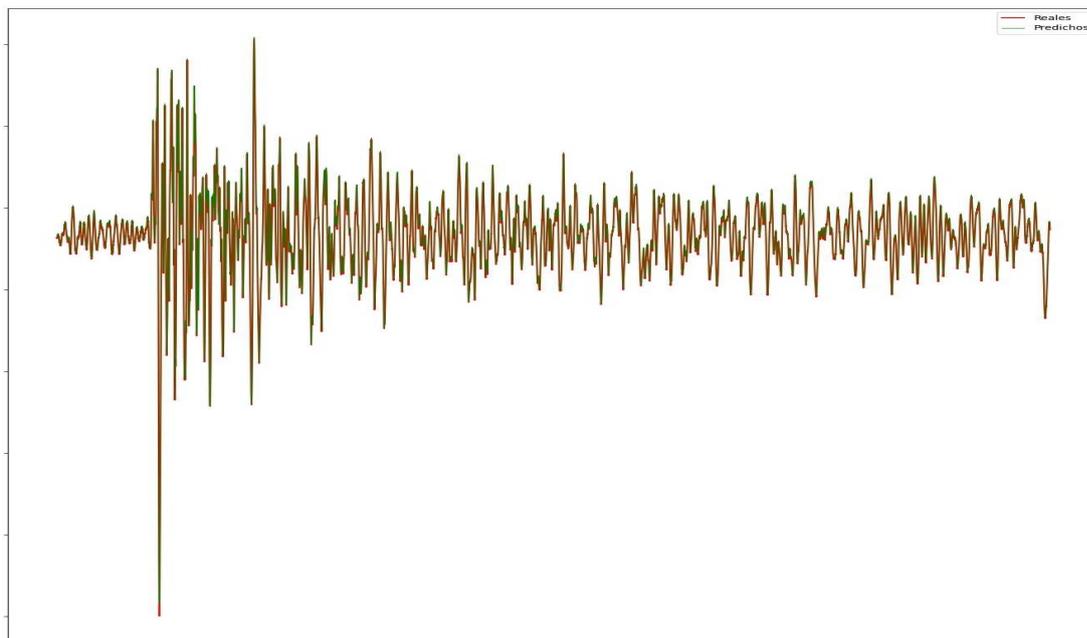
Capítulo 3. Redes neuronales recurrentes y convolucionales unidimensionales aplicadas a sismogramas para predicción

Estación Sísmica	Configuración y Modelo	Error Absoluto Promedio en la totalidad de las predicciones	Máximo Error Absoluto en una predicción	Desviación Estándar de los Errores Absolutos
GRFO	Vanilla LSTM	0.9328 %	12.5911 %	1.1455 %
	Vanilla GRU	<b>0.8189 %</b>	<b>12.0985 %</b>	<b>1.0876 %</b>
MBAR	Vanilla LSTM	0.7699 %	14.7452 %	0.9224 %
	Vanilla GRU	<b>0.6941 %</b>	<b>14.1023 %</b>	<b>0.8983 %</b>
WVT	Vanilla LSTM	0.7612 %	36.8227 %	1.1983 %
	Vanilla GRU	<b>0.7031 %</b>	<b>33.0637 %</b>	<b>1.0517 %</b>

**Cuadro 3.7:** Desempeño de modelos LSTM y GRU para sismogramas en la componente Este-Oeste del terremoto ocurrido el 2 de mayo de 2020 en Creta, Grecia. Se presentan los resultados de sismogramas de tres estaciones sísmicas.



**Figura 3.13:** Sismograma de la estación GRFO (Grafenberg, Alemania) para el terremoto del 2 de mayo de 2020 en Creta, Grecia, contra valores predichos (color verde) por la red neuronal Vanilla-GRU.



**Figura 3.14:** Sismograma de la estación WVT (Tennessee, USA) para el terremoto del 2 de mayo de 2020 en Creta, Grecia, contra valores predichos (color verde) por la red neuronal Vanilla-GRU.

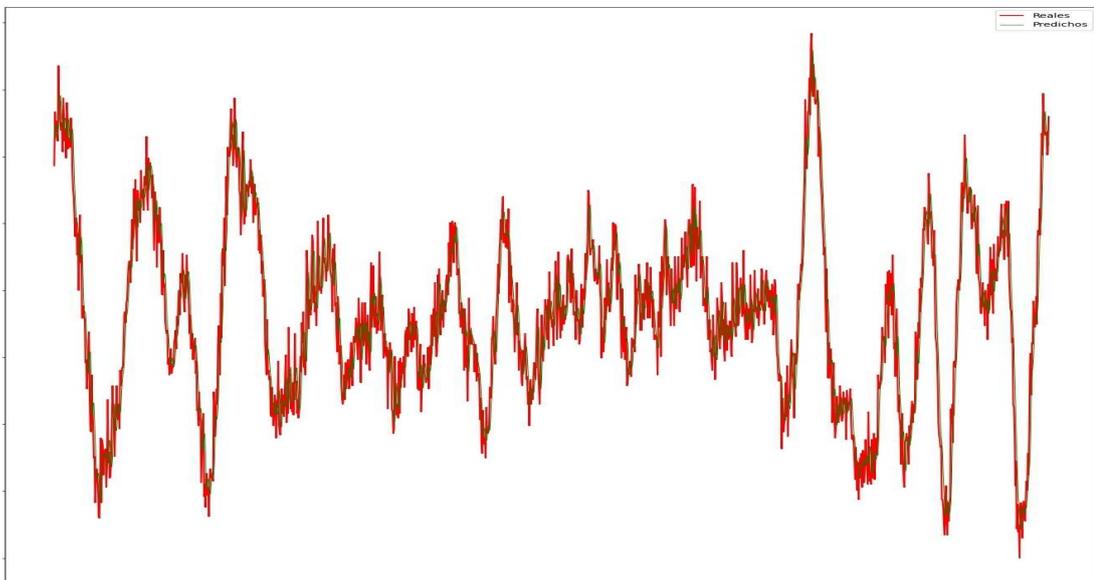
### Sismo en Sudan, África

El 20 de mayo de 1990 ocurrió un sismo de magnitud 7.2 a las 02:22:01 (hora del centro) con epicentro en Sudan, África. En la tabla 3.8 se observa el análisis de tres sismogramas en la componente espacial Este-Oeste con las redes neuronales recurrentes Vanilla-LSTM y Vanilla-GRU. Los sismogramas de las estaciones sísmicas PFO (California, USA), KIV (Kislovodsk, Rusia) y ESK (Escocia, Reino Unido) contienen 1440, 13 200 y 5760 lecturas, respectivamente. En las figuras 3.15 y 3.16 se muestran las lecturas reales de los sismogramas contra las lecturas obtenidas con la red neuronal recurrente Vanilla-GRU.

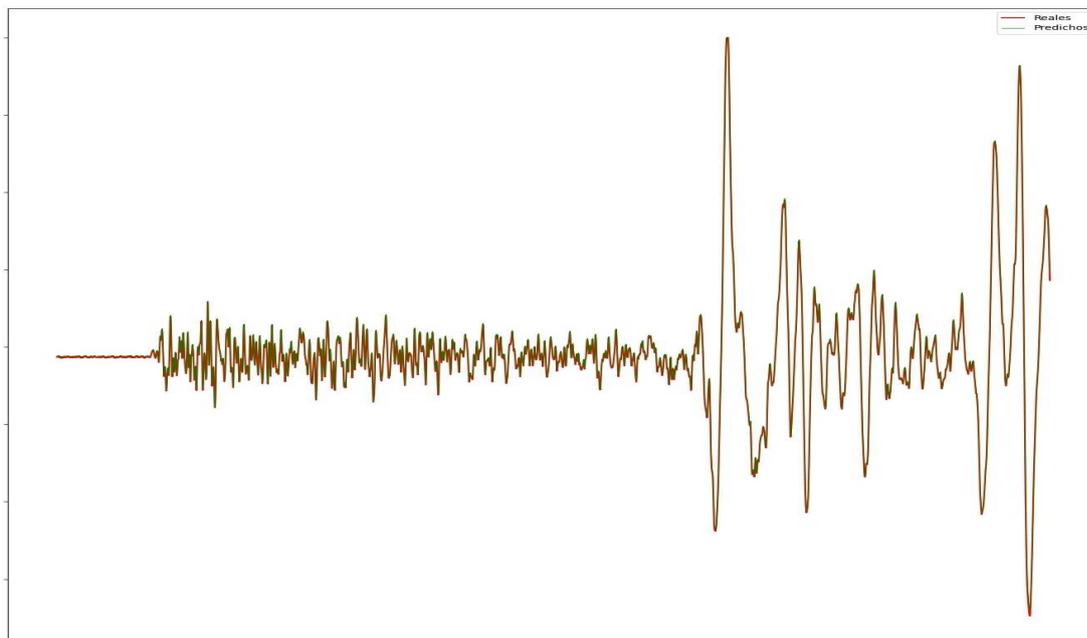
Capítulo 3. Redes neuronales recurrentes y convolucionales unidimensionales aplicadas a sismogramas para predicción

Estación Sísmica	Configuración y Modelo	Error Absoluto Promedio en la totalidad de las predicciones	Máximo Error Absoluto en una predicción	Desviación Estándar de los Errores Absolutos
PFO	Vanilla LSTM	9.8583 %	103.9340 %	9.3216 %
	Vanilla GRU	<b>9.4741 %</b>	<b>92.2121 %</b>	<b>8.7209 %</b>
KIV	Vanilla LSTM	1.0971 %	13.9426 %	1.2199 %
	Vanilla GRU	<b>0.9509 %</b>	<b>13.0960 %</b>	<b>1.1009 %</b>
ESK	Vanilla LSTM	1.9803 %	30.6515 %	2.4477 %
	Vanilla GRU	<b>1.7523 %</b>	<b>27.6654 %</b>	<b>2.1471 %</b>

**Cuadro 3.8:** Desempeño de modelos LSTM y GRU para sismogramas en la componente Este-Oeste del terremoto ocurrido el 20 de mayo de 1990 en Sudan, África. Se presentan los resultados de sismogramas de tres estaciones sísmicas.



**Figura 3.15:** Sismograma de la estación PFO (California, USA) para el terremoto del 20 de mayo de 1990 en Sudan, África, contra valores predichos (color verde) por la red neuronal Vanilla-GRU.



**Figura 3.16:** Sismograma de la estación KIV (Kislovodsk, Rusia) para el terremoto del 20 de mayo de 1990 en Sudan, África, contra valores predichos (color verde) por la red neuronal Vanilla-GRU.

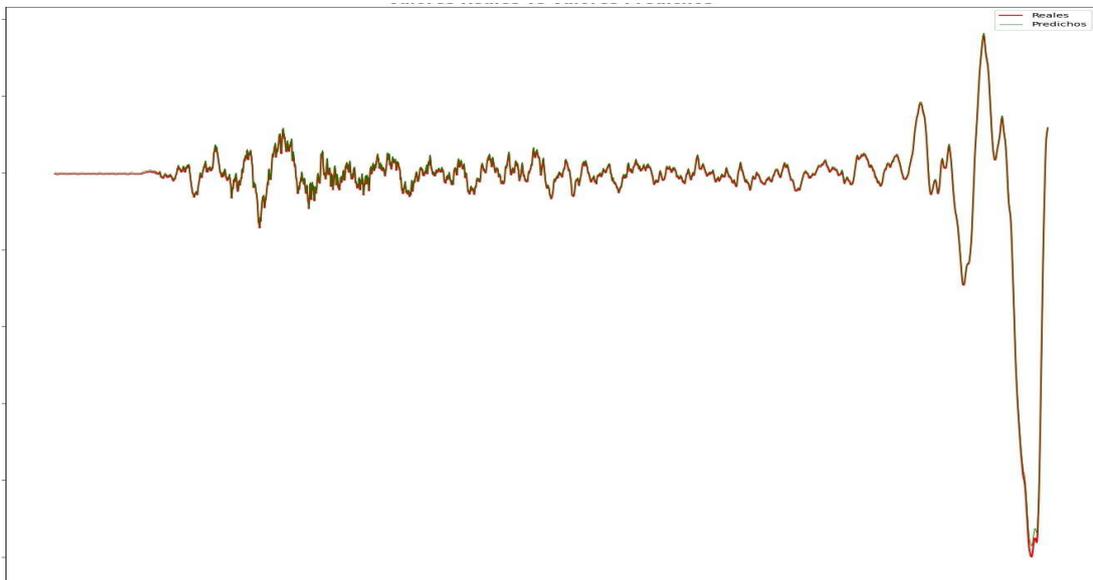
### Sismo en Honshu, Japón

El 11 de marzo de 2011 ocurrió un sismo de magnitud 9.1 a las 05:46:23 (hora del centro) con epicentro en Honshu, Japón. En la tabla 3.9 se observa el análisis de tres sismogramas en la componente espacial Este-Oeste con las redes neuronales recurrentes Vanilla-LSTM y Vanilla-GRU. Los sismogramas de las estaciones sísmicas INCN (Inchon, República de Corea), MBWA (Marble Bar, Australia) y TEIG (Nayarit, México) contienen 26 400 lecturas cada uno de ellos. En las figuras 3.17 y 3.18 se muestran las lecturas reales de los sismogramas contra las lecturas obtenidas con la red neuronal recurrente Vanilla-GRU.

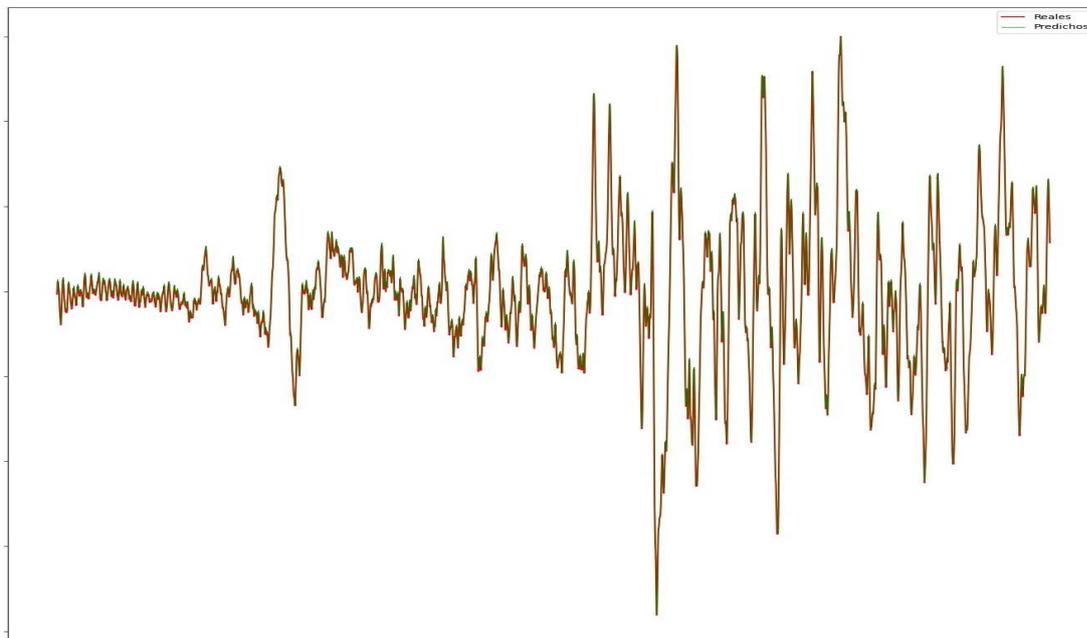
### Capítulo 3. Redes neuronales recurrentes y convolucionales unidimensionales aplicadas a sismogramas para predicción

Estación Sísmica	Configuración y Modelo	Error Absoluto Promedio en la totalidad de las predicciones	Máximo Error Absoluto en una predicción	Desviación Estándar de los Errores Absolutos
INCN	Vanilla LSTM	2.6086 %	54.7000 %	3.9990 %
	Vanilla GRU	<b>2.4857 %</b>	<b>49.5207 %</b>	<b>3.8690 %</b>
MBWA	Vanilla LSTM	<b>0.4048 %</b>	11.0304 %	<b>0.9312 %</b>
	Vanilla GRU	0.4050 %	<b>10.9688 %</b>	0.9353 %
TEIG	Vanilla LSTM	0.7328 %	18.9839 %	0.8416 %
	Vanilla GRU	<b>0.6693 %</b>	<b>17.5135 %</b>	<b>0.7578 %</b>

**Cuadro 3.9:** Desempeño de modelos LSTM y GRU para sismogramas en la componente Este-Oeste del terremoto ocurrido el 11 de marzo de 2011 en Honshu, Japón. Se presentan los resultados de sismogramas de tres estaciones sísmicas.



**Figura 3.17:** Sismograma de la estación MBWA (Marble Bar, Australia) para el terremoto del 11 de marzo de 2011 en Honshu, Japón, contra valores predichos (color verde) por la red neuronal Vanilla-GRU.



**Figura 3.18:** Sismograma de la estación TEIG (Nayarit, México) para el terremoto del 11 de marzo de 2011 en Honshu, Japón, contra valores predichos (color verde) por la red neuronal Vanilla-GRU.

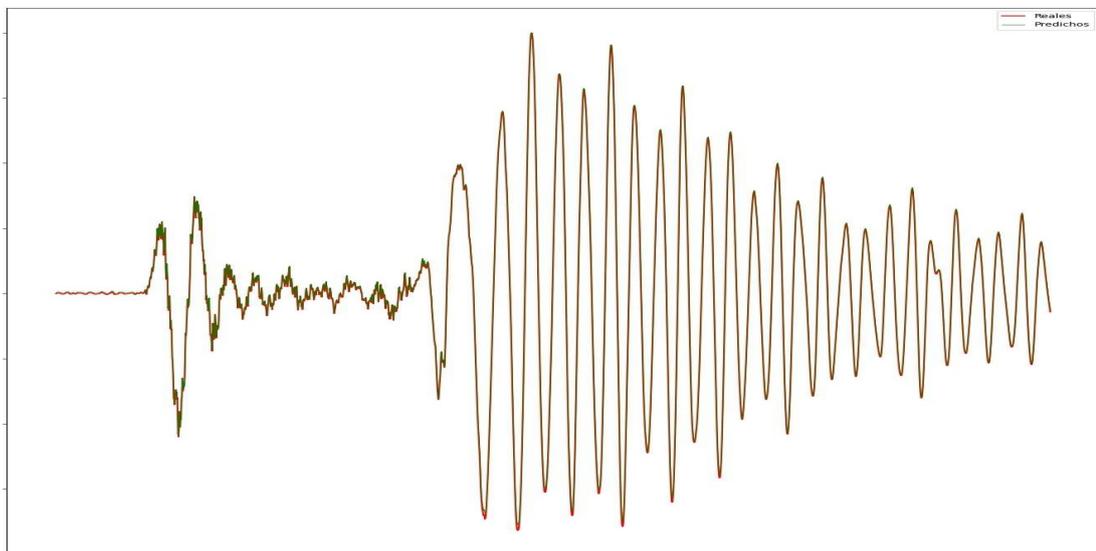
### Sismo en Isla Sur, Nueva Zelanda

El 21 de agosto de 2003 ocurrió un sismo de magnitud 7.2 a las 12:12:47 (hora del centro) con epicentro en Isla Sur, Nueva Zelanda. En la tabla 3.10 se observa el análisis de tres sismogramas en la componente espacial Este-Oeste con las redes neuronales recurrentes Vanilla-LSTM y Vanilla-GRU. Los sismogramas de las estaciones sísmicas TAU (Tasmania, Australia), SBA (Scott Base, Antártica) y TRQA (Tornsquist, Argentina) contienen 13 200, 26 400 y 26 400 lecturas, respectivamente. En las figuras 3.19 y 3.20 se muestran las lecturas reales de los sismogramas contra las lecturas obtenidas con la red neuronal recurrentes Vanilla-GRU.

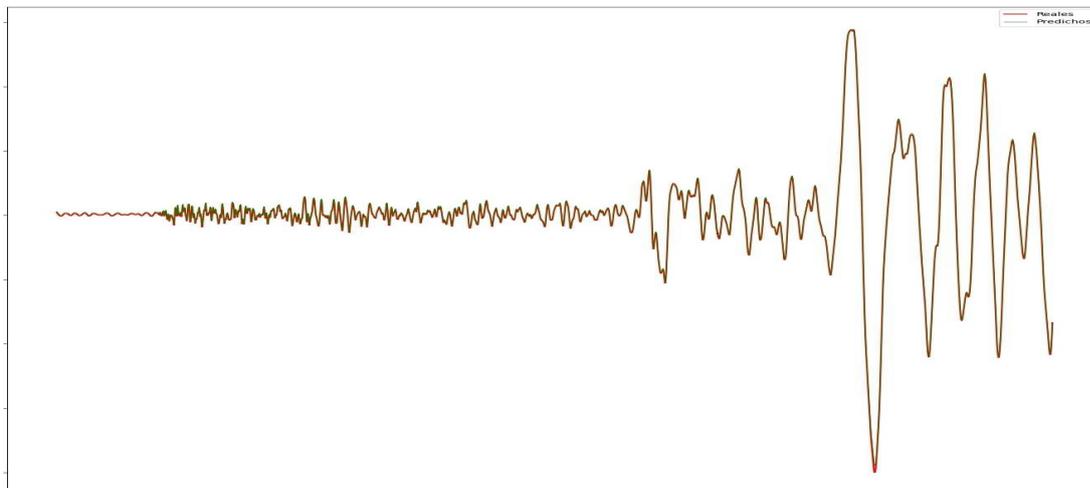
Capítulo 3. Redes neuronales recurrentes y convolucionales unidimensionales aplicadas a sismogramas para predicción

Estación Sísmica	Configuración y Modelo	Error Absoluto Promedio en la totalidad de las predicciones	Máximo Error Absoluto en una predicción	Desviación Estándar de los Errores Absolutos
TAU	Vanilla LSTM	1.5371 %	15.1306 %	1.8519 %
	Vanilla GRU	<b>1.3711 %</b>	<b>14.1552</b>	<b>1.7410 %</b>
SBA	Vanilla LSTM	0.4792 %	11.5936 %	0.7627 %
	Vanilla GRU	<b>0.4785 %</b>	<b>11.4707 %</b>	<b>0.7490 %</b>
TRQA	Vanilla LSTM	0.8033 %	26.0789 %	1.0142 %
	Vanilla GRU	<b>0.7101 %</b>	<b>23.5539 %</b>	<b>0.8995 %</b>

**Cuadro 3.10:** Desempeño de modelos LSTM y GRU para sismogramas en la componente Este-Oeste del terremoto ocurrido el 21 de agosto de 2003 en Isla Sur, Nueva Zelanda. Se presentan los resultados de sismogramas de tres estaciones sísmicas.



**Figura 3.19:** Sismograma de la estación TAU (Tasmania, Australia) para el terremoto del 21 de agosto de 2003 en Isla Sur, Nueva Zelanda, contra valores predichos (color verde) por la red neuronal Vanilla-GRU.



**Figura 3.20:** Sismograma de la estación SBA (Scott Base, Antártica) para el terremoto del 21 de agosto de 2003 en Isla Sur, Nueva Zelanda, contra valores predichos (color verde) por la red neuronal Vanilla-GRU.

Recapitulando, en esta sección se han analizado sismogramas con epicentros ubicados en cada uno de los diferentes continentes del mundo, con estaciones sísmicas distribuidas, igualmente, en los cinco continentes del mundo, en años que van desde la década de 1990 hasta el año 2020 y los resultados se han mantenido relativamente estables, a excepción del sismograma de la estación PFO localizada en California, Estados Unidos; su gráfica se muestra en la figura 3.15 y sus métricas en la tabla 3.8, dicho sismograma tan solo contaba con 1436 lecturas para un sismo ocurrido en 1990. Es notable que la frecuencia de muestreo de los sismómetros de ese tiempo eran bajas con respecto de los sismómetros más recientes, y seguramente la calidad de los sismómetros actuales es mejor que la de los sismómetros utilizados en 1990.

### 3.6.5. Comparando resultados de diferentes configuraciones de redes neuronales recurrentes con los obtenidos por las redes Vanilla-LSTM y Vanilla-GRU

En esta sección se optimizarán diferentes configuraciones de redes neuronales recurrentes tanto como sea posible con el objetivo de superar algunos de los resultados obtenidos con las redes neuronales recurrentes Vanilla-LSTM y Vanilla-GRU que se utilizaron para analizar los sismogramas de secciones anteriores. Se presentarán las siguientes configuraciones: *Stack-GRU*, *Bi-GRU*, *Conv1D-GRU*, *Conv1D-BiGRU* y *BiGRU-Attention*.

Capítulo 3. Redes neuronales recurrentes y convolucionales unidimensionales aplicadas a sismogramas para predicción

Estación Sísmica	Configuración y Modelo	Error Absoluto Promedio en la totalidad de las predicciones	Máximo Error Absoluto en una predicción	Desviación Estándar de los Errores Absolutos
AAIG	Vanilla-GRU	0.9158 %	14.2492 %	1.1293 %
	Stacked-GRU	1.1350 %	9.1222 %	0.9597 %
	Bi-GRU	1.0867 %	11.5832 %	1.0965 %
	Conv1D-GRU	0.3765 %	14.1666 %	0.5229 %
	Conv1D-BiGRU	0.3162 %	<b>2.8289 %</b>	<b>0.2111 %</b>
	BiGRU-Attention	<b>0.1557 %</b>	3.7804 %	0.2196 %
ZAIG	Vanilla GRU	0.5733 %	12.4035 %	0.9325 %
	Stacked-GRU	0.8301 %	4.5483 %	0.5021 %
	Bi-GRU	0.9585 %	6.9605 %	0.6703 %
	Conv1D-GRU	0.2490 %	14.0033 %	0.5678 %
	Conv1D-BiGRU	0.2079 %	<b>3.5211 %</b>	<b>0.1498 %</b>
	BiGRU-Attention	<b>0.0953 %</b>	4.1858 %	0.2279 %
THIG	Vanilla GRU	1.1208 %	21.5626 %	1.5502 %
	Stacked-GRU	1.3551 %	15.9183 %	1.4397 %
	Bi-GRU	1.2253 %	18.8218 %	1.5549 %
	Conv1D-GRU	0.5444 %	10.3047 %	0.6601 %
	Conv1D-BiGRU	0.4766 %	6.8224 %	0.4260 %
	BiGRU-Attention	<b>0.3081 %</b>	<b>5.2046 %</b>	<b>0.3620 %</b>

3.6. Resultados obtenidos con la redes neuronales recurrentes

Estación Sísmica	Configuración y Modelo	Error Absoluto Promedio en la totalidad de las predicciones	Máximo Error Absoluto en una predicción	Desviación Estándar de los Errores Absolutos
RPIG	Vanilla GRU	0.7096 %	19.6051 %	1.0532 %
	Stacked-GRU	1.0226 %	13.7615 %	0.9584 %
	Bi-GRU	0.8191 %	16.4429 %	1.0246 %
	Conv1D-GRU	0.3938 %	7.7373 %	0.2162 %
	Conv1D-BiGRU	0.4018 %	3.1079 %	<b>0.1157 %</b>
	BiGRU-Attention	<b>0.2323 %</b>	<b>2.6031 %</b>	0.1453 %
ANIG	Vanilla-GRU	0.6842 %	12.9554 %	0.8500 %
	Stacked-GRU	0.9534 %	6.6293 %	0.6831 %
	Bi-GRU	0.8113 %	9.1625 %	0.7756 %
	Conv1D-GRU	0.4034 %	14.0822 %	0.4335 %
	Conv1D-BiGRU	0.3833 %	<b>3.5903 %</b>	<b>0.1585 %</b>
	BiGRU-Attention	<b>0.1731 %</b>	4.0873 %	0.1678 %
TJIG	Vanilla-GRU	<b>0.8188 %</b>	18.3341 %	0.9641 %
	Stacked-GRU	0.9387 %	<b>7.5367 %</b>	0.6619 %
	Bi-GRU	0.8712 %	11.2936 %	0.8049 %
	Conv1D-GRU	0.8683 %	18.5504 %	0.7173 %
	Conv1D-BiGRU	0.9416 %	12.1412 %	0.6898 %
	BiGRU-Attention	0.8456 %	13.1974 %	<b>0.6197 %</b>

Capítulo 3. Redes neuronales recurrentes y convolucionales unidimensionales aplicadas a sismogramas para predicción

Estación Sísmica	Configuración y Modelo	Error Absoluto Promedio en la totalidad de las predicciones	Máximo Error Absoluto en una predicción	Desviación Estándar de los Errores Absolutos
MYIG	Vanilla-GRU	0.5982 %	20.0177 %	0.7173 %
	Stacked-GRU	0.9348 %	13.1674 %	0.6988 %
	Bi-GRU	0.7030 %	11.2936 %	0.8049 %
	Conv1D-GRU	0.4882 %	10.8633 %	0.3154 %
	Conv1D-BiGRU	0.4674 %	4.4633 %	0.1956 %
	BiGRU-Attention	<b>0.2199 %</b>	<b>4.1379 %</b>	<b>0.1655 %</b>

**Cuadro 3.11:** Comparación del desempeño de las diferentes configuraciones de redes neuronales recurrentes para el evento sísmico del 19 de septiembre de 2017 en Puebla, México. Se presentan los resultados de sismogramas de siete estaciones sísmicas ubicadas en la república mexicana, todas ellos en la componente espacial Este-Oeste y sus respectivas métricas. En letra negrita se resaltan los mejores resultados.

En la tabla 3.11 observamos las métricas obtenidas para las diferentes configuraciones propuestas de redes neuronales recurrentes, el modelo de red neuronal recurrente usado para cada una de ellas fue GRU. Se analizaron siete estaciones sísmicas diferentes en la componente espacial Este-Oeste para el terremoto del 19 de septiembre de 2017 en Puebla, México.

# Capítulo 4

---

## Análisis de Resultados

### 4.1. Resumen

En el capítulo anterior se realizaron las siguientes actividades:

- Se analizaron sismogramas en las tres componentes espaciales (Este-Oeste, Norte-Sur y Vertical) para eventos sísmicos ocurridos en México.
- Se analizaron sismogramas ocurridos en Chile, Grecia, Sudan, Japón y Nueva Zelanda.
- Se puso a prueba los dos modelos más conocidos de redes neuronales recurrentes: LSTM y GRU.
- Se construyeron y optimizaron diferentes configuraciones de redes neuronales recurrentes con el objetivo de superar los resultados obtenidos con la red neuronal recurrente Vanilla-GRU.

Los sismogramas analizados fueron extraídos de estaciones sísmicas ubicadas alrededor del mundo para sismos ocurridos en un intervalo de tiempo que va desde el año 1990 hasta el año 2020.

Los resultados base presentados en el capítulo anterior se obtuvieron con la configuración más básica de redes neuronales recurrentes: *Vanilla*. Obviamente, al igual que las demás configuraciones de redes neuronales recurrentes, la configuración Vanilla-GRU se optimizó tanto como fue posible.

### 4.2. Resultados obtenidos

En los siguientes secciones se comentarán los resultados más sobresalientes encontrados a lo largo del capítulo anterior.

### 4.2.1. Acerca de los sismos ocurridos en México que se analizaron con las redes neuronales recurrentes

Primeramente, hay que recalcar, nuevamente, que las redes neuronales Vanilla se entrenaron con un único sismograma y en una sola ocasión, todos los resultados presentados (gráficas y tablas) se deben a este único entrenamiento el cual no se actualizó en ningún momento a lo largo de este trabajo de tesis. Dicho sismograma correspondía a la componente espacial Este-Oeste y fue extraído de la estación sísmica IGIG (Irapuato, Guanajuato) para el terremoto ocurrido el 19 septiembre de 2017 en Puebla, México.

Se entrenaron las redes neuronales recurrentes con dicho sismograma y se aislaron las últimas 400 lecturas para formar el conjunto de prueba; habitualmente lo que se hace al formar el conjunto de entrenamiento y prueba es seleccionar aleatoriamente elementos del conjunto de datos para formar dichos conjuntos de tal forma que se asegure que la red neuronal tanto en su entrenamiento como en su prueba de desempeño haya estado expuesta a una parte significativa de todos los elementos del conjunto de datos. En esta ocasión se eligió segregar esas últimas 400 lecturas de la serie de tiempo para formar el conjunto de prueba con la finalidad de que al realizar la prueba de desempeño se obtuvieran resultados más reveladores en lo que concierne al desempeño que tendría la red neuronal en sus predicciones. Al evaluar el desempeño de la red neuronal con este conjunto de prueba de desempeño el resultado de las métricas fue aceptable, lo cual fue satisfactorio, pues, aunque se sabe que existe una ecuación que describe la propagación de las ondas sísmicas, también es cierto que al ver la gráfica de cualquier sismograma pareciera no haber ningún patrón, tendencia o estacionalidad que haga imaginar o intuir el porqué de los trazos tan irregulares que se observan en las gráficas de los sismogramas.

Posteriormente, se analizaron sismogramas del mismo evento sísmico del 19 de septiembre de 2017 en la componente espacial Este-Oeste extraídos de varias estaciones sísmicas pertenecientes a la red de estaciones sísmicas del Servicio Sismológico Nacional (SSN). Las métricas obtenidas mostraron resultados favorables, el porcentaje del error promedio absoluto de las predicciones por sismograma no superó el 1.3 %, la desviación estándar no superó el límite de 1.8 %; lo cual deja claro que los errores por predicción no tenían fluctuaciones considerables y que las predicciones realizadas por las redes neuronales recurrentes Vanilla estaban muy próximas a los valores reales. En las gráficas que se presentaron de las lecturas reales de los sismogramas contra las lecturas predichas por las redes neuronales Vanilla se pudo constatar visualmente que, efectivamente, las predicciones eran muy cercanas a las lecturas reales de los sismogramas.

Al ver la efectividad de las predicciones en sismogramas en la componente espacial Este-Oeste fue natural el poner a prueba las redes recurrentes Vanilla en sismogramas en las componentes espaciales Norte-Sur y vertical. Contrario a lo que se esperaba, las métricas obtenidas por las redes recurrentes Vanilla no se vieron afectadas, las predicciones obtenidas por las redes recurrentes siguieron manteniéndose dentro de los rangos en los que se mantuvieron las predicciones para sismogramas en la componente espacial Este-Oeste, lo cual, ciertamente, no era esperado, puesto que hay algunas ondas sísmicas como las ondas Love y ondas primarias que no tienen presencia en los sismogramas de la componente espacial vertical y se esperaba que las redes neuronales recurrentes Vanilla no se desempeñarían tan bien ya que no fueron expuestas a este tipo de patrones de comportamiento de algunas ondas sísmicas, sin embargo, se obtuvieron resultados favorables.

Al tener redes neuronales recurrentes que se desempeñaban bastante bien en sus predicciones en sismogramas en las tres componentes espaciales pertenecientes al mismo evento sísmico con el cual fueron entrenadas se decidió ponerlas a prueba con sismogramas de eventos sísmicos diferentes al del 19 de septiembre de 2017 en Puebla, México. Fue hasta cierto punto natural pensar que al cambiar el evento sísmico las redes neuronales recurrentes menguarían en su desempeño, pero no fue así, los resultados de las métricas al realizar las predicciones de sismogramas pertenecientes a los sismos ocurridos el 8 de septiembre de 2017 y 1 de febrero de 2019, ambos en Chiapas, México, se mantuvieron estables, no hubo ningún efecto negativo en el desempeño de las redes neuronales recurrentes. Hasta este punto ya no se tenía dudas de que se había logrado entrenar una red neuronal recurrente capaz de aprender el comportamiento que rige la propagación de las ondas sísmicas y por ende realizar predicciones muy próximas a las lecturas reales con base en lecturas pasadas.

Es posible asegurar, en vista de los resultados que se comentan en esta sección, que las redes recurrentes neuronales Vanilla aprendieron una relación de mapeo capaz de predecir lecturas contenidas en sismogramas con base a lecturas de sucesiones inmediatamente pasadas del sismograma en cuestión. La función de mapeo aprendida por las redes neuronales recurrentes ha mostrado ser efectiva para sismos acontecidos en territorio mexicano sin importar la fecha de ocurrencia, sismos que tienen cualidades particulares en común, tal como ser generados por el movimiento de placas tectónicas convergentes. En cuanto a las propiedades físicas de los suelos por los cuales se propagan las ondas sísmicas podemos concluir que no son un factor que afecten el desempeño de las redes neuronales recurrentes Vanilla puesto que se analizaron sismogramas de estaciones sísmicas ubicadas en el sur, centro y norte del país; recordemos que las características de los suelos del centro

del país son muy peculiares por su origen mismo, son suelos catalogados como blandos o de transición. También es posible adelantar que el analizar sismos en los cuales las ondas sísmicas tengan que atravesar medios líquidos para ser captadas por alguna estación sísmica no será un factor que diezme el desempeño de las redes neuronales recurrentes Vanilla, ya que se analizó un sismo con epicentro en Puebla, México y se analizaron sismos ocurridos a kilómetros de distancia de las costas de Chiapas, México.

### **4.2.2. Acerca de los sismogramas analizados pertenecientes a sismos con epicentro fuera del territorio mexicano**

Se pensó que al analizar sismogramas de eventos sísmicos fuera de México se podría impactar negativamente en el desempeño de las predicciones de las redes neuronales recurrentes Vanilla ya que se estarían involucrando diferentes tipos de orígenes en la generación de los eventos sísmicos; muchos de los eventos sísmicos en México se deben a placas tectónicas en subducción, fenómeno también conocido como límites convergentes. Probablemente, al analizar sismogramas de eventos sísmicos debidos a interacciones entre placas tectónicas diferentes a la subducción, por ejemplo, límites divergentes (placas alejándose una con respecto de la otra) o límites transformantes (placas tectónicas deslizándose y haciendo contacto en sus fronteras una con respecto de la otra) tendría un impacto negativo en la calidad de las predicciones, quizás también se podrían ver afectados los resultados de las predicciones al exponer las redes neuronales recurrentes Vanilla a terremotos intraplacas los cuales no tienen su origen en la interacción directa de placas tectónicas entre sí. También se pensó que al involucrar tipos de suelos con propiedades físicas distintas a las del territorio mexicano podría afectar el desempeño de las predicciones de las redes neuronales recurrentes ya que las ondas sísmicas se estarían propagando por suelos con propiedades distintas a las de México, aunado a esto, se pensó que al extraer sismogramas de estaciones sísmicas separadas del hipocentro sísmico por medio de océanos o mares podría, igualmente, disminuir el desempeño de las redes recurrentes Vanilla-LSTM y Vanilla-GRU, pues, sabemos que las ondas secundarias no se propagan en medios líquidos y por tanto se estaría exponiendo a la red neuronal recurrente a situaciones ajenas a las que experimentó en su entrenamiento.

En general, las métricas obtenidas por las redes neuronales recurrentes Vanilla para sismogramas de terremotos ocurridos fuera de México resultaron ser tan buenas como para sismogramas pertenecientes al sismo con el cual fueron entrenadas las redes neuronales recurrentes, no hubo un impacto notorio que fuera observable en todas las métricas obtenidas de los diferentes sismogramas analizados.

Planteando un umbral de 3 % de tolerancia en el porcentaje del promedio de los errores absolutos y 3.5 % en la desviación estándar, se tendría que tan solo las métricas de un sismograma superaron esas tolerancias, concretamente, el sismograma de la estación sísmica PFO ubicada en California, Estados Unidos, para un sismo ocurrido en Sudan, África, el 20 de mayo de 1990. Entre las particularidades de este sismo y el ya mencionado sismograma tenemos que: Sudan es un país que se encuentra en la región central de África, por tanto fue un sismo catalogado como intraplaca, además, la estación sísmica de la cual se extrajo el sismograma está cruzando el océano atlántico, y, por otra parte, los sismógrafos de la década de los 90 no tenían la misma frecuencia de muestreo y calidad en las mediciones que los sismógrafos más actuales poseen, en ese sismograma se obtuvieron tan solo 1440 lecturas cuando lo más común fue analizar sismogramas con un promedio de 30 000 lecturas. No se puede señalar con certeza la peculiaridad que provocó que se obtuvieran métricas con porcentajes de errores promedio y desviaciones estándar tan elevadas, cerca del 10 % para los errores promedio y alrededor de 9 % para la desviación estándar, aunque, es posible que alguna o todas las peculiaridades señaladas hayan tenido un impacto negativo en el desempeño de las redes neuronales recurrentes Vanilla. También hay que agregar que para ese mismo evento sísmico de la estación ESK ubicada en Escocia, Reino Unido, se obtuvo su sismograma correspondiente y dicho sismograma contaba con tan solo 5760 lecturas y al igual que en la estación PFO, el hipocentro del terremoto y la estación sísmica están separadas por medios líquidos, dos mares para ser concretos, el mar mediterráneo y el mar del norte, y aun así, los resultados de las métricas estuvieron dentro de los umbrales de tolerancia planteados para estas pruebas.

Los argumentos expuestos anteriormente hacen pensar que el problema reside en la calidad del funcionamiento en ese momento del sismógrafo que se usó en la estación sísmica PFO, ubicada en California, Estados Unidos. Aunque, por supuesto, no es una afirmación.

### **4.2.3. Acerca del desempeño de los modelos de redes neuronales LSTM y GRU**

Se analizaron un total de 37 sismogramas y de cada sismograma analizado se obtenían tres métricas, lo que hace un total de 111 métricas obtenidas a lo largo de este trabajo de tesis, de esas 111 métricas en 104 de ellas las redes neuronales recurrentes Vanilla-GRU superaron a las redes Vanilla-LSTM, lo cual equivale a un 93.69 % de las veces. Recuérdese que las redes Vanilla-LSTM y Vanilla-GRU compartían los mismos hiperparámetros y el mismo tipo de entrenamiento, la única diferencia entre una red neuronal y la otra fue la implementación del modelo de red

neuronal recurrente GRU o LSTM. La red Vanilla-GRU, en ocasiones, llegaba a mejorar el desempeño en las métricas en hasta tres décimas porcentuales, lo cual ya es un resultado considerable para un problema de regresión. Por lo cual queda claro que para este trabajo de tesis en particular, en el cual se desarrollaron modelos de aprendizaje profundo supervisado enfocado a predicción de valores contenidos en series de tiempo formadas a partir de lecturas de sismogramas es mejor utilizar el modelo de redes neuronales recurrentes del tipo GRU.

### 4.2.4. Acerca de los resultados obtenidos con las diferentes configuraciones de redes neuronales recurrentes

Con los resultados de las métricas obtenidos por las redes recurrentes Vanilla-LSTM y Vanilla-GRU siendo ya aceptables, y teniendo la posibilidad de implementar configuraciones más elaboradas de redes neuronales recurrentes apoyadas en redes neuronales convolucionales unidimensionales y mecanismos de atención, y con la posibilidad de optimizar sus hiperparámetros, se decidió construir y optimizar las siguientes configuraciones de redes neuronales recurrentes: *Stacked-GRU*, *BiGRU*, *Conv1D-GRU*, *Conv1D-BiGRU* y *BiGRU-Attention*. En todas ellas se usó el modelo de redes neuronales recurrentes GRU. El objetivo, claro está, fue superar el desempeño de la red neuronal Vanilla-GRU, objetivo que se cumplió con facilidad, pues, en algunos casos se superó el desempeño de la red Vanilla-GRU hasta por ocho décimas porcentuales, en la tabla 3.11 se puede observar los resultados mencionados.

La primera mejora que salta a la vista al observar la tabla 3.11 fue el desempeño en las métricas del porcentaje de desviación estándar y el porcentaje de máximo error relativo en una predicción que la configuración *Stacked* obtuvo, sobre todo en la métrica del máximo error relativo, en la cual se superó consistentemente y por un margen amplio a la configuración *Vanilla*. En la métrica de la desviación estándar también fue consistente la mejora en desempeño de la configuración *Stacked*, aunque no con el mismo margen de diferencia que en la métrica del máximo error relativo en una predicción.

También se observa que las configuraciones *Conv1D-GRU* y *Conv1D-BiGRU* superan a las configuraciones *Vanilla*, *Stacked* y *Bi-GRU* con bastante holgura en las métricas del porcentaje de error absoluto promedio y la desviación estándar. La configuración *Conv1D-BiGRU* supera en las tres métricas ampliamente a las configuraciones *Vanilla*, *Stacked* y *Bi-GRU*.

Observando la tabla 3.11 es fácil notar que la red neuronal que mejor desempeño presenta en la mayoría de las métricas fue la red neuronal recurrente *BiGRU-*

*Attention* y en segundo lugar la configuración de red neuronal *Conv1D-BiGRU*. Se tiene que señalar que el método de optimización que se aplicó a la configuración *BiGRU-Attention* fue el método manual. Entrenar dicha configuración de red neuronal consume un tiempo aproximado de 90 a 100 minutos, lo cual complica en demasía la opción de emplear otro algoritmo de optimización diferente al método manual por el tiempo que requeriría la implementación de algoritmos como Optimización Bayesiana o Algoritmos Genéticos los cuales para entregar buenos resultados demandan de una cantidad de pruebas que estén alrededor o por encima de las 100 repeticiones. De contar con un equipo de cómputo más potente es posible especular que la red neuronal *BiGRU-Attention* entregaría mejores resultados que los presentados en la tabla 3.11. Para las demás configuraciones no se tuvo este problema puesto que el tiempo de entrenamiento era considerablemente menor y fue posible aplicar la Optimización Bayesiana y Algoritmos Genéticos.

Es realmente de llamar la atención la mejora que el mecanismo de atención imprimió en la red neuronal recurrente bidireccional GRU presentada en este trabajo de tesis. Es conocido el gran éxito que los mecanismos de atención han tenido en diversas áreas de la Inteligencia Artificial como: *Natural Language Translation*, *Computer Vision*, *Multi-Modal Tasks*, *Graph-based Systems*, *Recomender System*, etc. En todas estas áreas los mecanismos de atención han llevado a los algoritmos que hacen uso de ellos a estar a la vanguardia (*state of the art*) en cuanto a sus resultados entregados [3].

Cabe mencionar que a la fecha hay muchas variantes de mecanismos de atención, el primer mecanismo de atención introducido por Bahdanau se le conoce como *atención aditiva*; que fue el tipo de atención que se aplicó en este trabajo de tesis. Hay que mencionar que existen otros tipos o variantes de mecanismos de atención tan conocidos como la atención aditiva, tales como: *atención multiplicativa* [20] y *auto-atención*, etc. Es seguro afirmar, al menos para este trabajo de investigación enfocado a predicción o regresión mediante diversas técnicas de Aprendizaje Profundo, que los mecanismos de atención han estado a la altura de su bien ganada fama y han resultado ser un elemento crucial para poder entregar muy buenos resultados.



# Capítulo 5

---

## Conclusiones

### 5.1. Conclusiones

El objetivo que se planteó en este trabajo de tesis fue encontrar configuraciones de redes neuronales recurrentes capaces de arrojar resultados favorables al aplicarlas a un problema de aprendizaje supervisado enfocado a predicción de valores en series de tiempo formadas a partir de lecturas contenidas en sismogramas. El objetivo se cumplió, tanto con redes neuronales recurrentes Vanilla y lo mismo sucedió con configuraciones más elaboradas apoyadas en redes neuronales convolucionales unidimensionales y mecanismos de atención.

Se probaron para este trabajo de tesis los dos modelos de redes neuronales recurrentes más empleados para tratar con secuencias de datos con dependencias temporales de largo plazo: LSTM y GRU. Además de emplear la configuración más básica de las redes neuronales recurrentes, también se probaron cinco configuraciones más que hacen uso de redes neuronales recurrentes, entre ellas, combinaciones de redes neuronales convolucionales unidimensionales con redes neuronales recurrentes y mecanismos de atención. Se planteó el problema de realizar predicciones de valores futuros en series de tiempo que contienen lecturas de sismógrafos en base a sucesiones de lecturas ordenadas por ocurrencia temporal inmediatamente anteriores al valor o valores que se deseaba predecir, esperando que las redes neuronales recurrentes fuesen capaces de establecer una relación o una función de mapeo capaz de hacer corresponder esa sucesión de valores ordenados pasados con el valor o valores inmediatamente posterior a ella.

Fue llamativo observar que redes neuronales recurrentes Vanilla entrenadas con un solo sismograma en la componente espacial Este-Oeste para el terremoto del 19 de septiembre de 2017 en Puebla, México, fuesen capaces de arrojar resultados aceptables para sismogramas generados por eventos sísmicos ocurridos en diferentes lugares del mundo, obtenidos de estaciones sísmicas distribuidas en los cinco continentes del mundo. Es claro que las redes neuronales recurrentes que se entrenaron fueron capaces de capturar el comportamiento que rige la propagación de

las ondas sísmicas a través de medios elásticos, de otra forma, las redes neuronales recurrentes no habrían sido capaces de realizar predicciones tan cercanas a los valores reales, en otras palabras. Planteando una analogía que ilustre un poco más lo que se realizó sería imaginar una función  $f$  cualquiera, de la cual no se sabe o no se está en condiciones de poder describir su comportamiento en base a sus variables independientes, pero sí que se conoce valores sucesivos de la misma, con esos valores conocidos se entrena una red neuronal recurrente esperando que la red neuronal sea capaz de forjar una relación de mapeo que de como resultado el próximo valor en la sucesión en base a valores pasados de la misma, por supuesto, se sigue sin poder describir la función  $f$  en base a sus variables independientes, pero ahora se ha ganado conocimiento acerca de su comportamiento por medio de redes neuronales recurrentes. En pocas palabras, eso fue lo que se realizó con las lecturas obtenidas de los sismogramas de los diferentes eventos sísmicos que se analizaron.

Con base en los resultados presentados en el capítulo tres se concluye que para el caso particular de este proyecto de tesis es mejor usar el modelo de redes neuronales recurrentes del tipo GRU.

Se encontró que la configuración que mejor resultados arroja en general en las métricas propuestas para evaluar el desempeño de las diferentes configuraciones es una combinación de redes neuronales recurrentes bidireccionales seguida de una capa de mecanismos de atención (*BiGRU-Attention*), en particular, atención aditiva. Y en un segundo lugar se tiene a la configuración *Conv1D-BiGRU*.

En cuanto a las técnicas de optimización, las que mejor resultados arrojaron en general fueron los Algoritmos Genéticos y la Optimización Bayesiana.

### 5.2. Trabajo a futuro

Las posibles ramificaciones que se desprenden de este trabajo de tesis con el objetivo de ampliar y mejorar los resultados obtenidos son las siguientes:

- Optimizar tanto como sea posible los hiperparámetros de la configuración de red neuronal recurrente más prometedora, *BiGRU-Attention*. Agregar a esta configuración redes neuronales convolucionales unidimensionales probablemente de como resultado mejores métricas.
- Se sabe que la forma en la cual se presentan los datos a las redes neuronales es de suma importancia, así que se propone experimentar con la forma en la cual se procesan o acondicionan las series de tiempo con el fin de obtener mejores resultados por las redes neuronales recurrentes.

- En cuanto al tipo de entrenamiento, también es posible intentar mejorarlo, por ejemplo, en lugar de entrenar la red neuronal con un solo sismograma, entrenarla con los tres sismogramas de las diferentes componentes espaciales que se captan en una estación sísmica, posiblemente esto mejore el desempeño en las métricas obtenidas por las redes neuronales ya que se estaría exponiendo a la red neuronal a todas las interacciones de las ondas sísmicas con sus medios de propagación en las tres dimensiones espaciales.
- Formar una base de sismogramas catalogados por el tipo de falla tectónica (límites convergentes, divergentes o transformantes, y si el terremoto fue interplaca ó intraplaca) que originó el sismo del cual fueron obtenidos y analizar estos sismogramas con la red neuronal *BiGRU-Attention* optimizada con el fin de encontrar alguna peculiaridad a destacar y de las mismas conjeturar las razones o motivos que generaron dichas peculiaridades.



# Apéndice A

---

## Acceso a la base de datos del SSN

### A.1. Procedimiento

Para acceder a la base de datos de la red de estaciones sísmicas de banda ancha del Servicio Sismológico Nacional (SSN) es necesario enviar un correo a la siguiente dirección electrónica: `snndata@sismologico.unam.mx`. El correo se redacta en inglés, explicando el porqué se desea acceder a la base de datos del SSN [29] y que uso se dará a dicha información. Si ellos consideran que el uso que se dará a la información es adecuado responderán con un correo que contiene las ligas para descargar un programa que actúa como interfaz con la base de datos del SSN. Dicho programa está disponible para los sistemas operativos Linux o Mac (en el caso particular de este trabajo de tesis se usó el sistema operativo Ubuntu-Linux<sup>1</sup>). En el mismo correo se envía una liga mostrando como es que se debe de citar y agradecer por el uso de la base de datos de SSN.

Una vez que se cuenta con el programa que da acceso a la base de datos, llamado *SSNstp\_linux*, se llevan a cabo los siguientes pasos:

- Se da permisos de ejecución al programa *SSNstp\_linux* con alguno de los siguientes comandos desde la línea de comandos de Linux:
  - `sudo chmod 755 program_path/SSNstp_linux`
  - `sudo chmod a+x program_path/SSNstp_linux`

Donde se tiene que el comando `sudo` da permisos de superusuario al introducir la contraseña de usuario y `program_path` es la ubicación del programa.

- Se ejecuta el programa con permisos de superusuario desde la línea de comandos de Linux con el siguiente comando:  
`sudo ./program_path/SSNstp_linux`

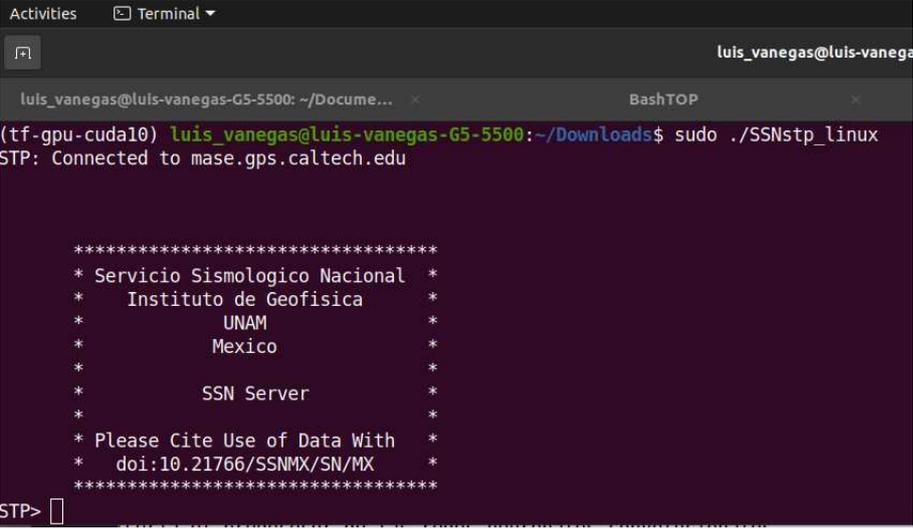
---

<sup>1</sup><https://ubuntu.com>

## Apéndice A. Acceso a la base de datos del SSN

---

- Si el procedimiento anterior se llevó a cabo con éxito se desplegará una pantalla como la que se muestra en la figura A.1.



```
luis_vanegas@luis-vanegas-G5-5500: ~/Downloads$ sudo ./SSNstp_linux
STP: Connected to mase.gps.caltech.edu

*****
* Servicio Sismologico Nacional *
* Instituto de Geofisica *
* UNAM *
* Mexico *
* SSN Server *
* Please Cite Use of Data With *
* doi:10.21766/SSNMX/SN/MX *
*****
STP> █
```

**Figura A.1:** *Menú que se obtiene cuando la conexión con la base de datos del SSN ha sido exitosa.*

No siempre se logra el acceso a la base de datos del SSN en la primera ocasión que se introduce el comando de ejecución del programa, hay que intentarlo hasta que se logre establecer la conexión.

- Una vez dentro de la base de datos del SSN al introducir el comando `HELP` se obtiene la pantalla que se muestra en la figura A.2 en la página 105, la cual describe los comandos más básicos.
- Los comandos que más se usarán en este trabajo de tesis son: `WIN` y `AVAIL`. El primero permitirá realizar la descarga de los sismogramas acorde a las especificaciones que se planteen en sus argumentos, y el segundo permitirá conocer las estaciones sísmicas que estaban disponibles y almacenando información en las fechas especificadas en sus argumentos.
- Los argumentos para el comando `WIN` son los siguientes:

`WIN net sta chan time_on time_off`, donde:

- `net`. Hace referencia al código de la red de estaciones sísmicas, por ejemplo, `IG` para el caso de la red de banda ancha de sismógrafos del SSN.

```

luis_vanegas@luis-vanegas-G5-5500: ~/Docume... x BashTOP x
STP> help

      Command and Help Summary

WIN      - get time window of data
ARRAY    - prefix station-number to filenames
STA      - station info
CHAN     - channel info
SAC      - output in SAC format
GAIN     - yes/no correction for station gain
DATE     - summary of how to specify dates and times
INPUT    - redirect source of commands to a file
OUTPUT   - make a shadow copy of the session in a file
STATUS   - status of the client and server
VERBOSE  - describe in detail the transactions
AVAIL    - list data streams that are available at a particular time
SET      - set some default parameters in STP

HELP Command:
  HELP [TOPIC]
      where topic is one of the entries above
UNIX shell escape:
  Any line starting with '!' will be executed as a UNIX
  24 -- return -->

```

Figura A.2: Ayuda que se obtiene cuando se introduce el comando `HELP`.

- **sta.** Codificación que requiere de cuatro letras y hace referencia a la estación sísmica, por ejemplo, *IGZA* que está ubicada en Zacatecas, Zacatecas, para la red de estaciones sísmicas *IG*. El argumento *sta* se puede remplazar por el símbolo `%` cuando se desea que se descargue información de todas las estaciones disponibles.
- **chan.** Codificación que requiere de tres letras y hace referencia a la componente espacial para la cual se desea obtener sismogramas. Las opciones disponibles son: HHE (Este-Oeste), HHN (Norte-Sur) y HHZ (Vertical). Cuando se desea descargar información en las tres componentes espaciales se especifica de la siguiente forma: `HH_`. El guión bajo es una forma de indicar que se requiere la información en las tres componentes espaciales.
- **time\_on.** Hace referencia a la fecha y hora a partir de la cual se desea empezar a descargar información, existen varias formas de expresar el tiempo de inicio, las sintaxis disponibles son las siguientes:
  1. `yyyymmddhhmmss.sss`  
Donde claramente, las cuatro 'y' corresponden al año, las dos m al mes, las dos d al día, los dos h a la hora, las dos m a los minutos, las dos primeras s a los segundos y las últimas tres s a los milisegundos.

## Apéndice A. Acceso a la base de datos del SSN

---

2. `yyyy/mm/dd,hh:mm:ss.sss`

3. `yyyy,mm,dd,hh:mm:ss.sss`

4. `yyyy/jjj,hh:mm:ss.sss`

Donde las tres 'j' representan el día correspondiente al calendario juliano.

5. `yyyy,jjj,hh:mm:ss.sss`

- `time_off`. Hace referencia a la fecha y hora a partir de la cual se desea interrumpir la descarga de información.

Cabe aclarar que las fechas y horas van sincronizadas con el *Tiempo Universal Coordinado* (UTC). Al ejecutar el comando `WIN` con sus respectivos argumentos se descargarán archivos de cada una de las estaciones sísmicas disponibles en el intervalo de tiempo especificado, y la codificación con la cual se descargarán los archivos es la siguiente:

```
FECHA(aaaammddhhmmss).RED(IG).ESTACIÓN(XX).CANAL(HHX).sac
```

- Los argumentos para el comando `AVAIL` son los siguientes:

`AVAIL time_on`

El resultado de este comando será las estaciones que en esa fecha y hora estaban grabando datos respectivos al movimiento del suelo.

Para abandonar el programa que conecta a la base de datos del SSN se escribe el comando `QUIT`.

## Apéndice B

# Acceso a la base de datos de IRIS

### B.1. Procedimiento

Para acceder a la base de datos de IRIS (*Incorporated Research Institutions for Seismology*) se ingresa la siguiente url: [https://ds.iris.edu/wilber3/find\\_event](https://ds.iris.edu/wilber3/find_event) en algún buscador, en esta url se obtendrá la pantalla que se observa en la figura B.1.

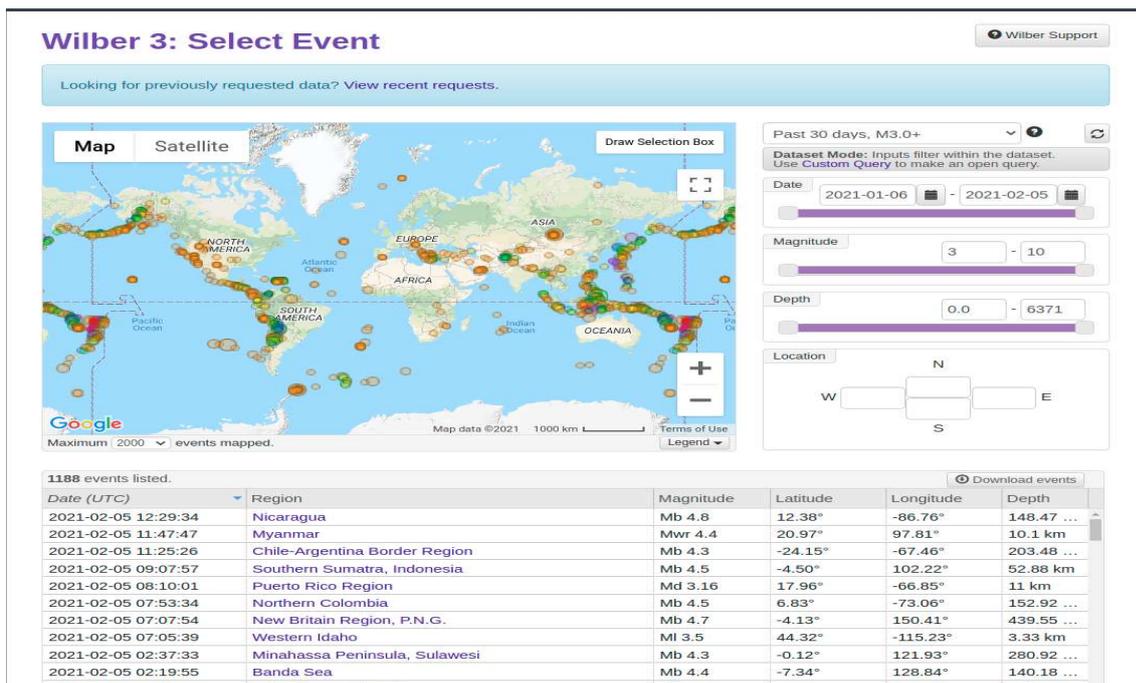


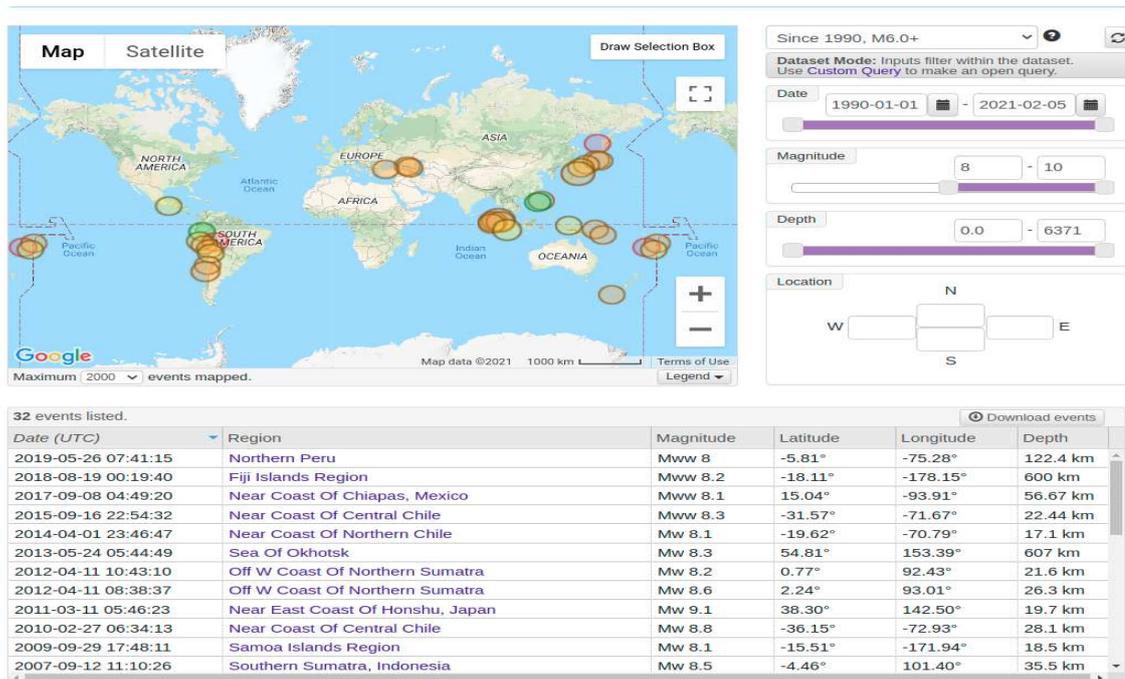
Figura B.1: Pantalla de inicio de la página de IRIS para descargar sismogramas de eventos sísmicos.

En la sección al fondo de la figura B.1 se observan los últimos eventos sísmicos al momento de visitar la página listados en una tabla, el orden de aparición es del más reciente al más antiguo. De izquierda a derecha aparecen las siguientes

## Apéndice B. Acceso a la base de datos de IRIS

columnas informativas: fecha, región, magnitud, latitud, longitud y profundidad del evento sísmico. En la parte superior izquierda se observa un mapa mundial con la ubicación de los últimos eventos sísmicos representados con círculos de diferentes colores y tamaños; la selección de los colores de los círculos da información acerca de la profundidad del hipocentro de los sismos y el tamaño de los círculos da información de la magnitud del evento sísmico. En la parte superior derecha se tienen filtros de búsqueda de eventos sísmicos, los filtros son: intervalo de tiempo, magnitud, profundidad y localización espacial (Norte, Sur, Este u Oeste).

Por ejemplo, se puede realizar una búsqueda de sismos con magnitudes mayores o iguales a 8.0 en un intervalo de tiempo que va desde enero 1 de 1990 hasta febrero 5 de 2021. En la figura B.2 se observan los resultados de dicha búsqueda. En la parte inferior izquierda de la imagen, antes de que la tabla de inicio, se observa que se encontraron 32 eventos sísmicos que cumplen con los filtros de búsqueda. En la parte superior izquierda se observa la localización de dichos sismos en un mapa mundial.



**Figura B.2:** Búsqueda de sismos con magnitud mayor o igual 8 en un intervalo de tiempo que va desde enero 1 de 1990 hasta febrero 5 de 2021.

Ahora, si se decidiera obtener sismogramas de distintas redes sismológicas, por ejemplo, del sismo que se encuentra hasta el fondo de la tabla en la imagen B.2, sismo ocurrido el 12 de septiembre de 2007 en la región sur de Sumatra, Indonesia;

sismo de magnitud 8.5 con una profundidad de 35.5 km. Al dar clic en dicho evento sísmico aparecerá la pantalla que se muestra en la figura B.3, se puede apreciar en la parte superior derecha los filtros de selección referentes a la red sismológica y a los canales o componentes espaciales, en este ejemplo se seleccionó la red global sismológica de estaciones de banda ancha y se seleccionó obtener sismogramas en las tres componentes espaciales; por supuesto, hay muchas más opciones para seleccionar que red sismológica se elegirá, se pueden elegir múltiples redes sismológicas a la vez. En la parte superior izquierda se tiene un mapa mundial con las estaciones sísmicas que captaron dicho evento sísmico, en color rojo las estaciones sísmicas más lejanas al epicentro del sismo y en color verde las estaciones sísmicas más cercanas al epicentro del sismo, en un círculo amarillo se observa el epicentro del evento sísmico. En la parte inferior de la imagen B.3 se observa las estaciones sísmicas con datos disponibles para descarga del evento sísmico, en total, para este ejemplo, hay 144 estaciones sísmicas con datos disponibles para descarga, las cuales están seleccionadas por medio de una paloma verde, si se deseará dejar fuera alguna o algunas estaciones sísmicas solo basta con dar clic en las palomas verdes y se segregará esa estación sísmica de los archivos de descarga.

Si se desea descargar los sismogramas seleccionados de las 144 estaciones sísmicas disponibles se da clic en el recuadro morado que dice *Request Data* y aparecerá el menú de selección que se muestra en la figura B.4.

**2007-09-12 MW8.5 Southern Sumatra, Indonesia**

Latitude	Longitude	Date	Depth	Magnitude	Description	Related Pages
4.4637° S	101.3959° E	2007-09-12 11:10:26 UTC	35.5 km	MW8.5	Southern Sumatra, Indonesia	IRIS Event Page

The map below shows stations operational during this event, filtered by the criteria in the form to the right.

**Request Only**

Networks:

Channels:

Distance Range: 0 - 180

Azimuth Range: -180 - 180  Invert

**Actions**

Show Record Section

Use the checkboxes below to add/remove individual stations from your request.

Station	Source	Net	Latitude	Longitude	Distance	Azimuth	Elevation	Station Name
<input checked="" type="checkbox"/> COCO	IRISDMC	II	-12.19°	96.83°	8.95°	-150.01°	1 m	West Island, Cocos (Keeling) Islands
<input checked="" type="checkbox"/> KAPI	IRISDMC	II	-5.01°	119.75°	18.30°	92.47°	300 m	Kappang, Sulawesi, Indonesia
<input checked="" type="checkbox"/> CHTO	IRISDMC	IU	18.81°	98.94°	23.40°	-5.85°	420 m	Chiang Mai, Thailand
<input checked="" type="checkbox"/> PALK	IRISDMC	II	7.27°	80.70°	23.75°	-60.49°	460 m	Pallekele, Sri Lanka
<input checked="" type="checkbox"/> MBWA	IRISDMC	IU	-21.16°	119.73°	24.41°	134.76°	190 m	Marble Bar, Western Australia
<input checked="" type="checkbox"/> QJZ	IRISDMC	IC	19.03°	109.84°	24.92°	19.25°	240 m	Qiongzong, Hainan Province, China
<input checked="" type="checkbox"/> DAV	IRISDMC	IU	7.07°	125.58°	26.75°	64.59°	150 m	Davao, Philippines
<input checked="" type="checkbox"/> DGAR	IRISDMC	II	-7.41°	72.45°	28.93°	-97.25°	1 m	Diego Garcia, Chagos Islands, Indian Ocean
<input checked="" type="checkbox"/> KMI	IRISDMC	IC	25.12°	102.74°	29.62°	2.46°	1975 m	Kunming, Yunnan Province, China
<input checked="" type="checkbox"/> NWAQ	IRISDMC	IU	-32.93°	117.24°	32.08°	154.44°	380 m	Narrogin, Australia

Figura B.3: Pantalla que aparece al seleccionar un sismo.

## Apéndice B. Acceso a la base de datos de IRIS

2007-09-12 MW8.5 Southern Sumatra, Indones

Latitude: 4.4637° S Longitude: 101.3959° E

The map below shows stations

Map Satellite

Show up to 5000 stations

Selected 144 out of 144 stations

Station	Source
<input checked="" type="checkbox"/> COCO	IRISDM
<input checked="" type="checkbox"/> KAPI	IRISDM
<input checked="" type="checkbox"/> CHTO	IRISDM
<input checked="" type="checkbox"/> PALK	IRISDM
<input checked="" type="checkbox"/> MBWA	IRISDM
<input checked="" type="checkbox"/> OIZ	IRISDM
<input checked="" type="checkbox"/> DAV	IRISDM
<input checked="" type="checkbox"/> DGAR	IRISDM
<input checked="" type="checkbox"/> KMI	IRISDM
<input checked="" type="checkbox"/> NWAO	IRISDM

### Request Data

Requesting 682 channels from 144 stations. Estimated size: 45 megabytes\*  
\* Rough estimate only; actual download size may be up to 2x this value.

Required fields are listed in bold.

**Time Range**

Starting: 1 minutes before P arrival

Ending: 10 minutes after P arrival

**Data Output**

Output Format: SAC binary (little-endian)

This will produce a separate file for each channel. These files can be downloaded individually, or may be bundled for easier downloading.

Bundle As: gzipped tar archive

The full SEED format has been retired. Please see this newsletter article for more details.

**Request Information**

Your name and the request label are used to uniquely identify your request, so you cannot reuse a label you used before.

Your Name: [input field]

Request Label: 2007-09-12 MW8.5 Southern Sumatra, Indones

If you provide an email address, you will be notified by email when your data is ready.

Email Notification: [input field]

Save these settings as defaults

Figura B.4: Pantalla que aparece al seleccionar 'Request Data'.

En la figura B.4, en la parte superior se da información acerca del tamaño de la descarga y del total de sismogramas que se descargarán. Se tiene un filtro de selección para especificar la ventana de tiempo de los sismogramas a descargar, la ventana de tiempo depende de la llegada de las onda P o ondas S, según se seleccione, en este ejemplo se seleccionó que la ventana de tiempo abriera un minuto antes de la llegada de las ondas P y cerrará diez minutos después de la llegada de las ondas P. Se puede elegir entre diferentes formatos de salida, en este ejemplo se seleccionó que el formato de salida fuera `.sac`, también se puede seleccionar cómo se presentarán los archivos descargados, si se presentan individualmente, o se agrupan en un archivo `tar`, o en un archivo `tar` comprimido. Por último, se pide información personal, nombre y correo para enviar la liga de descarga de los archivos.

La información difícilmente llega inmediatamente después de realizar la solicitud de descarga, incluso llega a tardar días, o a veces no es exitosa. Se recomienda que el correo que se introduzca sea de una institución educativa oficial para que la respuesta sea más breve. Es posible darle seguimiento al progreso de la solicitud de descarga. En la figura B.5 se observa un enlace en un recuadro gris que dice: *Track request progress*, al dar clic en él se puede seguir el progreso de la solicitud de descarga.

En la figura B.6 se aprecia la pantalla que aparece cuando el enlace de descarga está listo, ya solo se da clic en *Download data* para finalizar la descarga.

## B.1. Procedimiento

2007-09-12 MW8.5 Southern Sumatra, Indonesia

Latitude 4.4637° S Longitude 101.3959° E

The map below shows station locations

**Data Request Submitted**

Your data request was successfully received and is awaiting processing.

[Track request progress](#) (opens a new window)

[Return to Station List](#)

Map Satellite

Distance Range 0 - 180

Azimuth Range -180 - 180  Invert

**Actions**

[Show Record Section](#) [Request Data](#)

Show up to 5000 stations.

Use the checkboxes below to add/remove individual stations from your request.

Selected 144 out of 144 stations. Select All None One station every

Station	Source	Net	Latitude	Longitude	Distance	Azimuth	Elevation	Station Name
<input checked="" type="checkbox"/> COCO	IRISDMC	II	-12.19°	96.83°	8.95°	-150.01°	1 m	West Island, Cocos (Keeling) Islands
<input checked="" type="checkbox"/> KAPI	IRISDMC	II	-5.01°	119.75°	18.30°	92.47°	300 m	Kappang, Sulawesi, Indonesia
<input checked="" type="checkbox"/> CHTO	IRISDMC	IU	18.81°	98.94°	23.40°	-5.85°	420 m	Chiang Mai, Thailand
<input checked="" type="checkbox"/> PALK	IRISDMC	II	7.27°	80.70°	23.75°	-60.49°	460 m	Pallekele, Sri Lanka
<input checked="" type="checkbox"/> MBWA	IRISDMC	IU	-21.16°	119.73°	24.41°	134.76°	190 m	Marble Bar, Western Australia
<input checked="" type="checkbox"/> QIZ	IRISDMC	IC	19.03°	109.84°	24.92°	19.25°	240 m	Qiongzong, Hainan Province, China
<input checked="" type="checkbox"/> DAV	IRISDMC	IU	7.07°	125.58°	26.75°	64.59°	150 m	Davao, Philippines
<input checked="" type="checkbox"/> DGAR	IRISDMC	II	-7.41°	72.45°	28.93°	-97.25°	1 m	Diego Garcia, Chagos Islands, Indian Ocean
<input checked="" type="checkbox"/> KMI	IRISDMC	IC	25.12°	102.74°	29.62°	2.46°	1975 m	Kunming, Yunnan Province, China
<input checked="" type="checkbox"/> NWAO	IRISDMC	IU	-32.93°	117.24°	32.08°	154.44°	380 m	Narrogin, Australia

Figura B.5: Pantalla que aparece después de dar clic en Request Data.

**Wilber 3: Track Data Request** [Wilber Support](#)

Luis Alfonso: 2007-09-12 MW8.5 Southern Sumatra, Indonesia

You may bookmark this page and visit it later to monitor or download your data.

This page (and any requested data) will expire in a few days.

Event

Date	2007-09-12 11:10:26 UTC	Description	Southern Sumatra, Indonesia
Latitude	4.5° S	Magnitude	MW8.5
Longitude	101.4° E	Catalog	ISC
Depth	36	Contributor	ISC

[View this event in Wilber](#)

Requested Data

**Success** [Download data](#)

Completed 8 of 8 (100%)

- Done Prepare for processing
- Done Build selection file
- Done Fetch timeseries data
- Done Collect additional SACPZ files
- Done Extract SAC files
- Done Deliver files to pickup area
- Done Log data statistics
- Done Clean up

[View full log](#)

Data	682 channels
Output Format	Sacbl (tgz archive)
Submitted	2021-02-05 16:44:41 (2 minutes ago)
Completed	2021-02-05 16:46:42 (took 2 minutes)

Figura B.6: Pantalla que aparece al estar lista la descarga.



---

## Bibliografía

- [1] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate, 2015.
- [2] Vladimir Braverman. *Sliding Window Algorithms*, pages 2006–2011. 01 2016. doi: 10.1007/978-1-4939-2864-4\_797.
- [3] Sneha Chaudhari, Gungor Polatkan, Rohan Ramanath, and Varun Mithal. An attentive survey of attention models. *CoRR*, abs/1904.02874, 2019. URL <http://arxiv.org/abs/1904.02874>.
- [4] Junyoung Chung, Çağlar Gülçehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *CoRR*, abs/1412.3555, 2014. URL <http://arxiv.org/abs/1412.3555>.
- [5] Chollet. F. *Deep Learning with Python*. Manning Publication Co., USA, 1<sup>st</sup> edition, 2017.
- [6] Rosenblatt. F. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6):386–408, 1958. doi: 10.1037/h0042519.
- [7] K. Fukushima and S. Miyake. Neocognitron: A new algorithm for pattern recognition tolerant of deformations and shifts in position. 15:455–469, 1982. doi: 10.1016/0031-3203(82)90024-3.
- [8] Sepp. Hochreiter. and Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.
- [9] John H. Holland. *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence*. MIT Press, Cambridge, MA, USA, 1992. ISBN 0262082136.
- [10] Sutskever I., Vinyals O., and Quoc V. Sequence to sequence learning with neural networks. December 2014.

## Bibliografía

---

- [11] Aya Abdelsalam Ismail, Timothy Wood, and Héctor Corrada Bravo. Improving long-horizon forecasts with expectation-biased lstm networks. 2018.
- [12] Hopfield. J. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences of the United States of America*, 79(8):2554–2558, April 1982.
- [13] Bergstra James and Bengio Yoshua. Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13:281–315, 2012.
- [14] Snoek Jasper, Larochelle Hugo, and Adams Ryan. Practical bayesian optimization of machine learning algorithms. *Advances in Neural Information Processing Systems*, arXiv:1206.2944, 2012.
- [15] Mockus Jonas, Vytautas Tiesis, and Zilinskas Antanas. The application of bayesian methods for seeking the extremum. *Towards Global Optimization*, 2: 117–129, 1978.
- [16] S. Kiranyaz, T. Ince, R. Hamila, and M. Gabbouj. Convolutional neural networks for patient-specific eeg classification. 2015. doi: 10.1109/EMBC.2015.7318926.
- [17] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. pages 1097–1105, 2012. URL <http://papers.nips.cc/paper/4824-imagenet-classification-w%5Cnpapers3://publication/uuid/1ECF396A-CEDA-45CD-9A9F-03344449DA2A>.
- [18] Y. LeCun, B. Boser, J.S. Denker, R. E. Howard, W. Habbaerd, and L. D. Jackel. Handwritten digit recognition with back propagation network. pages 396–404, 1990. doi: 10.1111/dsu.12130.
- [19] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. doi: 10.1109/5.726791.
- [20] Minh-Thang Luong, Hieu Pham, and Christopher D. Manning. Effective approaches to attention-based neural machine translation. *CoRR*, abs/1508.04025, 2015. URL <http://arxiv.org/abs/1508.04025>.
- [21] Sazli. M. A brief review of feed-forward neural networks. *Communications, Faculty of Science, University of Ankara*, 50:11–17, 01 2006. doi: 10.1501/0003168.

- 
- [22] Turing. A. M. Computing machinery and intelligence. *Mind*, LIX:433–460, October, 1950.
- [23] G. Hinton N. Srivastaga, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. 2004. doi: 10.1016/j.micromeso.2003.09.025.
- [24] Russell. J. & Norving. P. *Artificial Intelligence: A Modern Approach*. Prentice Hall, 3<sup>rd</sup> edition, 2010.
- [25] Anibal Pedraza, Jaime Gallego, Samuel Lopez, Lucia Gonzalez, Arvydas Laurinavicius, and Gloria Bueno. *Glomerulus Classification with Convolutional Neural Networks*. Springer International Publishing, Cham, 2017. ISBN 978-3-319-60964-5.
- [26] Rumelhart. D. Hinton. G. & Williams. R. Learning representations by back-propagating errors. *Nature*, 323:533–536, October 1986. doi: <https://doi.org/10.1038/323533a0>.
- [27] Haykin. S. *Neural Networks, A Comprehensive foundation*. Prentice Hall, 1999. 2<sup>nd</sup> Edition.
- [28] Stein. S and Wyssession M. *Introduction to Seismology, Earthquakes, and Earth Structure*. Blackwell Publishing, 1<sup>st</sup> edition, 2003.
- [29] SSN. Servicio sismológico nacional, instituto de geofísica, universidad nacional autónoma de méxico. 2021. doi: <https://doi.org/10.21766/SSNMX/SN/MX>. URL <http://www.ssn.unam.mx>.
- [30] Melek Tendürüs, Gert Jan van Wijngaarden, and Henk Kars. *Long-term effect of seismic activities on archaeological remains: A test study from Zakynthos, Greece*. Geological Society of America, 10 2010. ISBN 9780813724713. doi: 10.1130/2010.2471(13). URL [https://doi.org/10.1130/2010.2471\(13\)](https://doi.org/10.1130/2010.2471(13)).
- [31] KyungHyun Cho & Bart van Merriënboer & Dzmitry Bahdanau & Yoshua Bengio. On the properties of neural machine translation: Encoder-decoder approaches. *CoRR*, abs/1409.1259, 2014. URL <https://dblp.org/rec/journals/corr/ChoMBB14.bib>.
- [32] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *CoRR*, abs/1706.03762, 2017. URL <http://arxiv.org/abs/1706.03762>.
- [33] McCulloch. W. S. & Pitts. W. A logical calculus of the ideas immanent in neurons activity. *Bull. Math. Biophys*, 5:115–133, 1943.

## Bibliografía

---

- [34] Pascanu R; Mikolov T; & Bengio Y. Understanding the exploding gradient problem. *CoRR*, abs/1211.5063, 2012. URL <http://arxiv.org/abs/1211.5063>.
- [35] Rui Zhao, Dongzhe Wang, Ruqiang Yan, Kezhi Mao, Fei Shen, and Jinjiang Wang. Machine health monitoring using local feature-based gated recurrent unit networks. *IEEE Transactions on Industrial Electronics*, PP:1–1, 07 2017. doi: 10.1109/TIE.2017.2733438.
- [36] Laura M. Zingaretti, Salvador Alejandro Gezan, Luis Felipe V. Ferrão, Luis F. Osorio, Amparo Monfort, Patricio R. Muñoz, Vance M. Whitaker, and Miguel Pérez-Enciso. Exploring deep learning for complex trait genomic prediction in polyploid outcrossing species. *Frontiers in Plant Science*, 11:25, 2020. ISSN 1664-462X. doi: 10.3389/fpls.2020.00025. URL <https://www.frontiersin.org/article/10.3389/fpls.2020.00025>.