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

# "*ENGINEERED FEATURES*" PARA LA ESTIMACIÓN DE MAGNITUD DE EVENTOS SÍSMICOS

MEMORIA PARA OPTAR AL TÍTULO DE INGENIERO CIVIL ELÉCTRICO

OSCAR ELADIO VÁSQUEZ PINO

PROFESOR GUÍA:
Néstor Becerra Yoma

MIEMBROS DE LA COMISIÓN:
Aarón Cofré Henriquez
Víctor Poblete Ramirez

SANTIAGO DE CHILE
2023

RESUMEN DE LA MEMORIA PARA OPTAR AL
TÍTULO DE INGENIERO CIVIL ELÉCTRICO
POR: OSCAR ELADIO VÁSQUEZ PINO
FECHA: 2023
PROF. GUÍA: NÉSTOR BECERRA YOMA

# "*ENGINEERED FEATURES*" PARA LA ESTIMACIÓN DE MAGNITUD DE EVENTOS SÍSMICOS

La estimación de magnitud de eventos sísmicos es una tarea de vital importancia que debe realizarse con precisión y en un tiempo acotado de tiempo, ya que, para grandes eventos la rápida respuesta es crucial en casos de alertas de Tsunamis y así minimizar los costos económicos y en vidas.

En este trabajo de memoria se evalúa el aporte de "Engineered Features" para la estimación de magnitud de eventos sísmicos. La arquitectura utilizada está basada en una red recurrente LSTM Bi-Direccional alimentada con features temporales. Luego la salida de la celda es concatena con los features globales e ingresados en una capa *Fully Connected* de salida.

Los resultados en conjunto con la velocidad de testeo de la red neuronal, permiten concluir que implementar un modelo basado en deep learning sería una buena herramienta para un sistema de alerta temprana de terremoto (EWW) y alerta temprana de tsunamis (TEW) en territorio Chileno, lo cual facilitaría bastante el constante monitoreo realizado por el Centro Sismológico Nacional (CSN).

La mejor configuración se obtuvo para el bloque LSTM Bi-Direccional concatenado con el feature global de ID-Estación en su expresión de enteros. Los mejores resultados por rango de magnitud son 3,93 % para M>4; 7,26 % para M<4 y 6,48 % para todas las magnitudes. Estos errores de estimación son totalmente aceptables como valores preliminares de magnitud y se obtienen en el orden de segundos, a diferencia de los métodos actuales que operan en el orden de minutos.

*Dedicado a la mejor mujer*
*del mundo, mi madre.*

***Oscar***

# Agradecimientos

Primero que todo me gustaría agradecer a toda mi familia, que somos caleta. A mis hermanas por enseñarme a ser metódico y perseverante, a mi tía Ercilia por alimentarme a lo largo de toda mi vida y a mi novia Ana por acompañarme en este largo proceso de titulación.

También quiero agradecer a mis amigos del colegio, a sus familias y a los que se fueron sumando. Son caleta de años de amistad y seguiremos creciendo juntos. Ojalá algún día levantemos una copita poh cabros. Quiero agradecer a mi grupo de *Pastoral Beauchef*, a quienes conocí desde que ingresé a esta hermosa Escuela y que sin duda fueron fundamentales en todo este proceso.

Me gustaría agradecer a mi Profe Guía Néstor Becerra, quien me dio la oportunidad de entrar al Laboratorio de Procesamiento y Transmisión de Voz (LPTV) aun cuando mis conocimientos en programación eran bajisimos. Sin duda vio algo en mi y confió en mis capacidades para emprender en este proyecto Fondef. Hoy puedo decir que luego de todo este proceso he agarrado el gusto por la programación, el procesamiento de señales y el deep learning. En el LPTV encontré un gran equipo y ambiente de trabajo, totalmente agradecido de los chicos Aaron, Cata, Marcelo, Nico, Edu, Rodrigo y Simón.

Por ultimo, quiero agradecer a las tías de Bienestar Estudiantil y a todo el equipo de apoyo psicológico de la Universidad, quienes me ayudaron y se preocuparon un montón en los momentos más difíciles que me ha tocado vivir y hoy puedo decir con orgullo que me levanté.

*Muchas veces pensé y cranee como escribiría estas palabras. Yo creo que sin duda ha sido la parte más difícil al escribir esta memoria. Porque pucha que me haz hecho falta querida madre. Estas palabras son para ti hacia donde sea que estés. Estuve, estoy y estaré eternamente agradecido por todas y cada una de las cosas que hiciste para forjar a la persona que soy hoy en día. Admiro tu esfuerzo, dedicación y por sobre todo amor que siempre nos entregaste a mis hermanas y a mi. Tu legado es único, fuiste, eres y seras una leyenda Viviana Shora Vivi. Gracias y eternas gracias for everything. Te amo.*

*Oscar Eladio*

# Tabla de Contenido

# Índice de Tablas

# Índice de Ilustraciones

# Capítulo 1

# Introducción

Un terremoto es un fenómeno natural que se produce cuando se libera energía acumulada en la corteza terrestre, generando vibraciones y movimientos bruscos en la superficie de la Tierra. Estos eventos, también conocidos como sismos, son causados por la liberación repentina de energía acumulada en las fallas geológicas, que son zonas de fractura en la corteza terrestre. Los terremotos pueden variar en magnitud y duración, desde temblores imperceptibles para la comunidad hasta un evento de carácter global como lo fue el Terremoto de Valdivia Mw 9.5 de 1960, el cual destruyó gran parte del territorio Chileno y provocó un Tsunami que golpeo las costas del Océano Pacifico a lo largo del mundo [1, 2]. Es por esto, que una rápida lectura, detección y clasificación de estos eventos es de suma importancia para poder tomar medidas de acción temprana, en pro de minimizar los costos humanos y económicos de la zona afectada.

La alerta temprana de Terremotos (*EEW, Earthquake Early Warning*) es una estrategia que ha cobrado bastante importancia en las ultimas décadas, donde el monitoreo en tiempo real permite estimar parámetros de las primeras ondas de llegada de un sismo y predecir si el movimiento en curso tiene las características de un evento importante [3, 4]. Si ese es el caso, se procede a alertar a las autoridades, comunidades y personas de las zonas aledañas. Las EEWs se comenzaron a utilizar en Japón desde la década de 1960 y se han ido implementando en países como México, Turquía, Italia, Rumania, Taiwan, China, EEUU, entre otros. Actualmente se han utilizado técnicas de inteligencia artificial (IA) para mejorar el rendimiento y automatizar los sistemas de alerta. Es por esto que la incorporación de éstos métodos de IA en la estimación de magnitud resulta bastante interesante para la comunidad científica, los cuales se deben ir probando a nivel global y local.

Los métodos de inteligencia artificial son capaces de calcular parámetros relevantes que permiten hacer una estimación temprana de la magnitud, ubicación y el potencial peligro del Terremoto en curso, de forma mucho más rápida que un monitoreo manual. Lo anterior facilita entre otras cosas identificar si el sismo en cuestión tiene el potencial de generar un tsunami [5, 6]. Para que esta tarea cobre sentido, el método implementado tiene que ser capaz de poder entregar una respuesta en el orden de segundos, es decir, antes de que lleguen los trenes de ondas más destructivos a la zona. Esto implica poder realizar la estimación con las ondas de cuerpo p y s de la (o las) estación (es) que detecte (n) primero el movimiento sin la necesidad de esperar el reporte de toda la red Sismológica [7, 8].

Pese al contexto tectónico en el cual se encuentra Chile y la gran cantidad de terremotos que han devastado al país, aún los métodos de alerta temprana y calculo de magnitud contienen procesos que se realizan de forma manual donde los analistas están encargados de identificar y picar las ondas de cuerpo. Esto implica una respuesta mayor que los sistemas automáticos que utilizan inteligencia artificial. Si bien se han realizado algunos estudios en este tema, estos presentan un error de predicción que aún limita su funcionalidad, sobre todo en la presencia de falsas alarmas que pueden provocar pánico en la población.

Dentro de los problemas que se tienen en Chile, es importante mencionar lo extensa que es su línea costera y la ausencia de cobertura azimutal de estaciones sismológicas. Esto dificulta la localización temprana de la fuente de los sismos generados en la fosa oceánica, los cuales además coinciden con ser los Terremotos más peligrosos. En primer lugar porque esta zona se caracteriza por tener Terremotos de alta magnitud debido a la gran extensión de la falla y en segundo lugar son potencialmente destructivos porque son capaces de levantar enormes volúmenes de agua generando un posterior Tsunami. Por ultimo, es importante mencionar la falta de datos a nivel local y mundial de eventos de gran magnitud, ya que, estos suceden con poca frecuencia.

Los "Engineered Features" son variables o atributos creados específicamente para mejorar el rendimiento de los modelos de machine learning. A veces, las características originales no son suficientes para capturar toda la información relevante en los datos, por lo que se crean características adicionales utilizando conocimientos expertos y técnicas de procesamiento de datos. Esto implica operaciones como transformaciones matemáticas, combinación de variables y normalización. El objetivo principal es proporcionar características más descriptivas y relevantes a los modelos, lo que mejora su capacidad para hacer predicciones o estimaciones precisas. Los "Engineered Features" son fundamentales en la construcción de modelos de aprendizaje automático efectivos, ya que permiten capturar relaciones y patrones subyacentes en los datos que no están presentes en las características originales [9].

Los "Engineered Features" aprovechan la experiencia humana y el conocimiento previo del dominio de estudio para extraer información valiosa de los datos y mejorar significativamente el rendimiento de los algoritmos de machine learning. Esta es una parte crucial del proceso y puede marcar la diferencia entre un modelo aceptable y uno excelente. En la memoria se analizan dos tipos de features, features temporales y features globales. Por el lado de los features temporales se experimentó con un feature de energía y otro dependiente del espectro de Fourier. Y en cuanto a los features globales se analizaron el tiempo de arribo S-P, indexación de la estación sismológica y las coordenadas de la estación.

La arquitectura implementada y optimizada en el presente trabajo de memoria, es capaz de entregar una estimación de magnitud de evento sísmica competitiva y aplicable en el contexto geológico nacional actual. Es importante destacar, que con la incorporación de métodos de inteligencia artificial en las ciencias de la Tierra, se podrán realizar estudios y descubrimientos en pro de entender fenómenos naturales complejos como los Terremotos, de los cuales aun hay mucho por investigar.

## 1.1. Objetivos

A continuación, se presenta el objetivo general y los objetivos específicos del trabajo de memoria.

### 1.1.1. Objetivo General

Estimar la magnitud de eventos sísmicos usando una estación y datos limitados de entrenamiento.

### 1.1.2. Objetivos Específicos

- Crear una base de eventos sísmicos para esta y futuras investigaciones en el marco de estimación de magnitud.
- Ganar expertiz en técnicas de deep learning en el marco del problema abordado.
- Evaluar el aporte de los "Engineered Features" cuando hay datos limitados de entrenamiento en el marco del proyecto abordado.

### 1.1.3. Descripción del Laboratorio

El Laboratorio de Procesamiento y Transmisión de Voz (LPTV), perteneciente a la Universidad de Chile, fue creado en el año 2000 por el Profesor Néstor Jorge Becerra Yoma como parte del proyecto "Procesamiento Robusto de Patrones Acústicos para Aplicaciones en telefonía e Internet" y desde su creación ha trabajado y desarrollado proyectos de investigación *CONICYT/FONDECYT* ligados a múltiples áreas de la ciencia.

Las áreas de investigaciones abarcan múltiples disciplinas de la ingeniería y de la ciencia, las cuales se pueden resumir en las siguientes: tecnologías de voz, QoS en internet, COVID, Sismología y usabilidad en ingeniería.

El LPTV ha mantenido colaboración con importantes universidades y centros de investigación: University of Edinburgh, UK; University of Southern California, USA; University of Colorado, USA; entre otras. Además, el Director del LPTV mantiene un rol activo en la comunidad internacional de tecnologías de voz, y es director y uno de lo co-fundadores de ISCA Special Interest Group of Iberian Languages.

En cuanto al proyecto de Sismología en el cual se centra el presente trabajo de memoria, el LPTV ha colaborado con la Universidad de la Frontera y con la Universidad Austral de Chile, cuyos equipos de trabajo son liderados por los profesores Fernando Huenupan y Víctor Poblete, respectivamente.

## 1.2. Estructura de la Memoria

La memoria se divide en cinco Capítulos, los cuales se mencionan a continuación:

- ***Capitulo 1 - Introducción:*** se presenta un prefacio del trabajo, los objetivos y la motivación del estudio.

- ***Capitulo 2 - Marco Teórico***: se describe el contexto geológico de Chile con los distintos tipos de sismos que ocurren en su territorio. Además, se define la escala de calculo de Magnitud de Momento [MW] formulada por Hanks y Kanamori. Luego, para una mejor compresión del modelo se presenta un resumen de Redes Neuronales Artificiales (ANN) y Deep Learning.

- ***Capitulo 3 - Implementación***: se presenta la metodología usada y se detalla la obtención de la base de datos, métricas utilizadas, preprocesamiento, extracción de características y la arquitectura de red DNN implementada.

- ***Capitulo 4 - Discusión de Resultados***: se desarrolla el análisis de los resultados obtenidos con las distintas configuraciones de features globales y temporales propuestas para el modelo. A modo general, se observó que la red entrenada con ambos tipos de features obtuvo mejores resultados que el modelo ejecutado solamente con features temporales.

- ***Capitulo 5 - Conclusiones***: se presentan las principales observaciones del modelo y las conclusiones del estudio realizado. El modelo que obtuvo el mejor resultado corresponde a la celda LSTM concatenada con el feature global de ID-Estación.

# Capítulo 2

# Marco Teórico

En esta sección se desarrollan los conceptos claves para un mejor entendimiento del trabajo de memoria, los cuales están enfocados a explicar el contexto sismológico chileno, Deep Learning y Estado del Arte.

## 2.1. Contexto Geológico Chileno

Chile está ubicado al extremo sur del continente americano, específicamente entre las latitudes $17°29'57''S$ y $56°32'12''S$. A lo largo del territorio nacional interactúan cuatro placas tectónicas: Placa de Nazca, Placa Sudamericana, Placa Antártica y Placa de Scotia. El presente trabajo de memoria se enfocará en la geología y actividad sísmica de la zona Centro y Norte del país. En la figura 2.1 se visualizan los sismos ocurridos en la zona geográfica entre el 12 de Noviembre de 2015 y 12 de Diciembre de 2021. Las magnitudes de los sismos gráficados varían entre 2.5 [MW] y 8.1 [MW].


Figura 2.1: Actividad Sísmica de Chile.

### 2.1.1. Tipos de Sismos

Los sismos son generados por distintos factores dependiendo de la zona donde se ocasiona la fractura, la clasificación de los eventos según su ubicación es la siguiente:

1. Interplaca: esta zona corresponde al limite de placas y es donde ocurre el proceso de subducción. Este lugar destaca por su gran cantidad de sismos y por la alta energía que se libera. En la figura 2.2 se pueden apreciar los sismos interplaca en la zona a), donde se ve una alta densidad de sismos hasta una profundidad aproximada de 50 [km]. Los sismos de alta magnitud pueden generar tsunamis devastadores.

2. Intraplaca Oceánica: corresponden a sismos que ocurren en la Placa de Nazca a profundidades superiores a 50 [km], dominio b). Ocurren principalmente por el esfuerzo de tensión que genera el manto sobre la Placa de Nazca al jalarla al interior de la Tierra.

3. Corticales: corresponde a sismos superficiales generados por fracturas en la placa Sudamericana, dominio c). Se destacan por tener pequeñas profundidades. Las deformaciones generadas por este tipo de contacto dieron origen al alzamiento de la cordillera de los Andes.

4. Outer-Rise: corresponden a los sismos generados detrás de la fosa, dominio d). Son ocasionados por el abombamiento de la Placa de Nazca y por el esfuerzo de flexión generado sobre ella en el proceso de subducción.

5. Sismos por falla transformantes: corresponde a sismos generados por el desplazamiento lateral de un bloque con respecto a otro, lo cual provoca ondas sísmicas. El lugar donde pueden encontrarse este tipo de Terremotos es el extremo sur de Chile, en la interacción transformante entre la Placa Sudamericana y la Placa de Scotia. Este tipo de sismo no está presente en la figura 2.2, ya que, en casi la totalidad del país predominan sismos de subducción.

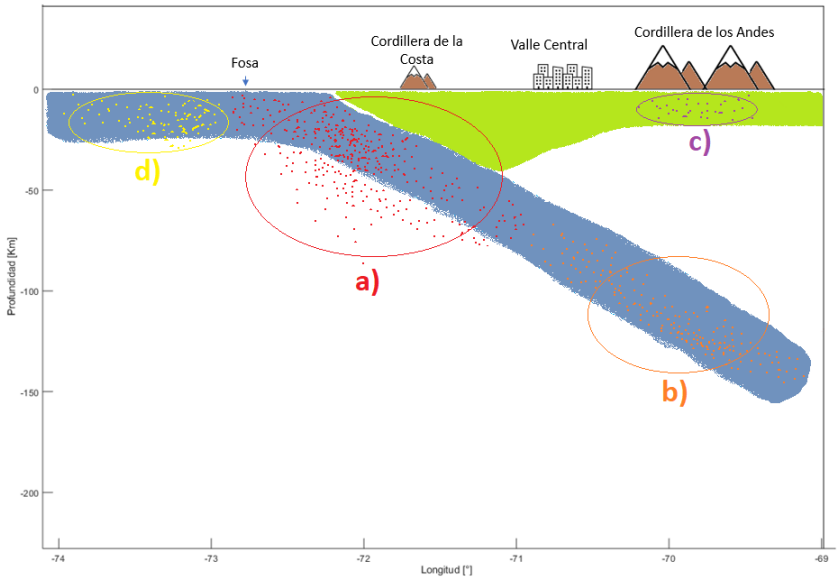


Figura 2.2: Tipos de Sismos en la subducción Placa de Nazca - Placa Sudamericana.

### 2.1.2. Magnitud de Momento [Mw]

La escala de magnitud fue formulada por Hanks y Kanamori en 1977 [10] y revolucionó el campo de la sismología al utilizar el Momento Sísmico como base para su cálculo. Este momento sísmico está relacionado con el área de ruptura y el deslizamiento en la falla. A diferencia de las escalas anteriores, como la Magnitud Local [Ml], Magnitud de ondas superficiales [Ms], Magnitud de ondas de cuerpo [mb] y Magnitud de Coda [mc], la escala Mw no se satura y permite la cuantificación precisa de Megaterremotos. Este avance significativo permitió finalmente el análisis y clasificación de los terremotos más grandes registrados en la historia.

Un caso particular fue el Terremoto de Valdivia 1960, el cual tenía asociada una magnitud de $M_s$=8,3 debido a la saturación mencionada de los antiguos métodos [11]. No fue hasta 1977, que su magnitud pudo ser calculada y comparada con otros Megaterremotos. La formula para calcular la Magnitud de Momento [Mw] se presenta en las ecuaciones 2.1 y 2.2, para momentos sísmicos en $[Nm]$ y $[dyna * cm]$ respectivamente.

$$M_w = \frac{2}{3} \log_{10} (M_{01}) - 6,07 \tag{2.1}$$

$$M_w = \frac{2}{3} \log_{10} (M_{02}) - 10,7 \tag{2.2}$$

- $M_{01}$: $M_0$ en [Nm]
- $M_{02}$: $M_0$ en [dyna*cm]

En la ecuación 2.3 se presenta una de las formas de calcular el Momento Sismico, el cual depende de la rigidez del terreno, el deslizamiento promedio de la falla, el largo de la falla y su ancho.

$$M_0 = \mu DLW \tag{2.3}$$

- $\mu$: rigidez
- D: deslizamiento promedio
- L: largo de falla
- W: ancho de falla

## 2.2. Resumen de Redes Neuronales Artificiales (ANN) y Deep Learning

Las redes neuronales artificiales (ANN) son modelos computacionales inspirados en la estructura y funcionamiento del cerebro humano. Son capaces de aprender de forma autónoma y encontrar patrones en grandes conjuntos de datos, lo que las hace útiles en una amplia variedad de aplicaciones de inteligencia artificial, como tareas de clasificación, regresión, reconocimiento de patrones, predicción de eventos, entre otros.

Las ANN están compuestas por capas de nodos interconectados llamados neuronas o perceptrones. Cada neurona recibe una o más entradas, aplica una función de activación y produce una salida. Las capas de las neuronas se conectan entre sí mediante pesos ajustables que determinan la fuerza de la conexión entre ellas. Durante el entrenamiento, los pesos se ajustan para minimizar el error entre las salidas de la red y los valores deseados.

En [12] se demostró que es posible aproximar cualquier función continua en un espacio de dimensión finita utilizando una red neuronal de una sola capa oculta que contenga una cantidad suficiente de neuronas y con función de activación sigmoide. El resultado se basa en la capacidad de las funciones sigmoide para aproximar funciones continuas arbitrarias de manera precisa y en la propiedad de superposición de las redes neuronales, que permite combinar múltiples funciones de activación para construir una aproximación más precisa.

Es importante destacar que la demostración se refiere a la capacidad teórica de las redes neuronales de una sola capa oculta, pero en la práctica, la precisión de la aproximación puede verse limitada por el número de neuronas de la capa oculta y la complejidad de la función objetivo. Además, la demostración se refiere a redes neuronales con una sola capa oculta, pero en la práctica, las redes neuronales con múltiples capas pueden ser más efectivas para la aproximación de funciones complejas.

El Deep Learning es una rama del aprendizaje automático que se enfoca en entrenar redes neuronales profundas con múltiples capas ocultas, para que estas puedan aprender y representar patrones de datos complejos de manera más efectiva. El objetivo es crear modelos que puedan detectar y clasificar patrones sutiles y abstracciones en grandes conjuntos de datos no estructurados, como imágenes, sonidos o texto. El entrenamiento de las redes neuronales y el Deep Learning requieren grandes cantidades de datos y tiempo de procesamiento, pero han demostrado tener un gran potencial en aplicaciones de reconocimiento de voz, visión artificial, procesamiento del lenguaje natural, robótica, entre otras áreas.

En [13] se demostró que las redes neuronales recurrentes también tienen la propiedad de universalidad, lo que significa que pueden aproximar cualquier función continua en el tiempo. El articulo se basa en que las RNN son capaces de representar cualquier sistema dinámico, que al usar una RNN con suficientes unidades ocultas y el entrenamiento adecuado, la red puede aproximar cualquier función continua a cualquier nivel deseado de precisión. Además, se analiza la relación entre las RNN y otros tipos de redes neuronales, como las redes neuronales feedforward, y destaca las ventajas de usar RNN para ciertos tipos de problemas, como el procesamiento de datos secuenciales.

En [14] se demostró que las redes neuronales convolucionales también tienen la propiedad de universalidad. El documento utiliza pruebas matemáticas para demostrar que las CNN profundas con funciones de activación de ReLU y capas pooling pueden aproximarse a cualquier función continua en un dominio compacto, siempre que la cantidad de filtros y la cantidad de capas sean lo suficientemente grandes. A diferencia de las redes neuronales recurrentes, las CNN suelen utilizarse cuando los datos tienen un caracter espacial, como ocurre en aplicaciones de reconocimiento de imagenes, procesamiento de lenguaje natural, clasificación de objetos, entre otros.

Es importante tener en cuenta que, si bien estas propiedades teóricas son interesantes, en la práctica puede ser difícil entrenar una red neuronal lo suficientemente profunda como para aproximar una función compleja. Además, el número de parámetros necesarios para aproximar una función puede ser muy grande, lo que puede llevar a problemas de sobreajuste y a una mala generalización a nuevos datos. Por lo tanto, es importante tener en cuenta estos desafíos prácticos al utilizar redes neuronales para aproximar funciones complejas o modelar fenomenos naturales como los terremotos.

### 2.2.1. Perceptron

El perceptrón propuesto por Frank Rosenblatt en [15], es considerado como una de las unidades básicas de las redes neuronales artificiales y uno de los primeros modelos de aprendizaje automático que se utilizó en problemas de clasificación binaria. El perceptrón consiste en una red neuronal de una sola capa, que recibe una entrada y produce una salida binaria. La entrada se compone de un vector de características, cada una de las cuales está asociada con un peso. Estos pesos se ajustan durante el proceso de entrenamiento para minimizar el error de predicción.

Se define como una función $f : \mathbb{R}^n \to \mathbb{R}$ compuesta por una entrada $x \in \mathbb{R}^n$, los pesos $w \in \mathbb{R}^n$, un bias $b \in \mathbb{R}^n$ y una función de activación no lineal $\varphi$, la cual realiza deformaciones a la suma ponderada para permitir el aprendizaje en la neurona. En la ecuación 2.4 se presenta la expresión de un perceptrón y su modelo en la figura 2.3.

$$y(x) = \varphi(x^T * w + b) \tag{2.4}$$

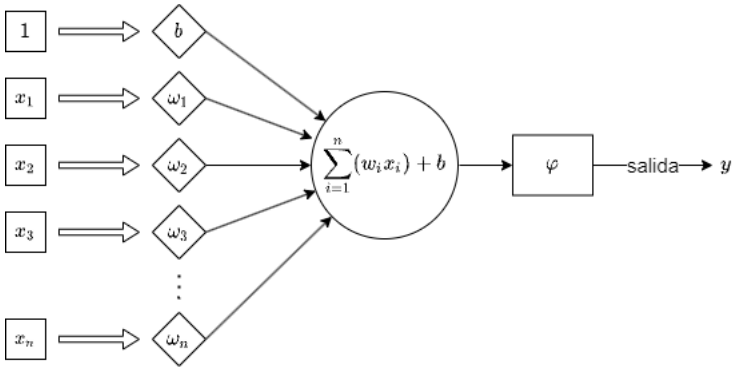


Figura 2.3: Modelo del Perceptrón o Neurona.

### 2.2.2. Aprendizaje Supervisado

El aprendizaje supervisado es uno de los algoritmos más populares de aprendizaje automático, utilizado para entrenar modelos de inteligencia artificial en función de los datos de entrada con sus etiquetas correspondientes. Durante el proceso de entrenamiento, la red va deduciendo patrones e información a partir de los datos que se le proporcionan, lo que le permite aprender a realizar predicciones precisas [16]. La efectividad del modelo se evalúa mediante pruebas realizadas con datos de validación y prueba. Estas pruebas se realizan para medir la capacidad del modelo para generalizar y predecir correctamente el valor correspondiente a cualquier entrada.

El término aprendizaje supervisado se origina en la idea de que un instructor o maestro proporciona un objetivo específico al modelo de aprendizaje automático, y este aprende a realizar predicciones a partir de los ejemplos que se le proporcionan. Una vez que el modelo ha visto suficientes ejemplos, debe ser capaz de generalizar y realizar predicciones precisas en nuevas situaciones.

Existen dos tipos de aprendizaje supervisado: el aprendizaje supervisado por clasificación y el aprendizaje supervisado por regresión. En el aprendizaje supervisado por clasificación, el objetivo es asignar una etiqueta o categoría a una entrada determinada. En cambio, en el aprendizaje supervisado por regresión, el objetivo es predecir un valor numérico para una entrada determinada. Para un mayor detalle visitar el capitulo 5 del libro *Deep Learning* [17].

### 2.2.3. Descenso grandiente estocastico (SGD)

El descenso de gradiente estocástico (SGD) es un algoritmo de optimización popular que se utiliza en el aprendizaje automático para minimizar la función de pérdida de un modelo [18]. Es una variante del algoritmo de descenso de gradiente estándar, el cual en su versión primaria actualiza los parámetros del modelo en función del promedio de los gradientes de la función de pérdida calculada sobre todo el conjunto de datos de entrenamiento. Esto puede ser computacionalmente costoso para grandes cantidades de información, especialmente si el modelo tiene muchos parámetros. Por el contrario, el algoritmo SGD actualiza los parámetros del modelo en función del gradiente de la función de pérdida calculada en un subconjunto aleatorio (o batch) de los datos de entrenamiento para cada iteración [19]. Luego, este gradiente se aplica para actualizar los parámetros del modelo de aprendizaje automático. Este proceso permite que el algoritmo sea mucho más rápido y eficiente.

La principal ventaja de SGD es que puede converger a la solución óptima más rápido que el algoritmo de descenso de gradiente estándar, especialmente cuando se trata de funciones de pérdida ruidosas o no convexas. Esto se debe a que el muestreo aleatorio de los datos introduce estocasticidad en el algoritmo, lo que lo ayuda a escapar de los mínimos locales y explorar el espacio de parámetros con mayor eficacia. SGD también tiene algunas desventajas, como la necesidad de ajustar la tasa de aprendizaje y el tamaño del lote, lo que puede afectar la velocidad de convergencia y la calidad de la solución. Además, SGD puede requerir más iteraciones para converger que el algoritmo de descenso de gradiente estándar, especialmente si el tamaño del lote es pequeño.

#### 2.2.3.1. Optimizador Adam

El optimizador de Adam es un algoritmo de optimización popular que se utiliza en el aprendizaje automático para actualizar los pesos de una red neuronal durante el entrenamiento. Adam combina las ventajas de otros dos algoritmos de optimización: AdaGrad [20] (modificación del gradiente estocástico (SGD) y propagación cuadrática media (RMSprop) [21]. El algoritmo mantiene una estimación actual del primer y segundo momento de los gradientes para ajustar de forma adaptativa la tasa de aprendizaje. El primer momento corresponde a la media de los gradientes y el segundo momento corresponde a la varianza no centrada.

El optimizador Adam es rápido y efectivo, capaz de manejar grandes conjuntos de datos y espacios de parámetros de alta dimensión [22]. También funciona bien con gradientes ruidosos o pequeños, lo que puede ser un desafío para otros algoritmos de optimización. Además, adapta automáticamente la tasa de aprendizaje en función del primer y segundo momento estimados, lo que lo hace más eficiente que los métodos que requieren un ajuste manual de la tasa de aprendizaje.

En general, el optimizador Adam es una herramienta poderosa y ampliamente utilizada en el campo del aprendizaje automático, que ayuda a los investigadores y profesionales a entrenar redes neuronales más precisas y eficientes.

### 2.2.4. Backpropagation

El algoritmo de backpropagation se propuso originalmente en la década de 1970 [23], pero su importancia no se reconoció por completo hasta 1986, cuando Rumelhart, Hinton y Williams lo redescubrieron y popularizaron [24]. Backpropagation es un algoritmo ampliamente utilizado para entrenar redes neuronales artificiales, lo que implica ajustar los pesos y sesgos de una red neuronal para minimizar la diferencia entre las salidas previstas y las salidas reales de la red para una entrada determinada.

La idea básica detrás de backpropagation es propagar el error o la diferencia entre las salidas predichas y reales de la red hacia atrás a través de la red, y usarlo para actualizar los pesos y sesgos de las neuronas en la red. Este proceso se repite para cada par de entrada-salida y, a lo largo de múltiples iteraciones, la red aprende a hacer mejores predicciones ajustando sus pesos y sesgos.

El algoritmo de backpropagation se basa en la regla de la cadena, la cual permite el cálculo de las derivadas parciales de cada uno de los parámetros de una red con respecto al coste (error). Estas derivadas parciales se utilizan luego para actualizar los pesos y sesgos mediante un algoritmo de optimización, como por ejemplo el descenso de gradiente estocástico o variantes. En resumen, la idea básica detrás de este algoritmo es propagar el error o la diferencia entre las salidas predichas y reales de la red y usarlo para ajustar la red.

Backpropagation es un algoritmo potente y eficiente para entrenar redes neuronales y se ha utilizado para lograr un rendimiento de vanguardia en una amplia gama de aplicaciones, incluido el reconocimiento de imágenes, el procesamiento del lenguaje natural y el reconocimiento de voz. Sin embargo, requiere una gran cantidad de datos de entrenamiento etiquetados y puede sufrir un sobreajuste si no se regulariza adecuadamente.

Para entrenar redes neuronales recurrentes (RNN) se utiliza una variación de Backpropagation llamada retropropagación a través del tiempo o *Backpropagation through time* (BPTT) [23], la cual está diseñada para manejar datos secuenciales. A diferencia de las redes neuronales *feedforward*, que procesan las entradas de forma independiente, las RNN mantienen estados internos que les permiten capturar y utilizar dependencias temporales en datos secuenciales. BPTT aborda el desafío de entrenar RNN desplegando las conexiones recurrentes a lo largo del tiempo, transformando efectivamente la red en una red neuronal de avance profundo. Esta red desplegada permite la aplicación del algoritmo de Backpropagation estándar, que se utiliza para calcular gradientes y actualizar los parámetros de la red.

### 2.2.5. Perceptrón Multicapa

El perceptrón multicapa (MLP) es un tipo de red neuronal artificial que se usa ampliamente para el aprendizaje supervisado en tareas de regresión o clasificación [25]. Se compone de múltiples capas de perceptrones, que son los componentes básicos de las redes neuronales. Cada perceptrón toma entradas y produce una salida basada en una suma ponderada de las entradas, que luego se pasa a través de una función de activación.

En unA MLP, la primera capa toma LOS datos de entrada, luego se pasan a través de una o más capas ocultas y, finalmente, la última capa produce la salida. Cada capa oculta puede tener cualquier número de perceptrones, y cada perceptrón de una capa dada está conectado a cada perceptrón de la capa siguiente. Los pesos de estas conexiones se aprenden durante el proceso de entrenamiento, donde la red ajusta sus pesos para minimizar el error entre sus predicciones y las etiquetas verdaderas.

Los MLP son muy flexibles y se pueden utilizar para una variedad de tareas, como la clasificación de imágenes, el reconocimiento de voz, procesamiento del lenguaje natural, ciencias atmosféricas, entre otras [26]. Se ha demostrado que son muy efectivos para modelar relaciones complejas no lineales entre variables de entrada y salida.

Una desventaja de los MLP es que pueden sufrir el problema del sobreajuste, donde la red se vuelve demasiado compleja y comienza a memorizar los datos de entrenamiento en lugar de aprender patrones generalizables. Las técnicas de regularización, como la disminución del peso y la deserción, pueden ayudar a mitigar este problema.

En resumen, las MLP son un tipo poderoso de red neuronal que puede aprender patrones complejos en los datos. Consisten en múltiples capas de perceptrones y se entrenan mediante retropropagación para ajustar sus pesos y minimizar el error. A pesar de algunas limitaciones, se utilizan ampliamente en varios campos y continúan siendo una herramienta importante para el aprendizaje automático.

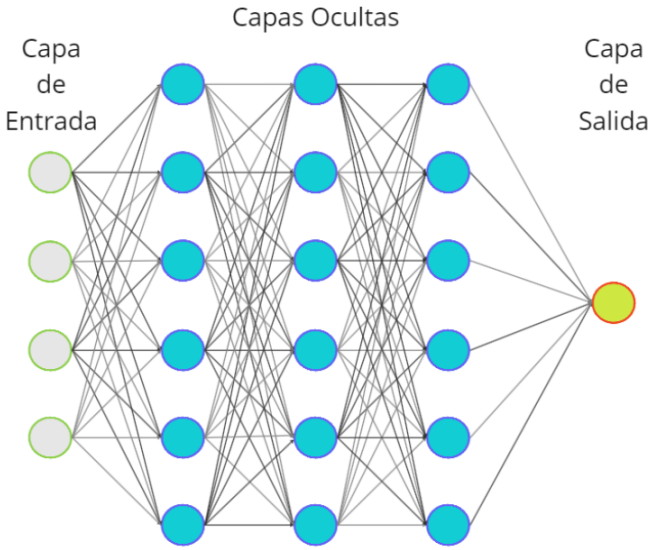


Figura 2.4: Diagrama de una arquitectura Perceptrón Multicapa (MLP).

### 2.2.6. Redes Neuronales Convolucionales (CNN)

Las redes neuronales convolucionales (CNN) son un tipo de red neuronal que se usa comúnmente para el reconocimiento de imágenes y tareas de visión por computadora [27]. Están inspiradas en la organización de la corteza visual de los animales, que tiene células que responden a características específicas del campo visual.

Las CNN funcionan mediante el uso de una serie de capas convolucionales para extraer características de una imagen de entrada. Cada capa convolucional aplica un conjunto de filtros a la imagen de entrada, lo que permite que la red identifique características importantes, como bordes, esquinas y formas. Los filtros se aprenden a través del proceso de entrenamiento y la red ajusta los valores de los filtros para maximizar su capacidad de clasificar correctamente las imágenes.

Después de las capas convolucionales, la red normalmente tiene una o más capas *Fully Connected* que toman la salida de las capas convolucionales y las usan para hacer una predicción final. Durante el entrenamiento, la red recibe un conjunto de imágenes etiquetadas y ajusta los pesos de sus capas para minimizar la diferencia entre sus predicciones y las etiquetas correctas.

Una de las ventajas clave de las CNN es que pueden aprender representaciones jerárquicas de la imagen de entrada. Las primeras capas de la red aprenderán características de bajo nivel, como bordes y esquinas, mientras que las capas posteriores aprenderán características de nivel superior, como formas y objetos. Esto permite que la red sea resistente a las variaciones en la imagen de entrada, como cambios en la iluminación o la orientación.

Las CNN se han utilizado para lograr un rendimiento de vanguardia en una amplia gama de tareas de visión artificial, como la detección de objetos, la segmentación de imágenes y el reconocimiento facial. También se han utilizado en otros dominios, como el procesamiento del lenguaje natural [28, 29] y el reconocimiento de voz, donde han mostrado resultados prometedores [30].

### 2.2.7. Redes Neuronales Recurrentes (RNN)

Las redes neuronales recurrentes son un tipo de arquitectura diseñada para procesar datos secuenciales como series temporales, voz o texto. A diferencia de las redes neuronales tradicionales, las RNN pueden mantener una "memoria" de entradas pasadas y usar esta información para informar el procesamiento de las entradas actuales. El componente básico de un RNN es una "celda", que toma un vector de entrada y produce un vector de salida, así como un vector de estado oculto que se pasa a la siguiente celda. Es decir, cada celda tiene 2 entradas (vector de datos y estado de la celda anterior) y 2 salidas (vector de salida y vector de estado actual).

En la figura 2.5 se presenta a la izquierda como se representa una red neuronal recurrente con una entrada $x_t$, una salida $h_t$ y el bucle de estado destacado en rojo. Es importante mencionar que el bucle se realiza en la celda y es entregada en el siguiente paso, lo cual se puede observar de mejor forma en el diagrama de la red neuronal recurrente desenrollada de la derecha. En rojo se destaca como cada celda le entrega la información de estado a la celda siguiente del instante t+1.

Las RNN se han utilizado con éxito en una amplia gama de aplicaciones, incluido el reconocimiento de voz, la traducción automática, los subtítulos de imágenes e incluso la generación de música. Sin embargo, entrenar RNN puede ser un desafío debido al *vanishing gradient problem*, donde el gradiente utilizado para actualizar los parámetros de la red se vuelve muy pequeño (desvanecimiento) a medida que se propaga hacia atrás en el tiempo hasta el punto de desaparecer, imposibilitando el entrenamiento de la red [31–33]. Para dar solución a este problema mencionado, más adelante se propuso la arquitectura *Long short-term memory* (LSTM) [34], la cual se detalla en la siguiente sección.

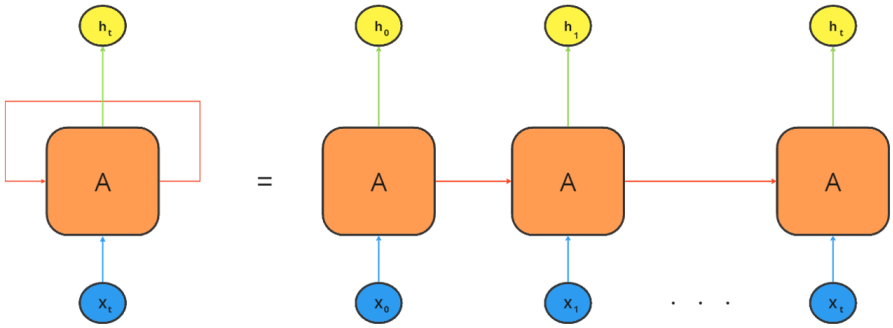


Figura 2.5: Diagrama de las Redes Neuronales Recurrentes (RNN).

### 2.2.8. Red LSTM

La arquitectura Long Short-Tem Memory Layer (LSTM) es un tipo de red neuronal recurrente (RNN) que viene a solucionar los problemas del desvanecimiento y explosión del gradiente, los cuales hasta la fecha no permitían el entrenamiento de redes neuronales profundas o el procesamiento de extensas secuencias de datos. Fue presentada por Hochreiter en 1997 [35] y rápidamente cobró popularidad por su amplia gama de usos. Esta arquitectura se caracteriza por ser capaz de aprender información relevante en el corto plazo y largo plazo. En la figura 2.6 se presenta un diagrama del funcionamiento de una celda LSTM para distintos instantes de tiempo t [36]. Para reforzar este concepto se recomienda visitar la sección de redes neuronales recurrentes.

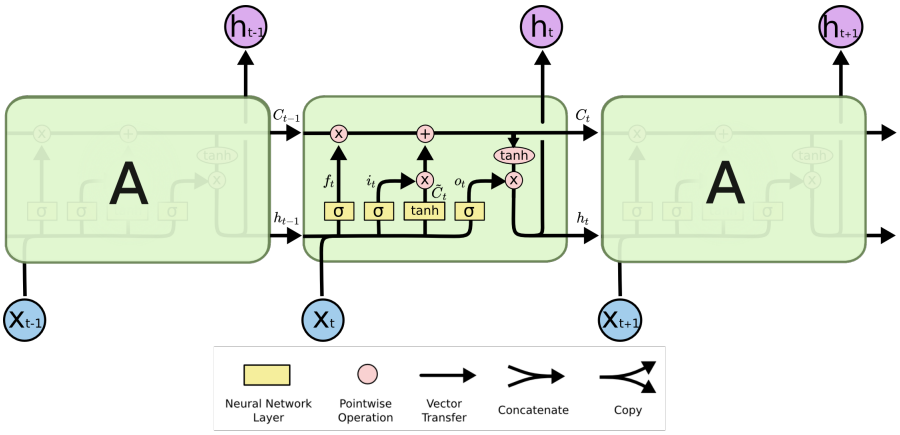


Figura 2.6: Diagrama Celda LSTM

El elemento principal de una celda LSTM es la llamada celda de estado, la cual permite el flujo de la información conservando un tipo de memoria. Esta celda funciona como una cinta transportadora a la cual se le pueden eliminar y agregar datos que provienen del estado anterior y de las compuertas inferiores. En la figura 2.7 se destaca dentro la red la celda de estado por donde viaja la información desde el instante $c_{t-1}$ a $c_t$. Además, se observan las operaciones multiplicación y suma que representan la interacción de la celda con las compuertas forget y update, respectivamente. A continuación se explicaran las partes de la celda LSTM.

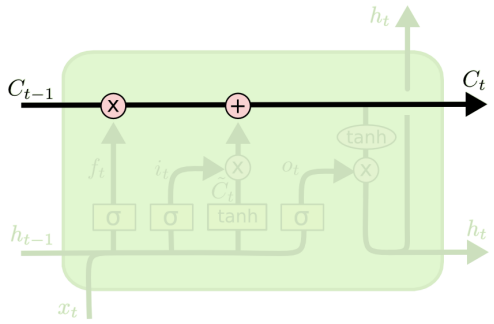


Figura 2.7: Celda de estado de una red LSTM.

Una celda LSTM está formada por 3 compuestas de redes neuronales: forget gate, update gate y output. La primera compuerta llamada forget gate (figura 2.8), está formada por una red neuronal y función de activación sigmoidal, la cual le da el efecto de válvula. Esta compuerta tiene la misión de decidir que información se va a descartar de la celda de estado y cual se mantendrá dentro de la memoria. Como se observa en la ecuación 2.5, la forget gate recibe el estado oculto anterior $h_{t-1}$ y la entrada $x_t$, los transforma y los ingresa a la función sigmoidal, generando un numero entre 0 y 1. Los coeficientes $W_f$ y $b_f$ son calculados en el proceso de entrenamiento de la red. Luego, la salida generada $f_t$, es multiplicada con cada elemento de la celda de estado $C_{t-1}$ anterior. Si uno de los elementos ponderadores de $f_t$ es 1 significa que se mantendrá la información por completo, mientras que si es 0 éste tendrá el efecto de olvidar esa porción de data de la memoria de la celda.

$$f_t = \sigma(W_f[h_{t-1}; x_t] + b_f) \tag{2.5}$$

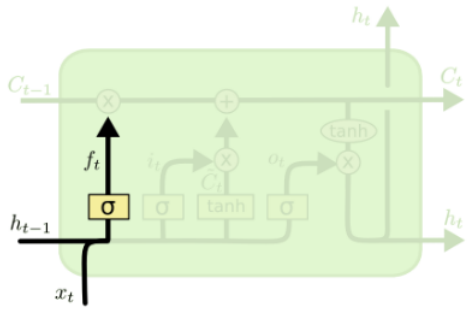


Figura 2.8: Diagrama de la Forget Gate o Compuerta de Olvido.

El siguiente paso es decidir que información nueva será incorporada al estado de celda o memoria. Este paso se divide en dos partes. Primero se decide que datos se actualizaran con la puerta de entrada o update gate (Figura 2.9), que al igual que en la forget gate se toma

el estado oculto anterior $h_{t-1}$ y la entrada $x_t$. Estos se transforman y son ingresados a una capa sigmoidal generando la salida $u_t$, la cual se detalla en la formula 2.6. Los coeficientes $W_i$ y $b_i$ serán aprendidos durante el proceso de entrenamiento. Luego, se crea un vector de valores candidatos $\tilde{C}_t$ a formar parte de la nueva memoria $C_t$. Este transforma y se ingresa a una función de activación *tanh* como se detalla en la ecuación 2.7. Al igual que en las capas sigmoidales, los coeficientes $W_C$ y $b_C$ se aprenden durante el entrenamiento.

$$i_t = \sigma(W_i \cdot [h_{t-1}; x_t + b_i] \tag{2.6}$$

$$\tilde{C}_t = \tanh(W_c \cdot [h_{t-1}; x_t] + b_c) \tag{2.7}$$

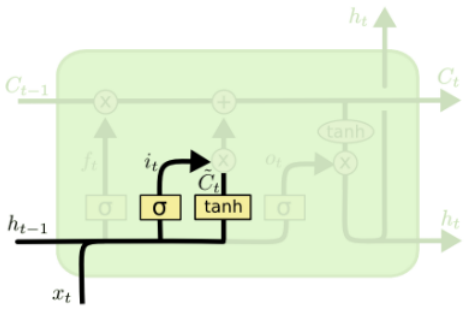


Figura 2.9: Modelo de la Update Gate o Compuerta de Entrada

Una vez calculados los términos de las compuertas forgert gate y update gate, se procede a actualizar la memoria de la red LSTM. En primer lugar, se elimina la información irrelevante de la celda multiplicando el estado anterior $C_{t-1}$ con la salida $f_t$. A continuación, se agregan los nuevos valores candidatos $\tilde{C}_t$ escalados por $i_t$, que representa cuanto se decidió actualizar cada valor de estado. Luego, el resultado se suma al vector de la celda, generando así la memoria actualizada $C_t$ detallada en la ecuación 2.8. Para reforzar la comprensión de este proceso de actualización, se recomienda consultar la figura 2.10.

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \tilde{C}_t \tag{2.8}$$

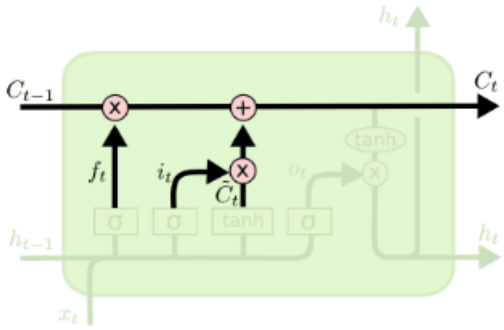


Figura 2.10: Actualización de la celda de estado o memoria

Finalmente, la tercera compuerta llamada output gate, es la encargada de calcular la sa-

lida y el nuevo estado oculto de la red. Este estado es una versión filtrada del estado de la celda generado previamente. En primer lugar se escala la nueva celda de estado para dejarla con valores entre -1 y 1 con la función *tanh*. Luego, se utiliza la output gate para determinar que porciones de la celda de estado pasaran a formar parte del nuevo estado oculto. El efecto sigmoidal detallado en la ecuación 2.9 .es análogo a las compuertas anteriores, calculando los parámetros $W_0$ y $b_0$ durante el entrenamiento. Finalmente, se filtran los valores de la celda de estado por el vector generado por la output gate. Para reforzar la comprensión de este bloque de salida, se recomienda consultar la figura 2.11.

$$o_t = \sigma(W_0 \cdot [h_{t-1}, x_t + b_0]) \tag{2.9}$$

$$h_t = o_t \cdot \tanh(C_t) \tag{2.10}$$

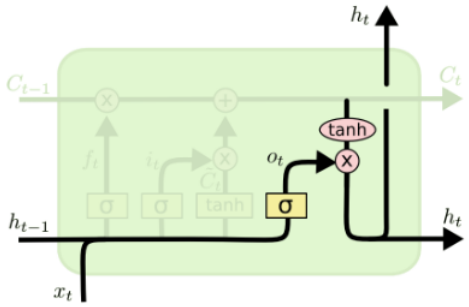


Figura 2.11: Modelo de la Output Gate o Compuerta de Salida

## 2.3. Estado del Arte

El estado del arte se dividirá en dos temáticas principales. En primer lugar los estudios relacionados a la extracción de características enfocada en sismología y en segundo lugar las publicaciones recientes de estimación de magnitud aplicando métodos de inteligencia artificial y estimación de magnitud con una sola estación.

### 2.3.1. Feature Engineering Extraction

En [37] se utiliza una serie de features para predicción de Terremotos. El objetivo del articulo es resaltar la importancia de implementar métodos de Machine Learning en la Sismología y de encontrar el mejor conjunto de características para utilizarlos en la red. Dentro la larga lista de features analizados se pueden encontrar: Mín, Máx, Media, Desviación estándar, FFT, Modos de la señal, nº de aparición de los modos, curtosis, asimetría, entre otras. Para la cuantificación del desempeño de la red se utilizó la métrica de error relativo porcentual media (MAE).

### 2.3.2. Estimación de Magnitud

En [38] se estudia la viabilidad de predecir la respuesta estructural de los edificios analizando los primeros segundos de onda p, con el objetivo de aplicarlo en alertas tempranas de terremotos (EEW). En el articulo se comparan los rendimientos de linear least square regression (LSR) con 4 técnicas de aprendizaje automático (ML): Random Forest, Gradient Boosting, Support Vector Machines y K-Nearest Neighbors. Los 4 métodos de aprendizaje automático demuestran ser siempre superiores a least square regression (LSR), por la complejidad y no linealidad del fenómeno tectónico. El articulo destaca y hace énfasis en que los modelos locales no se pueden exportar directamente y que estos deben ajustarse a la zona para poder implementarse. Se presenta una corrección en función de la magnitud de cada evento para poder exportar el modelo a otros lugares físicos. El estudio trabaja con data de los países de EE.UU y Japón.

En [39] se utilizó una combinación de redes neuronales convolucionales con random forest para discriminar entre ondas p y señales de ruido en una traza sismica. Una de las aplicaciones que se menciona en el articulo es la aplicación del modelo en el problema de activación de alerta temprana de terremotos (EEW). Es importante mencionar que las falsas alarmas son faltas bastante graves en la sociedad actual, por lo que se necesita un modelo con un alto rendimiento. En los resultados del articulo se obtuvo una precisión de 99,6 % para eventos de $M > 4$-5 y decae a 98,8 % para $M < 4$. En este estudio se concluye con un modelo que es capaz de discriminar entre eventos sísmicos e impulsos de ruido, pero no entrega como resultado un valor o estimación de la magnitud del evento. Este método podría combinarse con sistemas de estimación de magnitud como parte de un diseño de una alerta temprana de terremotos (EEW) y alerta temprana de tsunami (TEW).

En [40] se utiliza una Long Short-Term Memory (LSTM) para clasificar las vibraciones de un terremoto en fuente cercana y fuente lejana, con el objetivo de mejorar las EEW cuando el epicentro se ubica a pocos kilómetros de una ciudad. Se mencionan las dificultades que tienen los sistemas de alarma temprana dependientes de más de una estación, donde no es posible aplicar estos sistemas para hipocentros muy cercanos a la zona objetivo.

En [41] se presenta el primer método basado en redes neuronales convolucionales (CNN), llamado ConvNetQuake. Este modelo extrae las caracteristicas de la señal sin ningún tipo de preprocesamiento y clasifica los sismos por rangos de distancias y rangos de magnitudes. El articulo menciona que su metodología puede ser implementada en alerta temprana de terremotos (EEW). En [42], se empleó una CNN para extraer las características de las trazas sísmicas de la representación de tiempo-frecuencia de las señales. El modelo está entrenado con la base de datos de uso publico de Stanford Earthquake (STEAD) [43], la cual está compuesta por más de 450.000 sismos. El modelo es capaz de entregar la distancia epicentral con un error absoluto medio (MAE) de 4,51 [km], una profundidad con MAE 6,15 [km] y una magnitud sismica de MAE 0,26.

En [44] se utiliza un algoritmo de Gutenberg para la alerta temprana de terremotos basado en un clasificador bayesiano que detecta y estima la magnitud de un evento sísmico. Este algoritmo utiliza un banco de filtros para un análisis de tiempo-frecuencia de las señales en tiempo real y luego estima las probabilidades posteriores tanto de la magnitud como de la distancia fuente-estación. Al extraer la información directamente del contenido de frecuencia de la señal observada, este algoritmo no necesita de una estimación de ubicación previa para hacer su estimación. Además, en el artículo se señala que el método presenta saturación en la estimación de eventos de alta magnitud, la cual depende de la longitud de la señal, por lo que estas estimaciones deben ser consideradas como pie mínimo y de característica observacional en vez de predicativas. Por último, el articulo concluye que los resultados obtenidos apoyan la idea de que los terremotos son fenómenos no deterministas.

En [45] se propone un algoritmo basado en redes neuronales artificiales para la estimación rápida de magnitud de momento [Mw] utilizando una sola estación. El modelo plantea emitir una alerta con la estimación realizada con los primero 3 segundos desde el arribo de la onda p. En el articulo se presenta un error absoluto de estimación menor a 0,35 testeando el modelo con terremotos de la región de Sumatra con magnitud 6 o mayor. En [46] se presenta un modelo basado en redes neuronales convolucionales (CNN), para detectar, localizar y caracterizar sismos con una sola estación. Este articulo se destaca por no necesitar aplicar ningun tipo de preprocesamiento a las trazas ni tampoco es necesario tener conocimientos de la zona de estudio. El modelo es capaz de detectar eventos del ruido con un accuracy del 97 %; determinar si el evento se encuentra a menos de 10 [km] de distancia de la estación con un accuracy del 94 % y clasificar el evento entre 4 rangos de magnitud con un accuracy del 68 %.

En [47] se utiliza la técnica de support vector machine regression (SVMR) para la estimación de la magnitud local de un evento, procesando solo 5 segundos de la señal desde el inicio de la onda p en el sismograma. El articulo trabaja con 863 registros sísmicos de la zona de Colombia, con una magnitud entre 2 y 4. Los resultados obtenidos de la estimación es de un error absoluto medio (MAE) de 0,19.

# Capítulo 3

# Implementación

## 3.1. Metodología

Para organizar el trabajo de memoria se realizó una metodología que establece en el primer bloque la revisión bibliográfica donde se investigaron las bases en conceptos sismológicos y procesamientos de señales, hasta un análisis exhaustivo del estado del arte en el uso de inteligencia artificial en sismología y en la estimación de magnitud de un evento sísmico. Luego, se dio paso a la redacción del Marco Teórico, el cual desarrolla los conceptos necesarios para la compresión del presente escrito.

El siguiente bloque corresponde a la búsqueda, descarga y elaboración de la base de datos de eventos sísmicos. Para la creación de ésta se utilizó la información entregada por el Centro Sismológico Nacional (CSN) para la realización del proyecto, la cual viene documentada en un catalogo sísmico. Luego, con la información del catalogo, se procede a descargar la data desde el *Incorporated Research Institutions for Seismology* (IRIS) mediante una rutina escrita en Python utilizando las múltiples funciones que entrega el framework Obspy para el procesamiento de datos sismológicos [48]. Finalmente, se le realiza un preprocesamiento a la base de datos que involucra el filtrado, revisión y etiquetado de las trazas para descartar eventos que no apliquen al modelo. Para mayor detalle consultar la sección 3.2 del trabajo de memoria.

El tercer bloque corresponde al proceso de Extracción de Características (Features) desde la base de datos. Para esto los features serán divididos en 2 grupos: features temporales y features globales. Como dice su nombre, los features temporales son características que dependen del tiempo a medida que transcurre la traza sísmica. Las características temporales seleccionadas para el trabajo corresponden a features espectrales obtenidos mediante la Transformada rápida de Fourier (FFT) y a la energía por frame. Para este ultimo, se proponen tres distintos métodos para el calculo de la energía de la traza. Para mayor detalle consultar la sección 3.4 del trabajo de memoria.

El quinto y ultimo bloque abarca todos los procesos de optimización de la red DNN. Esto considera realizar experimentos con distintas configuraciones de preprocesamiento (filtrado, factor de escala, criterios de corte, entre otros), variación de de hiperparametros y selección de conjuntos de features.

## 3.2. Base de Datos

Chile es un país altamente sísmico, por lo que una detección y estimación temprana de magnitud de los Terremotos es una tarea muy importante de la cual está encargado el *Centro Sismológico Nacional* (CSN). Es importante mencionar que un evento grande podría no solo traer efectos negativos por el movimiento en tierra, además podría desencadenar un catastrófico Tsunami que de ser localizado y alertado con tiempo, disminuiría los costos económicos y sobre todo en vidas humanas.

El CSN pertenece a la Facultad de Ciencias Físicas y Matemáticas de la Universidad de Chile y está compuesto por especialistas en Ciencias de la Tierra y personal de apoyo especializado. Su misión es entregar datos e información de calidad a la *Oficina Nacional de Emergencia del Ministerio del Interior y Seguridad Publica* (ONEMI), *Servicio Hidrográfico y Oceanográfico de la Armada* (SHOA), autoridades, mundo científico, expertos en prevención, mitigación y reducción de riesgo sísmico y a la comunidad en general.

Para la creación de la base de datos se contó con la ayuda del CSN, quienes facilitaron la información de su catalogo sísmico y un listado de recomendación con las estaciones para realizar la investigación. En la tabla 3.1 se presenta el listado de estaciones sismológicas con su respectivo nombre, código, red, y sus coordenadas geográficas.

Tabla 3.1: Estaciones disponibles para realizar la investigación.

| **Estación** | **Código** | **Red** | **Latitud** | **Longitud** |
|---|---|---|---|---|
| Estación Visviri | PB18 | CX | -17.59 | -69.48 |
| Estación Chusmiza | GO01 | GRO-CHILE | -19.67 | -69.19 |
| Estación Quilagua | PB09 | CX | -21.80 | -69.24 |
| Estación Pedro de Valdivia | PB06 | CX | -22.71 | -69.57 |
| Estación Paranal | PB14 | CX | -24.63 | -70.40 |
| Estación Maricunga | AC02 | C1 | -26.84 | -69.13 |
| Estación Tierra Amarilla | GO03 | GRO-CHILE | -27.59 | -70.23 |
| Estación Juntas del Toro | CO01 | C1 | -29.98 | -70.09 |
| Estación Combarbalá | CO02 | C1 | -31.20 | -71.00 |
| Estación CCHEN | MT16 | C1 | -33.43 | -70.52 |

Con la ayuda del catalogo se procede a descargar desde el *Incorporated Research Institutions for Seismology* (IRIS) las trazas sísmicas con una rutina de Python y la aplicación de la librería Obspy. IRIS es un consorcio formado por más de 125 universidades de EE.UU dedicadas a la operación de instalaciones científicas para la medición, gestión y distribución de datos sismológicos. Su acceso es de carácter publico, ya que, de esta forma buscan fomentar la cooperación en investigación y educación de la Sismología.

La traza sísmica contiene la señal del sismograma de los canales N-S, E-O y. eje vertical Z. Ademas, contienen información de la fecha y hora del inicio del evento, la magnitud y la estación de observación. Los sismogramas vienen en cuentas y sin filtrar, por lo que se les realiza un preprocesamiento con el fin de facilitar el aprendizaje de la información por parte de la red.

El preprocesamiento de la data considera las siguientes etapas:

- Conversión de la traza de cuentas a velocidad.
- Remoción de la respuesta instrumental.
- Amplificación de los puntos por un factor de escala con el propósito de evitar la obtención de valores negativos al aplicar la Transformada Rápida de Fourier a la Señal (FFT).
- Ajustar el muestreo de las señales sísmicas para que todas queden a una tasa de muestreo de 40 Hz.
- Corte de eventos aislados y guardado en un archivo único, es decir, se descartan trazas con una seguidillas de eventos (multieventos) que no cumplan con el criterio de corte. El criterio definido está formado por un silencio de al menos 6 segundos previo al evento (!SIL), el evento sísmico (EVENTO) y finalmente un tiempo de silencio posterior a la Coda (!SIL).

En la figura 3.1 se puede observar la zona geográfica y la distribución de las estaciones sismológicas disponibles para la investigación.

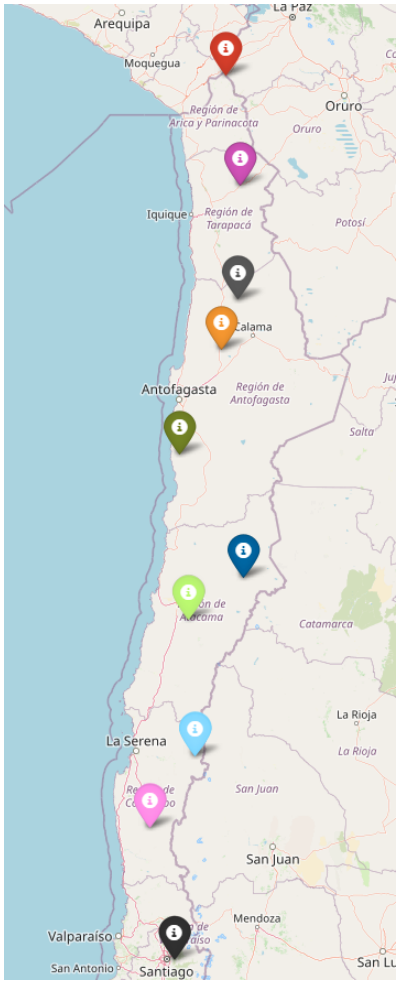


Figura 3.1: Mapa de la zona geográfica a estudiar.

En una fase inicial se utilizó la base de datos de [8], la cual fue extendida y analizada en profundidad en este trabajo de memoria. La base de datos cuenta con 1525 eventos, lo cuales se dividieron en los conjuntos de entrenamiento, testeo y validación según la distribución de la tabla 3.2.

Tabla 3.2: Distribución de la base de datos.

| **Base de Datos** | **Train** | **Val** | **Test** | **Total** |
|---|---|---|---|---|
| Zona Norte-Central | 913 | 306 | 306 | 1525 |
| | 59,9 % | 20,1 % | 20,1 % | 100 % |

En la figura 3.2 se presenta el mapa de la zona centro y norte de Chile con los eventos que forman parte de la base de datos extendida utilizados en la investigación. Donde a diferencia de [8] hay tres veces más eventos, por lo que se ve aumentada su densidad y representatividad de la zona.

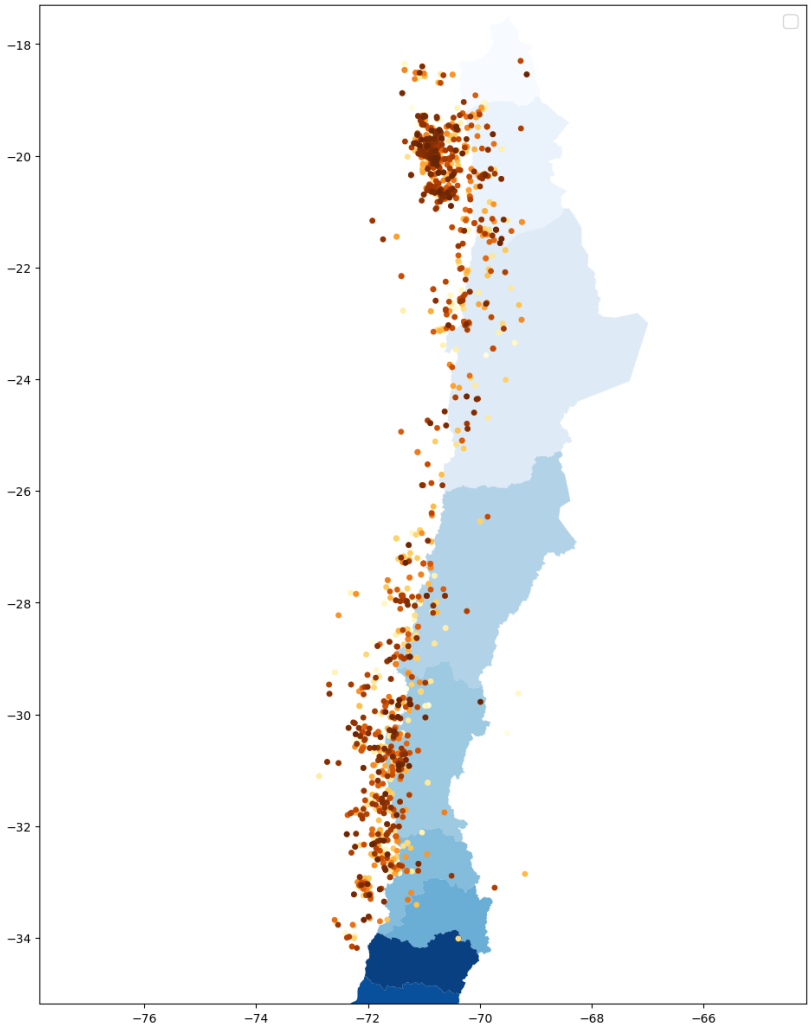


Figura 3.2: Eventos Sísmicos de la base de datos extendida.

## 3.3. Preprocesamiento

Para el preprocesamiento de los datos, es importante mencionar que las trazas sísmicas vienen en un formato de cuentas. Por lo que primero que todo hay que transformarlas a velocidad y eliminar la respuesta instrumental que viene incluida. Posteriormente, la amplitud se multiplicó por una constante $(10E+10)$ para evitar números negativos después de aplicar log al valor absoluto de la FFT. La frecuencia de muestreo también se ajustó a 40 Hz. Finalmente, las señales se procesaron aplicando un filtro paso alto con una frecuencia de corte de 1 Hz para eliminar las bajas frecuencias de ruido.

## 3.4. Extracción de Features

La Extracción de Features es un proceso fundamental para recopilar información minuciosa de los datos. Hacer una buena extracción permite mejorar el rendimiento de la red, sobre todo cuando no se tiene una gran extensión de data, como ocurre en el caso de la sismología donde hay una escasez de grandes eventos en las bases de datos. Otro elemento a favor es la reducción de dimensionalidad de los datos convirtiendo las muestras de la señal en una secuencia de vectores con información menos redundante. Las Características o Features serán divididos en 2 tipos: Features Temporales y Features Globales.

En la figura 3.3 se presenta el diagrama conceptual del proceso de extracción de características de los features temporales y features globales. Es importante destacar que los features temporales se ingresan a la red LSTM, mientras que los features globales se concatenan con la salida de la LSTM antes de entrar a la capa *Fully Connected*.

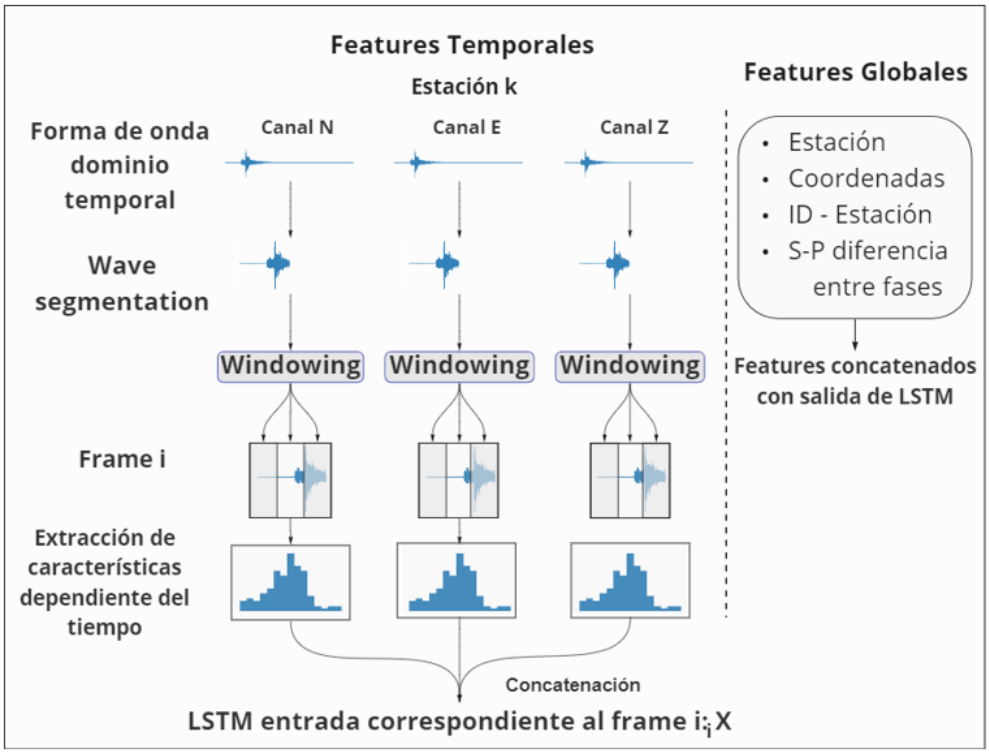


Figura 3.3: Mapa conceptual del proceso de extracción de características.

### 3.4.1. Features Temporales

Los features temporales escogidos para extraer de la señal sísmica son la Transformada de Fourier con sus variantes y un feature de energía calculado de tres formas distintas. Estas características se utilizaron como entrada para la red LSTM Bi-Direccional.

#### 3.4.1.1. Espectro de Frecuencia

Se escogió este feature porque es la herramienta matemática clásica a la hora de analizar una señal temporal. El dominio de la frecuencia permite a los analistas visualizar las bandas dominantes con mayor facilidad e identificar el contenido espectral que define a un evento tectónico. Es por esto que se que optó por entrenar a la red neuronal con esta entrada para ver si lograba aprender y estimar la magnitud de los eventos del conjunto de test. Para su calculo se utilizó la función *np.fft.fft* de Numpy que calcula la Transformada Discreta de Fourier (DFT) con el eficiente algoritmo de la Transformada rápida de Fourier (FFT) [49].

La Transformada Discreta de Fourier (DFT) viene dada por la ecuación 3.1, donde N es el largo de una señal x[n].

$$X\left[k\right] = \sum_{n=0}^{N-1} x\left[n\right] \cdot e^{-\frac{2\pi i}{N} kn}\,, \qquad con\ k = (1, ..., N-1) \tag{3.1}$$

En el proceso de experimentación y optimización del modelo, se experimentó con 2 expresiones para el feature de frecuencia. Primero se corrieron experimentos con la Transformada Rápida de Fourier (FFT) en su versión lineal y luego, se le agregó el $\log_{10}$ al calculo. Finalmente, el agregar el $\log_{10}$ al calculo mejoró el rendimiento de la red, por lo que se optó por esa configuración.

#### 3.4.1.2. Energía

La energía de la señal sísmica es el segundo feature temporal propuesto para la extracción de características. Para su obtención se ocuparon tres métodos de calculo, con los que se fue experimentando en el proceso de optimización de la red con el fin de encontrar la mejor configuración.

1. E1: este feature está basado en el teorema de Parseval para calcular la energía por frame. Se realizaron pruebas con su forma lineal y forma logarítmica, cuyas expresiones se detallan en las ecuaciones 3.2 y 3.3 respectivamente.

   - Con escala lineal:
   $$E_1 = \sum_i (|y_i|^2) \tag{3.2}$$

   - Con escala logarítmica:
   $$E_1 = \sum_i log_{10}(|y_i|^2) \tag{3.3}$$

2. E2: Este feature de energía es similar a E1, pero con la diferencia que no se considera el valor absoluto en el cálculo. Se realizaron pruebas con su forma lineal y forma logarítmica, cuyas expresiones se detallan en las ecuaciones 3.6 y 3.7 respectivamente.

   - Con escala lineal:
   $$E_1 = \sum_i (x_i^2) \tag{3.4}$$

- Con escala logarítmica:

$$E_1 = log_{10}(\sum_i x_i^2) \tag{3.5}$$

3. **E3:** este feature de energía es similar a $E_2$ con la diferencia que para esta expresión se normaliza por el máximo de la secuencia. Se realizaron pruebas con su forma lineal y forma logarítmica, cuyas expresiones se detallan en las ecuaciones 3.6 y 3.7 respectivamente.

   - Con escala lineal:

$$E_3 = E_2 - max(E_2) \tag{3.6}$$

   - Con escala logarítmica:

$$E_3 = \frac{E_2}{max(E_2)} \tag{3.7}$$

### 3.4.2. Features Globales

Los Features Globales son características propias de la traza que no tienen una dependencia temporal y que a priori se cree que pueden mejorar el rendimiento de la red neuronal. Estas características son: Coordenadas del punto de observación (Estación), Indexación de la estación y Delta de tiempo de arribo de las ondas de cuerpo p y s. Estos son concatenados con la salida de la LSTM Bi-Direccional y posteriormente ingresados en conjunto a la capa *Fully Connected*, la cual entrega la magnitud final del evento.

#### 3.4.2.1. Coordenadas de la estación

En primer lugar se propone entregar las coordenadas del punto de observación (Estación), con el propósito de que la red pueda aprender y encontrar una relación de como se atenúa la señal para distintas magnitudes de eventos con respecto al punto de medición. Otro elemento que apoya la implementación de este feature, es que la red logre identificar como se observan los distintos eventos para cada una de las estaciones disponibles de referencia, aprendiendo caracteristicas propias de la zona aledaña.

#### 3.4.2.2. Indexación de la estación

La indexación de la estación consiste en asignar un índice a cada estación disponible, esta puede ser realizada de norte a sur, con alguna configuración definida previamente o de forma aleatoria. Este tipo de Feature parece sencillo, sin embargo, al momento de optimizar a la red neuronal se observa que al asignar un identificar a cada punto de observación facilita el entrenamiento del modelo. Para la experimentación se probó con indexación *One-Hot-Encoding* y con números enteros positivos, de donde destaca la configuración directa con números enteros para cada estación de referencia.

#### 3.4.2.3. Delta de tiempo de ondas s y p (S-P)

En la literatura se ha visto que es posible realizar una estimación rápida de la distancia hipocentral, la cual se basa en el hecho de que la onda p es más rápida que la onda s. Esto implica que el tiempo de viaje desde el foco a la estación es menor para la onda p y que a medida que se aleja el punto de observación la llegada de ambas ondas se ve desplazado en el sismograma. Al reemplazar las velocidades promedio de las ondas de cuerpo en las capas superficiales de la tierra en la formula de distancia hipocentral, se obtiene un factor de $\approx 8$

que si se multiplica por el $\Delta t_s - t_p$, da como resultado una estimación rápida de la distancia del foco a la estación. Este supuesto se puede asumir, ya que, las velocidades de las ondas p y s no tienen grandes variaciones en los primeros 300 kilómetros de profundidad.

En la figura 3.4 se presenta un sismograma de ejemplo para facilitar la comprensión y visualización de las ondas que forman parte de un evento sísmico.

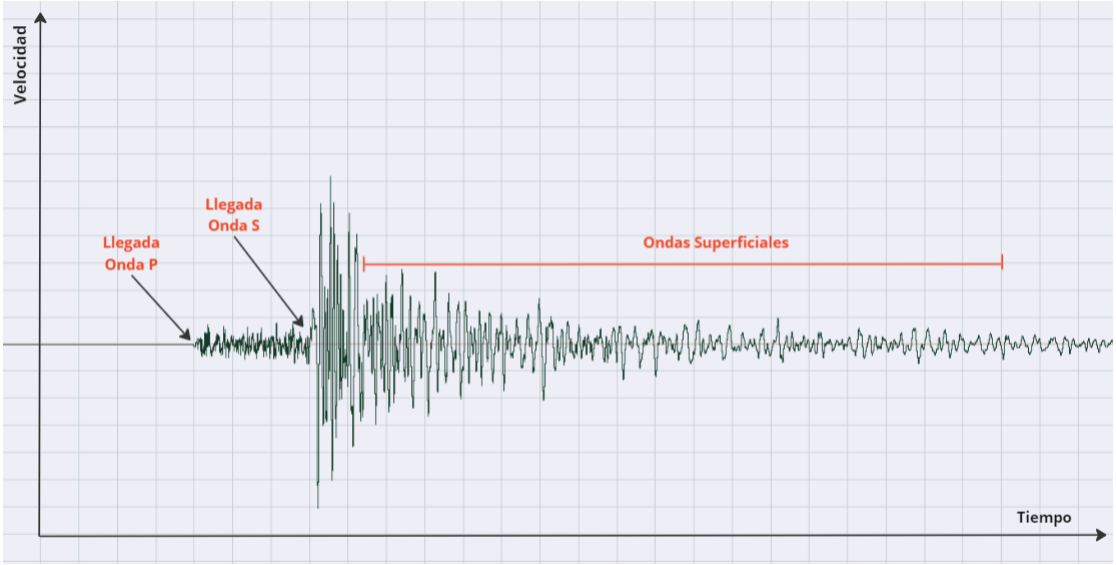


Figura 3.4: Sismograma con sus ondas de cuerpo y ondas superficiales.

A continuación, se presenta la demostración y desarrollo matemático de la estimación rápida de distancia hipocentral, donde se comienza desde la siguiente ecuación:

$$D_h = \frac{(t_s - t_p)}{(\frac{1}{V_s} - \frac{1}{V_p})} \tag{3.8}$$

- $D_h$: distancia hipocentral en [m]
- $V_s$: Velocidad onda s en $[\frac{m}{s}]$
- $V_p$: Velocidad onda p en $[\frac{m}{s}]$
- $t_s$: tiempo de llegada onda s en [s]
- $t_p$: tiempo de llegada onda p en [s]

La velocidad promedio para las onda p y s son 7500-8500 y 3500-4500 $[\frac{m}{s}]$ respectivamente [50]. Al Reemplazar en la ecuación 3.8 y expresando el resultado en [km], la diferencia temporal de los tiempos de llegada queda multiplicada por un factor $\approx 8$, por lo que la ecuación se simplifica y queda expresada como en 3.9.

$$D_h = (t_s - t_p) * 8 \tag{3.9}$$

- $D_h$: distancia hipocentral en [km]
- $t_s$: tiempo de llegada onda s en [s]
- $t_p$: tiempo de llegada onda p en [s]

## 3.5. Métricas

El objetivo de llevar a cabo la tarea de evaluar el desempeño de una Red DNN después de obtener los resultados de estimación de magnitud de un sismo, es verificar la capacidad de la red neuronal para realizar estimaciones precisas, confiables y competitivas en términos de monitoreo sísmico. La evaluación permite determinar si la red ha aprendido patrones relevantes y si ha sido capaz de generalizar correctamente a partir de los datos de entrenamiento. Es esencial tener una comprensión clara de la precisión de las estimaciones de magnitud necesarias para poder tomar decisiones claves en situaciones de sismicidad, como la evaluación de riesgos o la implementación de medidas de prevención y respuesta temprana ante terremotos. Cada uno es un problema distinto y con requerimientos diferentes.

Las tres métricas propuestas son: error medio absoluto (MAE), error porcentual absoluto medio (MAPE), y error cuadrático medio (MSE). Éstas permiten cuantificar y comparar el rendimiento de la red en diferentes aspectos clave. El MAE, proporciona una medida de la diferencia promedio entre las estimaciones de magnitud y los valores reales, mientras que el MAPE, indica el porcentaje promedio de error en las estimaciones de magnitud. Además, se utilizó el MSE para medir la dispersión de los errores, al calcular la media de los errores cuadráticos entre las predicciones y los valores reales. Estas métricas facilitan el proceso de evaluar la precisión y el desempeño de la red en la estimación de magnitudes sísmicas, proporcionando una visión completa de la factibilidad de implementarse en el monitoreo nacional.

### 3.5.1. Error relativo porcentual medio (MAPE)

Esta métrica fue escogida por el cuerpo docente y el equipo de investigación del LPTV, la cual es la que principalmente se utiliza para comparar los resultados en la estimación de magnitud. Su formula se define en la ecuación 3.10.

$$\mu_{Error(\%)} = \frac{1}{N}\sum_{n=1}^{N}\left|\frac{M_{real} - M_{Estimada}}{M_{real}}\right| * 100 \tag{3.10}$$

Donde N es el número de pruebas realizadas, $M_{real}$ corresponde a la magnitud de referencia entregada por el CSN y $M_{Estimada}$ es la magnitud estimada por la Red DNN.

### 3.5.2. Error absoluto medio (MAE)

Esta métrica fue escogida debido a que en el estado del arte de estimación de magnitud sísmica con métodos de inteligencia artificial, la mayoría de los artículos presentan los errores de sus resultados en MAE. Por lo tanto, con el fin de poder comparar la red DNN implementada con la literatura y así analizar su factibilidad para alertas tempranas, se propone calcular esta métrica. Su formula de cálculo se presenta en la ecuación 3.11.

$$MAE = \frac{1}{N}\sum_{n=1}^{N}|M_{Registrado} - M_{Estimada}| \tag{3.11}$$

### 3.5.3. Error cuadrático medio (MSE)

El error cuadrático medio es una de las métricas más utilizadas en métodos de regresiones lineales e inteligencia artificial. Si bien no se ha visto muchos artículos de Sismología, debido a su importancia se propone calcular este error. Su formula se define en la ecuación 3.12.

$$MSE = \frac{1}{N}\sum_{n=1}^{N}(M_{Registrado} - M_{Estimada})^2 \tag{3.12}$$

### 3.5.4. SNR

El SNR, del ingles *Signal-To-Noise Ratio*, es una métrica que establece una relación entre la potencia de una señal limpia con la potencia del ruido de fondo presente en el registro. Las principales formas de cálculo son el simple cociente entre las potencias de la señal y el ruido, presentado en la ecuación 3.13. Y en el área del procesamiento de señales es común su uso en la versión en decibelios presentada en la ecuación 3.14. En aplicaciones sismológicas, se espera que para sismos de mayor magnitud la diferencia Señal-Ruido sea mayor que para sismos de baja magnitud. Por lo tanto, se puede implicar que su SNR también sea mayor para este caso de análisis.

$$SNR = \frac{P_S}{P_R} \tag{3.13}$$

$$SNR_{dB} = 10 \cdot \log_{10}\frac{P_S}{P_R} \quad [dB] \tag{3.14}$$

- $P_S$: Potencia de la señal
- $P_R$: Potencia del ruido

## 3.6. Optimización de la Red DNN

Para el trabajo de memoria se experimentó con un arquitectura de red basada en una LSTM Bidireccional con una Capa Densa de salida. La información que alimenta la arquitectura de aprendizaje profundo se compone de dos entradas: las características dependientes del tiempo que alimentan la LSTM Bidireccional y las características globales que son concatenadas con la salida del bloque LSTM, para luego ingresarse en una Capa Densa (MLP) o tambien llamada *Fully Connected*. Luego, la Capa Densa desemboca en una neurona la cual entrega una estimación de la magnitud del terremoto.

Una de las principales razones para utilizar una topología basada en una red LSTM, es por su capacidad de modelar dependencias a largo plazo [35]. A diferencia de otras arquitecturas de redes neuronales, la red LSTM no necesita que la secuencia de vectores de entrada tenga el mismo tamaño, lo que se relaciona con la dinámica de los eventos sísmicos. En la figura 3.5 se presenta un diagrama con la red utilizada en el trabajo de memoria.

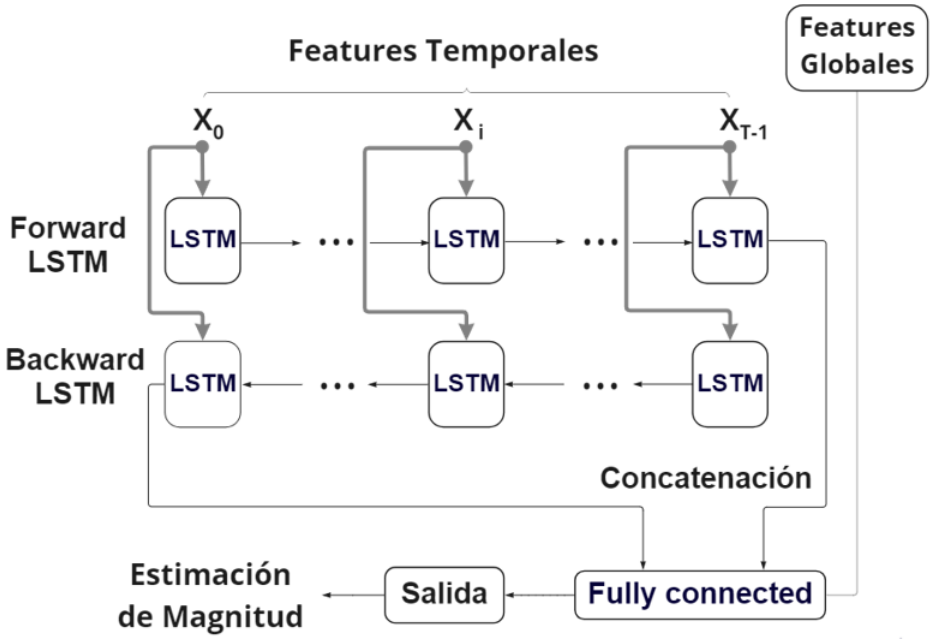


Figura 3.5: Mapa conceptual de la arquitectura de red neuronal utilizada.

Para la optimización de la red neuronal se mantuvo como base el feature temporal obtenido con la FFT y se varió el feature de energía con sus distintas versiones de cálculo. Por otro lado, para los features globales se experimentó con cada feature por separado (con variaciones en su representación) y la combinatoria de los features globales ID-Estación/S-P.

Por último, se extrajeron resultados con los casos *Raw Data* y con la red entrenada sin features globales. A pesar de que se esperaba desde antes un peor desempeño con este último caso, su contraste con la red final permite demostrarla importancia de los "Engineered Features" a la hora de diseñar un modelo basado en redes neuronales profundas. Los resultados obtenidos y su respectivos análisis se presentan en la siguiente sección.

# Capítulo 4

# Discusión de Resultados

En este capitulo se presentarán los resultados principales obtenidos en el proceso de optimización del modelo de estimación de magnitud de eventos sísmicos basado en redes neuronales artificiales. El análisis y comparación de errores mencionados en cada sección es realizado con la métrica del error porcentual relativo (MAPE), a menos que se especifique lo contrario. Para una mayor comprensión, el análisis de resultados se divide en features temporales y globales. Luego, con la mejor configuración escogida se analiza el desempeño del modelo para los rangos de magnitud M>4, M<4 y todas las magnitudes.

## 4.1. Features Temporales

En esta subsección se discutirán los resultados obtenidos ingresando solamente features temporales como el análisis del espectro de Fourier con sus variaciones y un feature de energía con sus respectivas alternativas de calculo.

En la tabla 4.1 se presentan las métricas de error para el caso de uso de la data raw sin ningún tipo de procesamiento, versus los casos en que se extrae un feature característico del dominio de Fourier. Los peores resultados se observan para la data raw (27,83 %) y la FFT (27,61 %), feature que considera la concatenación de la parte real e imaginaria de la Transformada rápida de Fourier. El tercer resultado, el cual presenta una mejora importante, corresponde a la versión del feature de Fourier utilizado en [8], basado en el cálculo del $\log_{10}$ del valor absoluto de la FFT. Es importante destacar que este método de cálculo elimina el factor de la fase de la señal al aplicar el valor absoluto a la señal, obteniendo las amplitudes independiente de su signo. El rendimiento observado permite concluir que la versión con valor absoluto facilita el aprendizaje por parte de la red neuronal, la cual es capaz de estimar la magnitud de un evento con un error mucho menor alcanzando el valor de (10,98 %). Es por esto que este feature se mantiene fijo para el resto de la investigación, el cual sienta las bases para la optimización.

Tabla 4.1: Features de Fourier v/s Raw Data

| **Feature Fourier** | **MAPE** | **MAE** | **MSE** |
|---|---|---|---|
| Raw | 27,83 % | 0,96 | 1,29 |
| Transformada de Fourier | 27,61 % | 0,96 | 1,28 |
| Original | 10,98 % | 0,38 | 0,26 |

En la tabla 4.2 se presentan los resultados obtenidos para los 3 métodos distintos del cálculo de la energía por frame de la traza sísmica propuestos para el aprendizaje de la red. En primer lugar, se observa que E1 tiene el mayor error (10,80 %) de las 3 energías, donde no se observa una potencial mejora. Mientras que E2 y E3 presentan un error similar (7,00 % y 6,97 % respectivamente). Por lo tanto, se puede concluir que se pueden ocupar ambos features de energía para la optimización del modelo y para el paso posterior de encontrar el mejor feature global que minimice el error de la estimación de magnitud.

Tabla 4.2: Feature temporal de Energía.

| **Energía** | **MAPE** | **MAE** | **MSE** |
|---|---|---|---|
| Sin Energía | 11,03 % | 0,38 | 0,27 |
| E1 | 10,80 % | 0,38 | 0,26 |
| E2 | 7,00 % | 0,24 | 0,11 |
| E3 | 6,97 % | 0,24 | 0,11 |

Este feature de energía además de ser importante por su mejora en el error de la red, también es fundamental en el momento de la selección de los datos de la traza sísmica realizada por el extractor de características. La mejor configuración observada en el tuneo de la red, fue aplicar un corte del evento cuando la amplitud del sismo decae al valor del 3 % del máximo del sismo por frame. El realizar esta segmentación disminuye la velocidad de entrenamiento en aproximadamente 3 a 4 veces, lo que facilitó bastante el proceso de tuneo de la red y en caso de realizar estudios futuros en esta materia, se sugiere utilizar una segmentación por energía.

En la figura 4.1 se presenta una traza sísmica referencial de un evento con sus ondas de cuerpo p y s destacadas en rojo. Desde el punto de vista geofísico se espera que en estas zonas se encuentre la mayor cantidad de información del proceso de ruptura de una placa tectónica. Luego, la amplitud de la señal va decayendo rápidamente y debería converger al ruido de fondo en segundos o minutos. Es por esto que se realiza un corte destacado en azul correspondiente al 3 % del valor máximo del evento, lo que disminuye considerablemente la cantidad de información extraída de cada archivo y acelera la obtención de resultados.

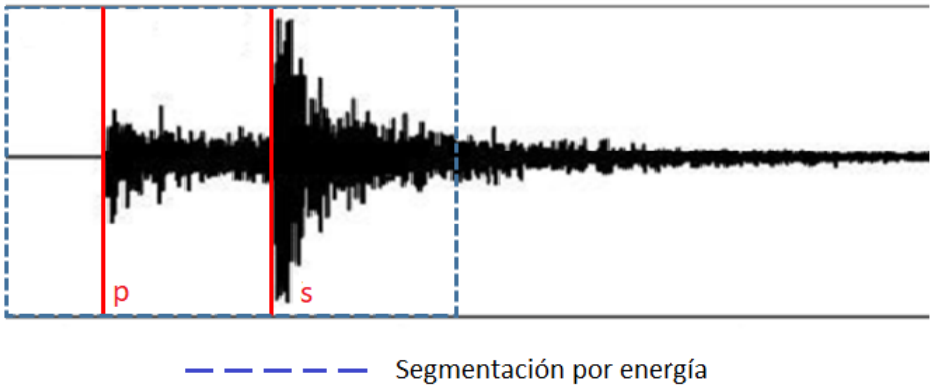


Figura 4.1: Sismograma referencial con el corte de la traza en el 3 % de su valor máximo.

## 4.2. Features Globales

En esta subsección se discutirán los resultados obtenidos con los distintos features globales propuestos en la investigación. Luego, se selecciona el mejor modelo y se evalua su desempeño para distintos rangos de magnitud.

En la tabla 4.3 se presentan las métricas de los mejores experimentos incluyendo todas las magnitudes conectando uno o más features globales a la salida de la celda LSTM. El primer feature y más destacado corresponde a la indexación de la estación sísmica (ID-Estación), el que minimiza el error alcanzando un valor de 6,48 %. En un comienzo se esperaba que cambiando este índice por las coordenadas de la estación se llegara a resultados similares, ya que, al igual que el ID, las coordenadas simbolizan un representación única para cada estación de observación. Sin embargo, el error se ve claramente incrementado alcanzando un valor de error de 6,59 %. Esto demuestra que una representación simple como un numero entero, permite un mejor aprendizaje que un numero más grande de 3 componentes como lo es una ubicación geográfica.

Debido al buen rendimiento del feature de indexación, se experimentó con asociar una neurona de entrada para cada estación con una codificación de one-hot, lo cual no representó mejoras en la estimación de magnitud de la red. Por el contrario el error subió a 7,56 % aun cuando se mantuvieron todos los parámetros de la configuración ID original. Es por esto que se optó por mantener una representación simple del punto de observación o medición.

Tabla 4.3: Resultados obtenidos con Features Globales para todas las magnitudes.

| **Feature Global** | **MAPE** | **MAE** | **MSE** |
|---|---|---|---|
| Sin Global Features | 10,98 % | 0,38 | 0,26 |
| ID Estación | 6,48 % | 0,22 | 0,09 |
| Coordenadas | 6,59 % | 0,22 | 0,10 |
| ID One-hot | 7,56 % | 0,26 | 0,13 |
| Tiempo S-P | 6,81 % | 0,23 | 0,11 |
| ID con S-P | 6,61 % | 0,22 | 0,10 |

El siguiente feature presentado corresponde a concatenar la salida de la celda LSTM con la diferencia temporal de arribo de las ondas de cuerpo p y s. A priori se esperaba que este podía ser el mejor de los features globales, debido a que está información representa una estimación de la distancia desde la ruptura o hipocentro a la estación que recibe la señal. Si bien mejora el error v/s el caso sin features globales, con un error de 6,81 %, no fue capaz de mejorar el rendimiento del ID o de las Coordenadas de la estación de monitoreo.

Estos últimos features mencionados obtuvieron los mejores resultados del proceso de optimización de la red. Si bien en la teoría ambos representan la misma idea de darle una identificación a cada estación, matemáticamente son valores muy distintos. Por un lado se tienen las coordenadas que se componen por un vector de 3 componentes, mientras que el ID Estación es un número entero de dos dígitos. Lo que permite concluir que el aprendizaje de la

red neuronal se ve claramente facilitado cuando se le entrega la identificación de la estación con una representación simple y concisa.

Por ultimo, se experimentó incorporando los Features Globales de S-P e ID-Estación juntos para entrenar la red, lo cual al igual que los casos anteriores mejora el rendimiento v/s sin usar Global Features, obteniendo un error de 6,6 %. Sin embargo, corresponde a un error mayor que utilizar el ID de la Estación como único feature global. Por lo tanto, se descarta esta configuración para la selección del mejor modelo. Pensando en la aplicación de alertas tempranas de terremotos (EEW) y de alertas tempranas para tsunamis (TEW), se concluye que los errores obtenidos son totalmente aceptable como una estimación preliminar de la magnitud de un evento sísmico.

En la tabla 4.4 se presentan las métricas de los mejores experimentos considerando los eventos sismicos con magnitud mayor a 4. En este conjunto se encuentran desde los sismos perceptibles por la población hasta los eventos potencialmente peligrosos y de los cuales tenemos una cantidad limitada de datos. En particular, los sismos de magnitud mayor a 8 son mucho más complejos de analizar, ya que, corresponde a un fenomeno que no puede modelarse como una fuente puntal, sino que corresponde a una ruptura que puede alcanzar centenas de kilometros de ruptura en la placa.

En general, los experimentos realizados tienen un comportamiento similar a la base de datos completa, donde en primer lugar la red sin considerar features globales alcanza el mayor error con 7,41 %. Luego, el segundo mejor feature corresponde al uso de la diferencia temporal de las ondas de cuerpo p y s, el cual disminuye el error hasta 4,39 %. Nuevamente los mejores modelos se obtienen asignando una identificación a la estación que registra el evento. La configuración que minimiza el error es el IDEstación básico de numero entero, descartando los resultados obtenidos con las Coordenadas y con el ID One-hot.

Tabla 4.4: Resultados obtenidos con Features Globales para M > 4.

| **Feature Global** | **MAPE** | **MAE** | **MSE** |
|---|---|---|---|
| Sin Global Features | 7,41 | 0,36 | 0,20 |
| ID Estación | 3,93 | 0,19 | 0,06 |
| Coordenadas | 4,24 | 0,20 | 0,07 |
| ID One-hot | 4,67 | 0,22 | 0,09 |
| Tiempo S-P | 4,39 | 0,21 | 0,07 |
| ID con S-P | 4,05 | 0,19 | 0,06 |

En la tabla 4.5 se presentan las métricas de los mejores experimentos considerando los eventos sismicos con magnitud menor a 4. Este segmento resulta interesante de analizar para estudios de micro-sismos y caracterización de la sismicidad en una zona de estudio. Como primera observación es importante mencionar que es el conjunto de la base de datos que tiene los peores resultados. La razón que se propone es que para la red es dificil estimar la magnitud de eventos con bajo SNR, lo cual se observa para los eventos de baja magnitud. En cuanto a los resultados, el mayor error lo obtiene el modelo sin features globales con un valor de 10,19 %.

Luego, todos los features globales representan mejoras en el rendimiento de la red, logrando el mínimo de 7,26 % con el ID Estación. Para este caso de análisis el feature de Tiempo S-P obtiene resultados muy similares a los de ID-Estación, llegando a obtener errores menores en algunos experimentos de testeo. Sin embargo, para el conjunto de validación el ID-Estación siempre logró resultados menores, por lo que se opto por esa configuración.

Tabla 4.5: Resultados obtenidos con Features Globales para M < 4.

| **Feature Global** | **MAPE** | **MAE** | **MSE** |
|---|---|---|---|
| Sin Global Features | 10,19 % | 0,29 | 0,13 |
| ID Estación | 7,26 % | 0,20 | 0,07 |
| Coordenadas | 7,35 % | 0,21 | 0,07 |
| ID One-hot | 8,58 % | 0,24 | 0,09 |
| Tiempo S-P | 7,32 % | 0,21 | 0,07 |
| ID con S-P | 7,35 % | 0,21 | 0,07 |

En la tabla 4.6 se presenta como se ven modificadas las métricas al variar el *Learning Rate* (Lr) del optimizador de la red. Al momento de experimentar con features globales se observó que la red era bastante sensible a este parámetro. Por lo que se mantuvo fija la configuración de la red considerando solamente el ID-Estación para encontrar el valor óptimo, el cual corresponde a un $Lr = 1x10^{-4}$. Luego, se corrieron todos los experimentos de los features globales de las tabla 4.3, 4.4 y 4.5 con este valor de Lr, con el fin de ser consistentes con los resultados presentados y que estos sean comparables entre si.

Tabla 4.6: Resultados obtenidos variando el Learning Rate con Features Globales y M > 4.

| **Learning Rate** | **MAPE** | **MAE** | **MSE** |
|---|---|---|---|
| $1x10^{-2}$ | 10,79 % | 0.53 | 0,46 |
| $1x10^{-3}$ | 6,64 % | 0,31 | 0,15 |
| $1x10^{-4}$ | 3,93 % | 0,19 | 0,06 |
| $1x10^{-5}$ | 4.14 % | 0.20 | 0.08 |
| $1x10^{-6}$ | 9.33 % | 0.47 | 0.36 |

## 4.3. Análisis gráfico por conjunto de datos

En esta subsección se discutirán los resultados obtenidos gráficamente agrupando la base de datos por tramos de magnitud. Se definieron 3 casos principales: Todas las magnitudes disponibles, Magnitudes mayor a 4 y Magnitudes menor a 4. Para este análisis se consideró el mejor modelo optimizado previamente, el cual corresponde a la celda LSTM entrenada con ambos Features Temporales y el Feature Global de ID-Estación. Este feature se concatena con la salida del bloque LSTM y posteriormente se ingresa a la Capa *Fully Connected*, la cual entrega el valor final de la estimación de magnitud.

En la tabla 4.7 se presentan los mejores resultados obtenidos una vez realizado el tuneo de la red para cada segmento de magnitud. El menor error se tiene para el conjunto de magnitud mayor a 4, el cual se diferencia principalmente de M <4 por su alto SNR en sus sismogramas. Si bien el conjunto M < 4 presenta un error claramente mayor, es considerado como un error totalmente aceptable para una estimación preliminar de magnitud. Finalmente, la data completa con todas las magnitudes incluidas en el entrenamiento, obtuvo un error de 6,48 % en su estimación de magnitud sísmica.

De los resultados se puede concluir que dependiendo del objetivo que se busca en el monitoreo se pueden utilizar distintos modelos. Sin embargo, para la tarea de alertas tempranas de terremotos (EEW) y alertas tempranas de Tsunami (TEW) el rango mayor a M4 viene a ser el más relevante con eventos que pueden ser potencialmente destructivos.

Tabla 4.7: Resultados modelo optimizado

| **Magnitud** | **MAPE** | **MAE** | **MSE** |
|---|---|---|---|
| Todas las magnitudes | 6,48 | 0,22 | 0,09 |
| Magnitud > 4 | 3,93 | 0,19 | 0,06 |
| Magnitud < 4 | 7,26 | 0,20 | 0,07 |

### 4.3.1. Todas las magnitudes

En la figura 4.2 se presenta la magnitud estimada v/s la magnitud real del sismo, donde cada punto representa el promedio de 30 experimentos con su respectiva desviación estándar en las barras. Como se puede observar los datos se ajustan de buena forma a la recta, sobre todo para magnitudes entre M2 y M7. En el extremo izquierdo se puede ver que la red tiende a sobre-estimar la magnitud de los eventos, donde el bajo SNR dificulta el aprendizaje de la red y se ve reflejado en el valor entregado por el modelo. En cambio, en el extremo derecho ocurre lo contrario, donde se presenta un evento con magnitud sobre M8 con una estimación cercana a M7, lo que da a entender de una sub-estimación de grandes eventos por parte de la red.

Este problema de sub-estimación no es directo de solucionar, ya que, no se cuenta con datos de grandes terremotos para entrenar los modelos. Esta escasez de datos es debida a lo poco frecuente que son los megaterremotos sobre magnitud M8.5. Es por esto, que se propone complementar el sistema con herramientas extras a la hora de pensar en una alerta temprana de terremotos.

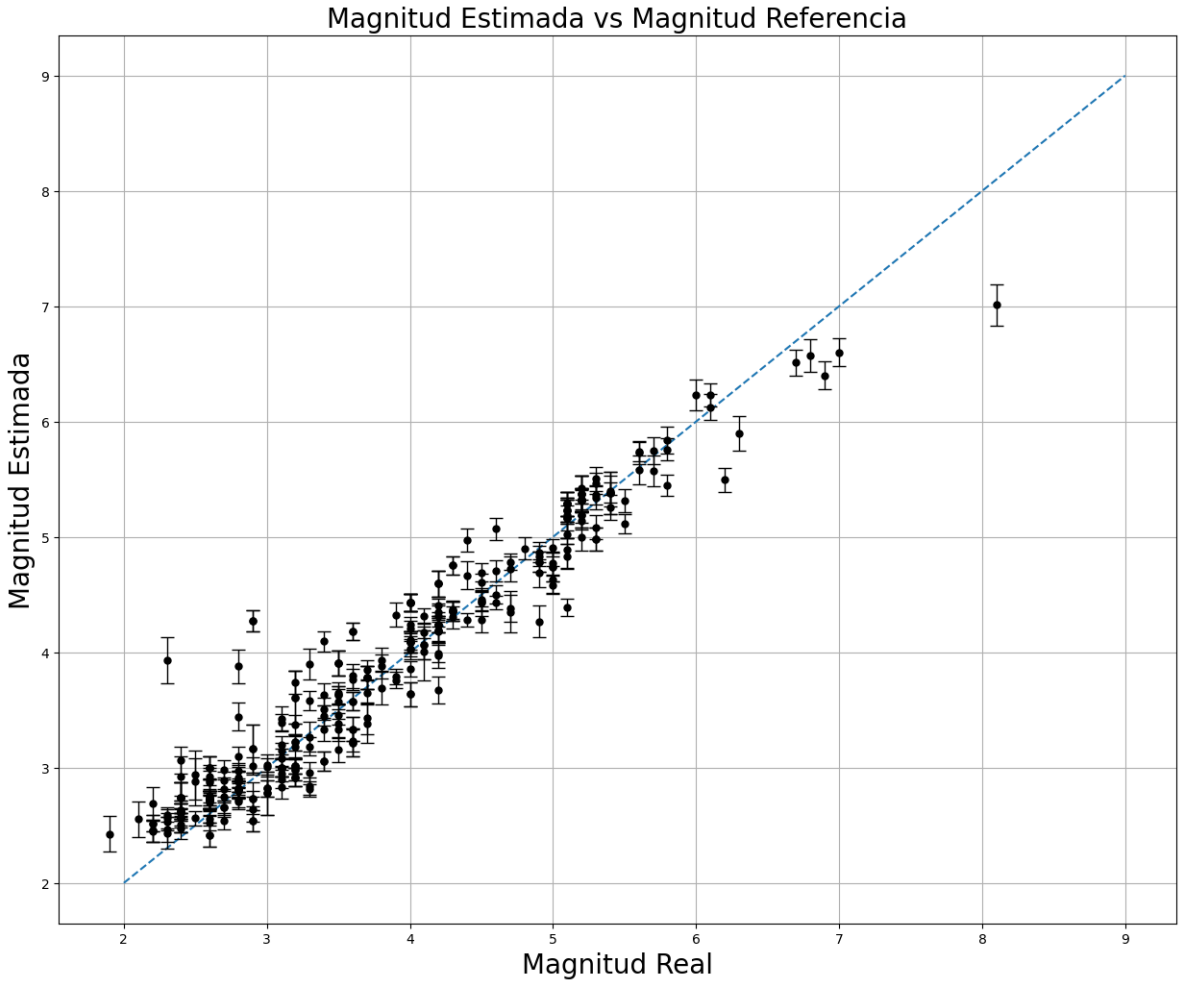


Figura 4.2: Magnitud estimada versus magnitud de referencia para todo el rango de magnitud. El punto representa la media entre 30 experimentos con su Desviación estándar en las barras.

Este comportamiento se vio para todos los experimentos realizados durante el trabajo de memoria, llegando a la conclusión que la red es capaz estimar de forma bastante precisa a los sismos entre magnitud M2 y M7, que coincide con las zonas con más datos. Los sismos menores M2 son difíciles que detectar, mientras que para los sismos mayores a M8,5 no existen datos recientes en Chile. Sin embargo, la red demuestra que entrega estimaciones totalmente aceptables con los errores preliminares del Centro Sismológico Nacional (CSN) en el orden de segundos, por lo que se propone utilizar esta herramienta en conjunto con otros bloques con el objetivo de uso en alertas tempranas de terremotos y tsunamis.

En la figura 4.3 se presenta un grafico similar al previamente analizado, con los mismos puntos de promedio para cada evento, pero esta vez las barras corresponden al valor maximo y minimo obtenido para cada sismos. En particular, este resultado fue sugerido por el equipo de trabajo del Centro Sismologico Nacional (CSN), con el fin de visualizar el peor de los casos de estimación de magnitud de un evento.

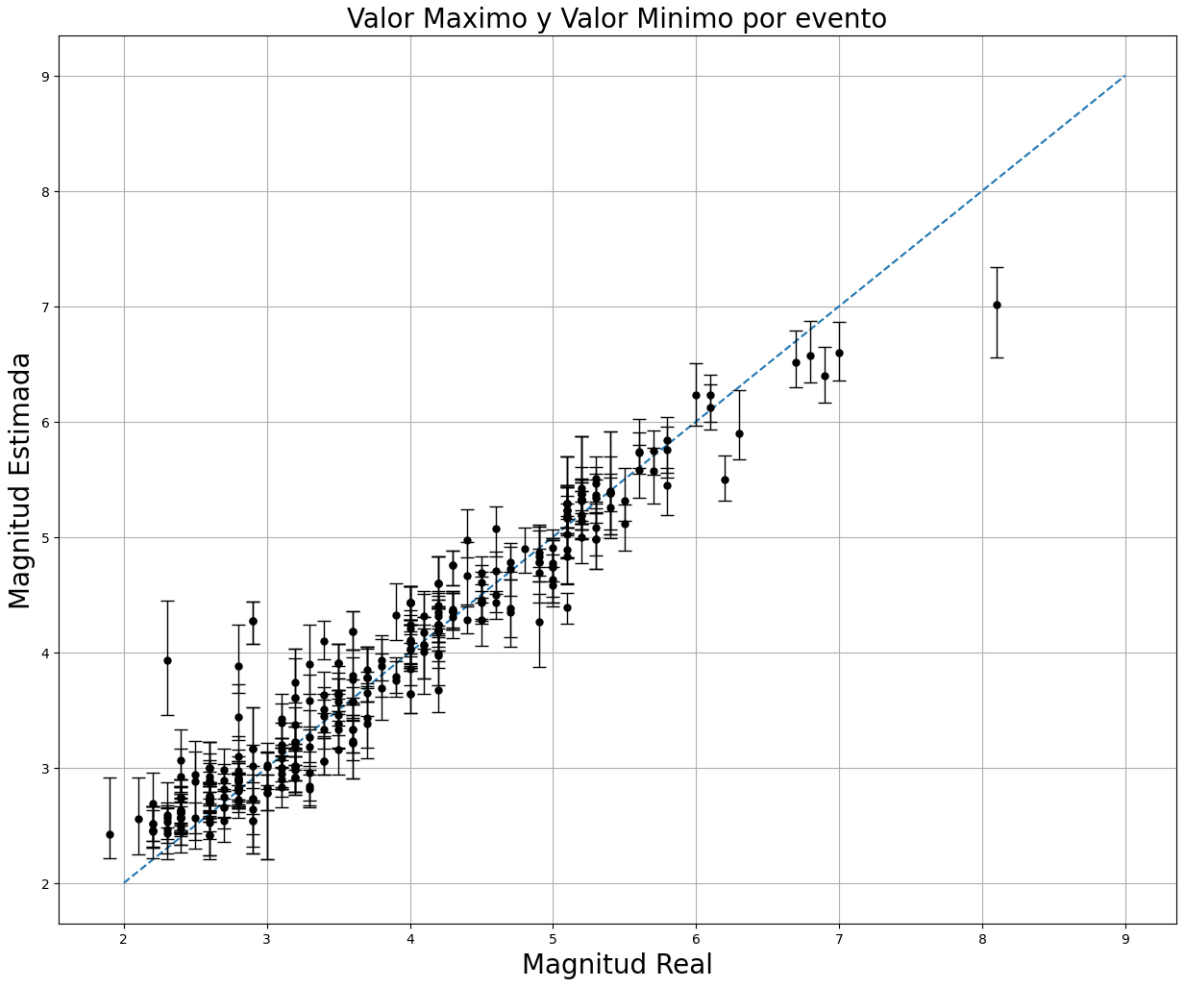


Figura 4.3: Magnitud estimada versus magnitud de referencia para todo el rango de magnitud. El punto representa la media entre 30 experimentos con su estimación máxima y mínima en las barras.

### 4.3.2. Magnitud mayor a 4

En la figura 4.4 se presenta la magnitud estimada v/s la magnitud real del sismo, donde al igual que para todas las magnitudes cada punto representa el promedio de 30 experimentos con su respectiva desviación estándar en las barras.

De la figura se observa que los datos se ajustan bastante bien a la recta, pero manteniendo una subestimación del evento más grande. Sin embargo, es importante destacar que al entrenar la red con este conjunto acotado de magnitudes (M $>$ 4), la red realiza una mejor estimación de los terremotos más grandes, logrando una menor subestimación y un valor de magnitud más cercano al real.

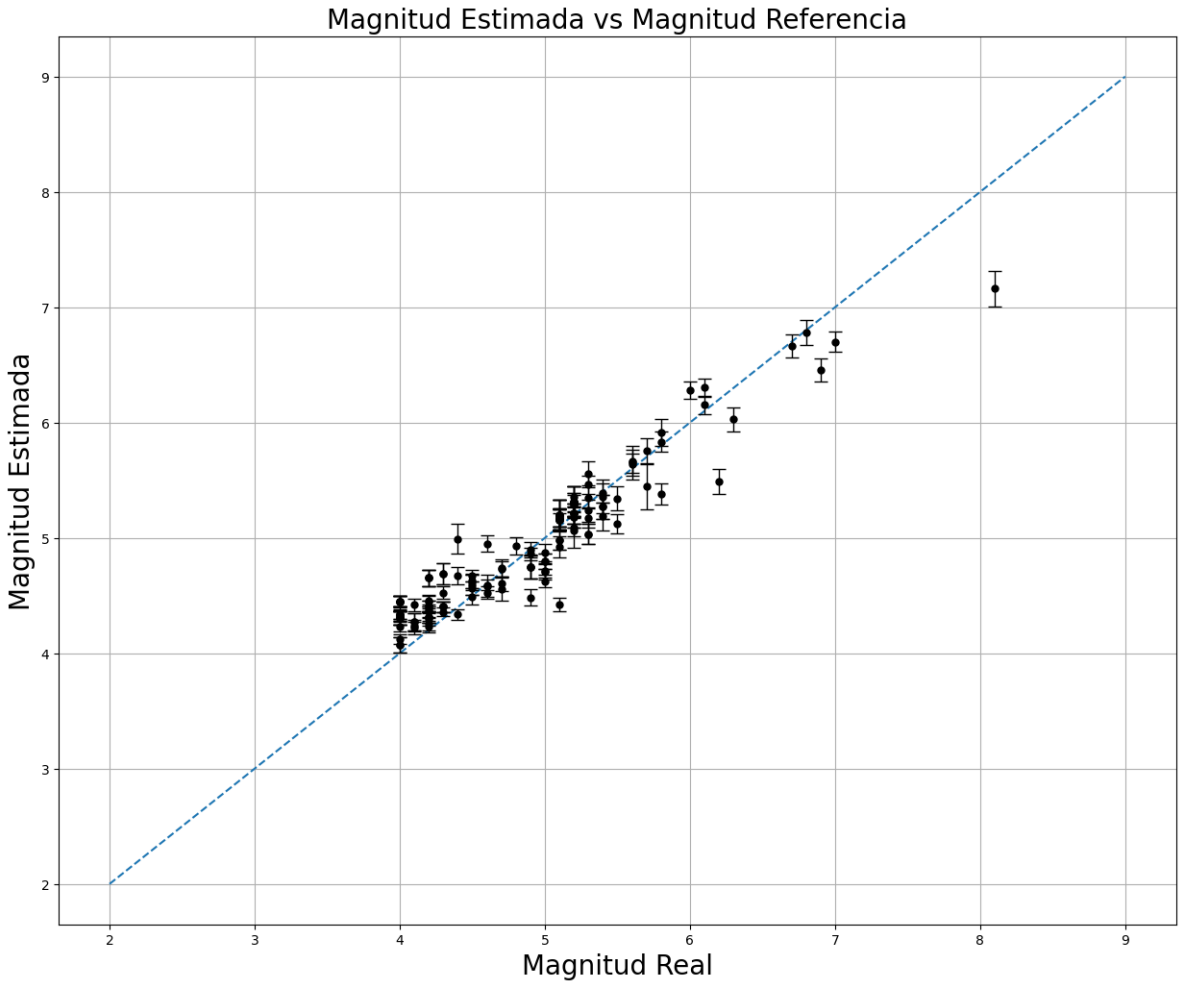


Figura 4.4: Magnitud estimada versus magnitud de referencia para magnitudes mayor a M4. El punto representa la media entre 30 experimentos con su Desviación estándar en barras.

En la figura 4.5 se presenta la magnitud estimada v/s la magnitud real con las estimaciones máximas y mínimas de cada evento para el conjunto de magnitudes mayor a 4. Lo más importante a resaltar de esta figura es que el evento mayor presenta una mejor estimación con una sub-estimación menor que en el caso de todas las magnitudes.

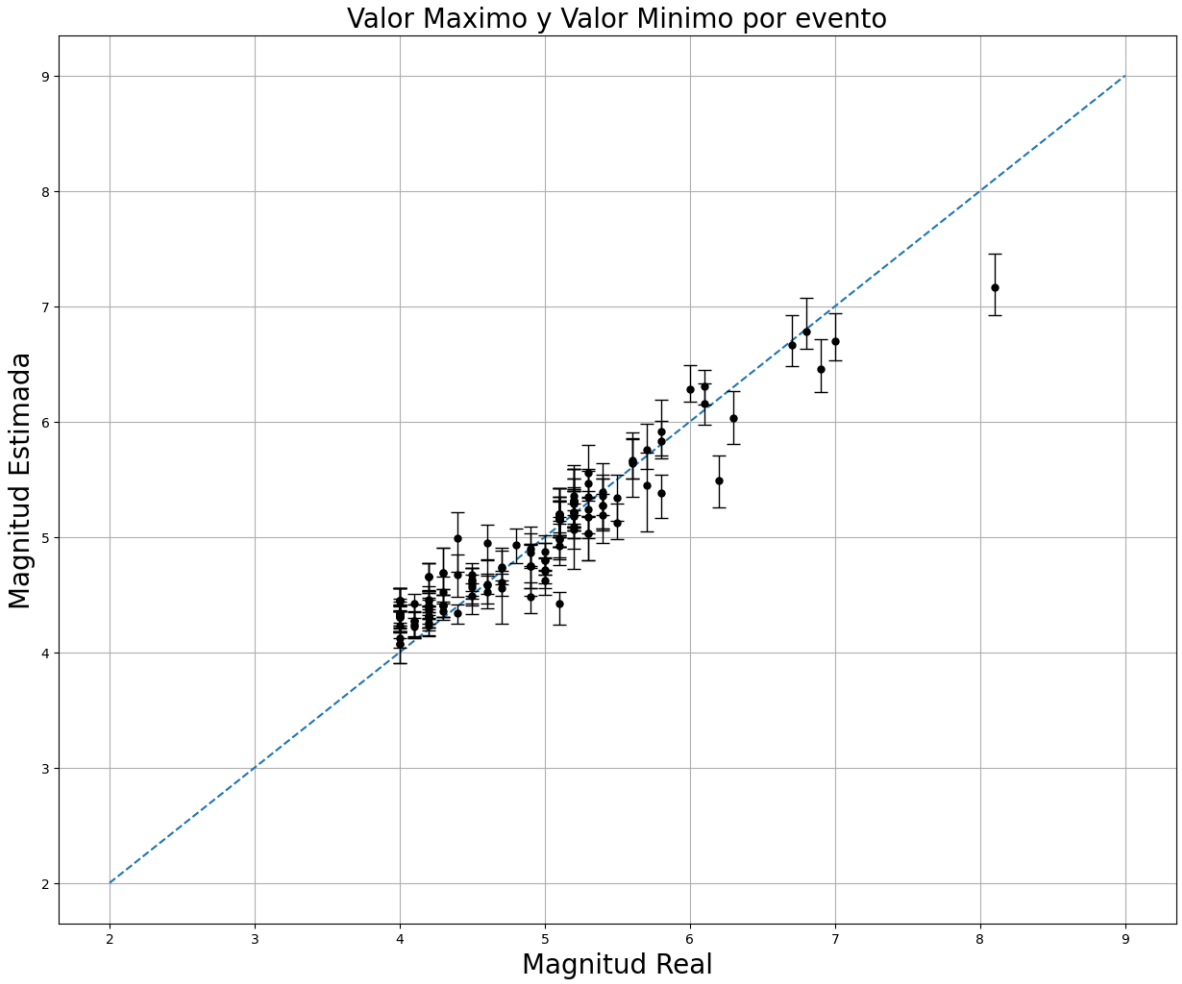


Figura 4.5: Magnitud estimada versus magnitud de referencia para magnitudes mayor a M4. El punto representa la media entre 30 experimentos con la estimación máxima y mínima en las barras.

### 4.3.3. Magnitud menor a 4

En la figura 4.6 se presenta la magnitud estimada v/s la magnitud real del sismo, donde al igual que para los conjuntos analizados previamente cada punto representa el promedio de 30 experimentos con su respectiva desviación estándar en las barras. Para este caso de análisis se observa un ajuste de los datos a la recta pero en menor grado que los segmentos anteriores. En este conjunto existen eventos que ensucian claramente el resultado, ademas los sismos menores a 2,5 M presentan una sobre estimación en su magnitud, lo cual demuestra la dificultad que resulta para la red estimar la magnitud de los eventos con bajo SNR.

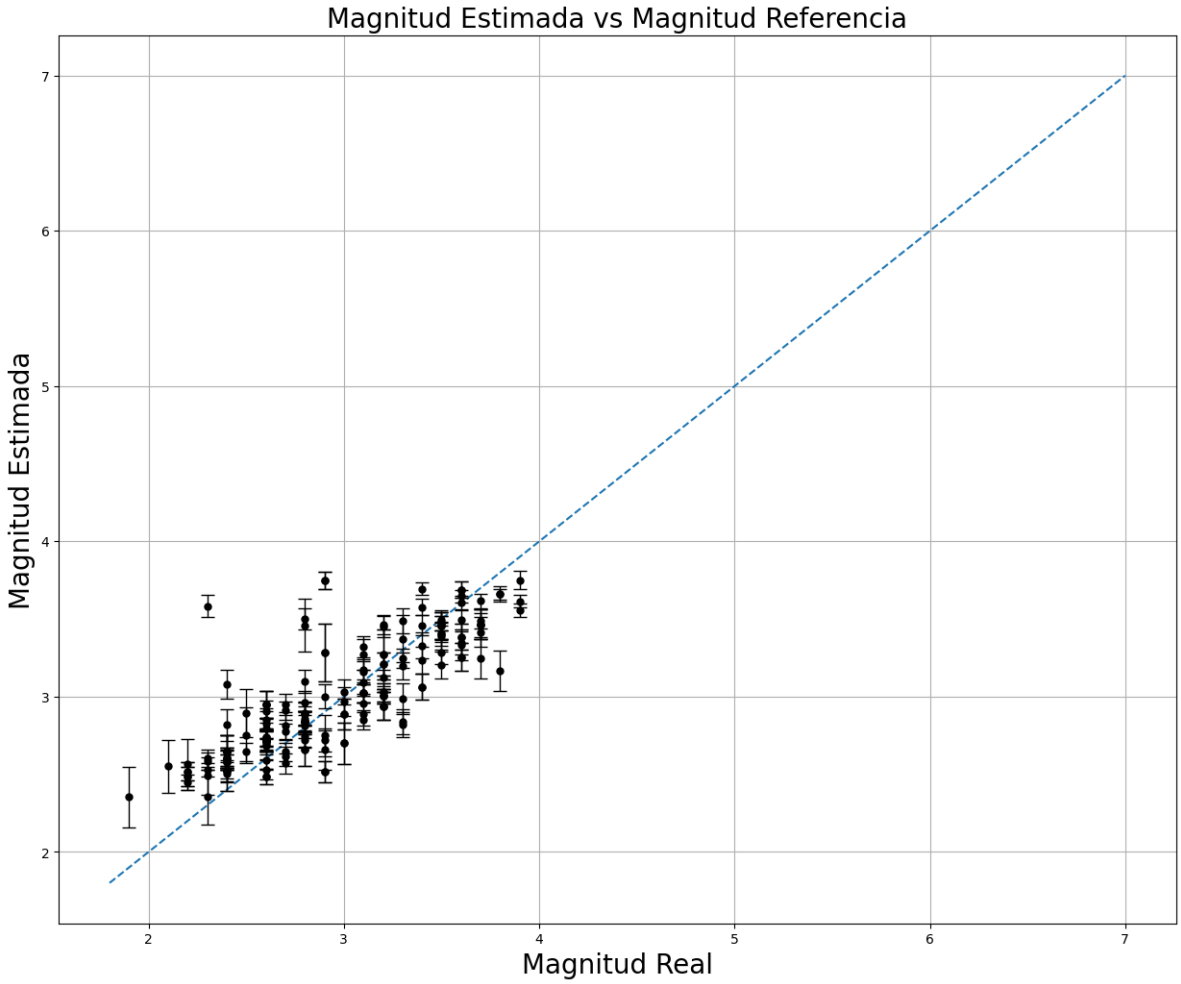


Figura 4.6: Magnitud estimada versus magnitud de referencia para magnitudes menores a M4. El punto representa la media entre 30 experimentos con su Desviación estándar en barras.

En la figura 4.7 se presenta la magnitud estimada v/s la magnitud real con las estimaciones máximas y mínimas de cada evento para el conjunto de magnitudes menor a 4. En general los datos se ajustan a la recta, pero con presencia de eventos de baja magnitud con una amplia diferencia en la estimación máxima y mínima. Para trabajos futuros se sugiere aumentar la cobertura incrementado la cantidad de estaciones de monitoreo. Además, es importante destacar que encontrar y etiquetar eventos de magnitud menor a 2,0 es una tarea difícil, principalmente por el bajo SNR de este tipo de eventos y por la distancia que hay a éstos. Por lo tanto, se propone mejorar la cobertura de estaciones con el fin aumentar el SNR para los eventos de baja magnitud.

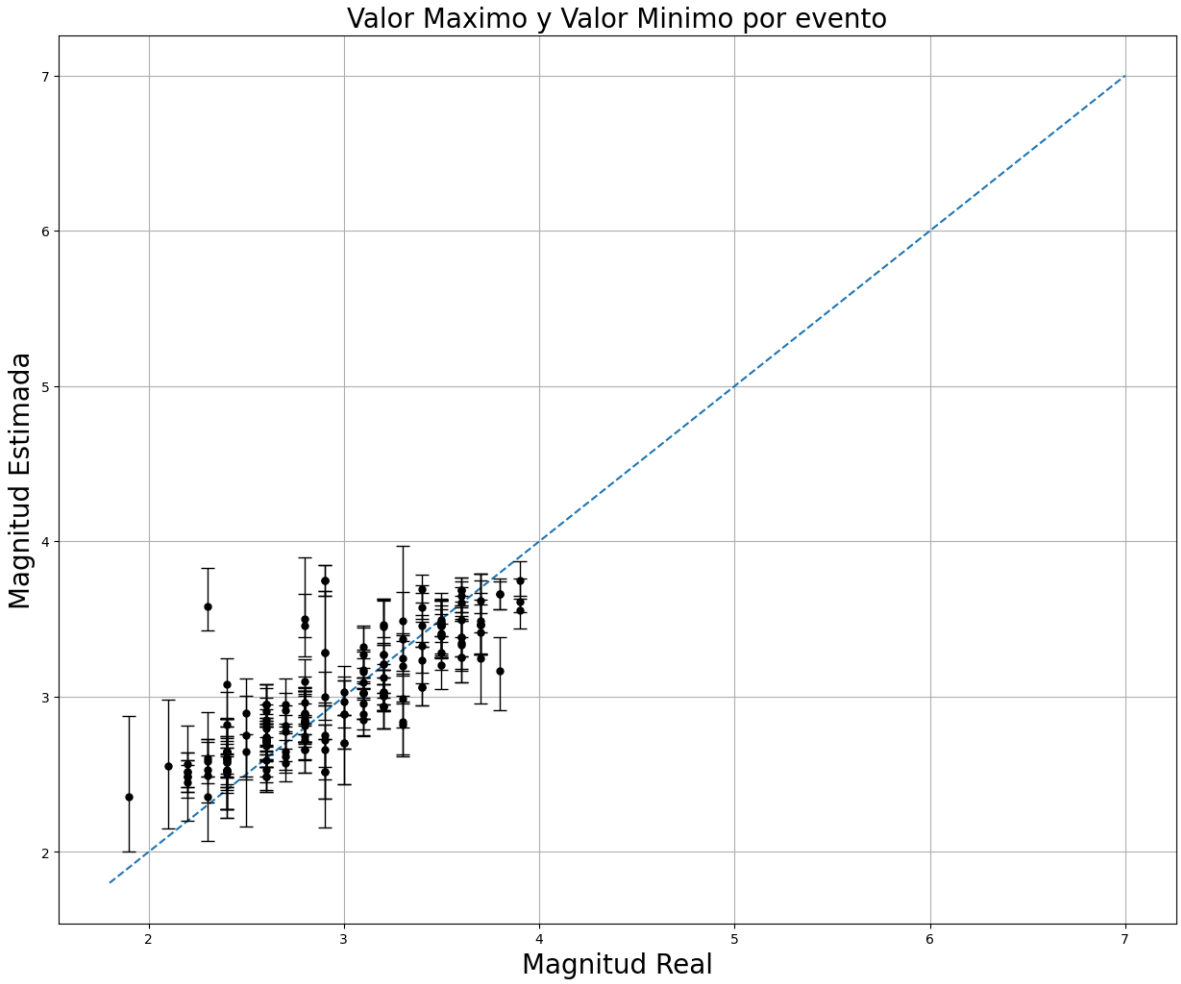


Figura 4.7: Magnitud estimada versus magnitud de referencia para magnitudes menor a M4. El punto representa la media entre 30 experimentos con la estimación máxima y mínima en las barras.

En la tabla 4.8 se presenta un resumen con los principales v/s de los distintos casos de análisis desarrollados en la presente investigación. Además, se muestran los mejores resultados obtenidos con los modelos configurados para magnitudes mayor y menor a 4.

Tabla 4.8: Resumen con las principales comparaciones realizadas en el trabajo de memoria.

| **Versus Interesantes** | **MAPE** | **MAPE** |
|---|---|---|
| Engineered Features v/s Raw Data | 6,48 - 6,81 | 27,83 |
| Global Features v/s Without Global F. | 6,48 | 10,98 |
| M > 4 v/s M < 4 | 3,93 | 7,26 |
| Energía v/s Sin Energía | 6,97 | 11,03 |

# Capítulo 5

# Conclusión

## 5.1. Conclusiones

Del trabajo de investigación realizado se concluye que el modelo basado en una red LSTM más Features Globales, obtiene errores de estimación de magnitud totalmente aceptable como una estimación preliminar de magnitud. Además, es importante destacar que el modelo es capaz de testear con nuevos eventos en el orden de segundos, mientras que los analistas las reportan en minutos, obteniendo errores similares. Por lo que se podría implementar con una herramienta de cálculo a la hora de recibir un sismo. Sin embargo, es necesario realizar más estudios y desarrollarlo para que sea aplicable en tiempo real.

El análisis realizado por segmentos de magnitud permitió identificar que la red neuronal realiza mejores estimaciones para terremotos de alta magnitud cuando es entrenada con sismos de magnitud mayor a 4. Por lo que sería un resultado interesante agregar una mayor cantidad de datos de esta índole. Sin embargo, para que haya registro de un sismo de este tipo, es necesario que primero ocurra un terremoto y que se tenga la mayor cantidad de registros para entrenar un modelo. Por otro lado, el análisis de micro-sismicidad y caracterización de clusters también es un tópico interesante de investigación, donde se sugiere utilizar el modelo en su versión entrenada con sismos $M < 4$.

El modelo que obtuvo mejores resultados corresponde a la red entrenada con sismos de magnitud mayor a M4, alcanzando el error MAPE de 3,93 % para el conjunto de test. Este modelo utiliza los features temporales de Fourier y Energía para alimentar el bloque LSTM-Bidireccional y a su salida se le concatena el feature global de ID-Estación con su representación en números enteros. Es importante mencionar que esta división por rango de magnitud mejoró la sub-estimación que presentaba el modelo entrenado con todos los rangos de magnitud. Sin embargo, no fue posible corregirla en un 100 % para eventos grandes sobre magnitud M7.5

Con los resultados y la velocidad de testeo de la red neuronal, es claro evidenciar que implementar un modelo basado en deep learning sería una buena herramienta de uso en un sistema de alerta temprana de terremoto (EWW) y alerta temprana de tsunamis (TEW) en territorio Chileno, lo cual facilitaría bastante el monitoreo realizado por el Centro Sismológico Nacional (CSN). Por lo que se propone continuar con la investigación, para trabajar en la implementación en tiempo real de los modelos.

## 5.2. Trabajos Futuros

- Se propone testear con el modelo en otras zonas del mundo y analizar su desempeño en otros contextos geológicos. Esto con el propósito de estudiar si hay relaciones entre distintos ambientes y si es posible entrenar un modelo de estimación de magnitud con datos de distintas zonas de la Tierra.

- Se propone incorporar sismos de gran magnitud (M >8.5) para analizar si es posible mejorar la sub-estimación que provoca la red para este tipo de eventos, los cuales coinciden con ser los más peligros y en los que se debe actuar de forma más rápida. Además, se propone agregar sismos de baja magnitud (1.8 < M) para abordar problemas de sobre-estimaciones en el caso contrario. Si bien este ultimo segmento no es de interes para alertas tempranas, si puede ser útil para comprender la micro-sismicidad de distintas zonas del territorio nacional.

- Se propone la experimentación con redes neuronales atencionales (transformers), las cuales en estudios recientes han demostrado tener un buen desempeño en problemas con datos secuenciales o temporales [51].

# Bibliografía


[1] Ruiz, S., Moreno, M., Melnick, D., Campo, F., Poli, P., Baez, J., Leyton, F., y Madariaga, R., "Reawakening of large earthquakes in south-central chile: The 2016 mw7.6 chiloé event," Geophysical Research Letters, 2017, doi:10.1002/2017GL074133.

[2] Ruiz, S. y Madariaga, R., "Historical and recent large megathrust earthquakes in chile," Tectonophysics, vol. 733, 2018, doi:10.1016/j.tecto.2018.01.015.

[3] Cremen, G. y Galasso, C., "Earthquake early warning: Recent advances and perspectives," Earth-Science Reviews, vol. 205, p. 103184, 2020.

[4] Kanamori, H., "Earthquake hazard mitigation and real-time warnings of tsunamis and earthquakes," Pure and Applied Geophysics, vol. 172, pp. 2335–2341, 2015.

[5] Saad, O. M., Hafez, A. G., y Soliman, M. S., "Deep learning approach for earthquake parameters classification in earthquake early warning system," IEEE Geoscience and Remote Sensing Letters, vol. 18, pp. 1293–1297, 2020.

[6] Festa, G., Picozzi, M., Caruso, A., Colombelli, S., Cattaneo, M., Chiaraluce, L., Elia, L., Martino, C., Marzorati, S., y Supino, M., "Performance of earthquake early warning systems during the 2016–2017 mw 5–6.5 central italy sequence," Seismological Research Letters, vol. 89, pp. 1–12, 2018.

[7] Colombelli, S., Caruso, A., Zollo, A., Festa, G., y Kanamori, H., "Ap wave-based, on-site method for earthquake early warning," Geophysical Research Letters, vol. 42, pp. 1390–1398, 2015.

[8] Cofré, A., Marín, M., Pino, O. V., Galleguillos, N., Riquelme, S., Barrientos, S., y Yoma, N. B., "End-to-end lstm-based earthquake magnitude estimation with a single station," IEEE Geoscience and Remote Sensing Letters, vol. 19, pp. 1–5, 2022, doi:10.1109/LGRS.2022.3175108.

[9] Heaton, J., "An empirical analysis of feature engineering for predictive modeling," pp. 1–6, 2016, doi:10.1109/SECON.2016.7506650.

[10] Kanamori, H., "A moment magnitude scale," Solid Earth, vol. 84, doi:https://doi.org/10.1029/JB084iB05p02348.

[11] Kanamori, H. y Anderson, D. L., "Theoretical basis of some empirical relations in seismology," Bulletin of the seismological society of America, vol. 65, pp. 1073–1095, 1975.

[12] Cybenko, G., "Approximation by superpositions of a sigmoidal function," Mathematics of Control, Signals and Systems, vol. 2, pp. 303–314, 1989, doi:10.1007/BF02551274.

[13] Schäfer, A. M. y Zimmermann, H. G., "Recurrent neural networks are universal approximators," pp. 632–640, Springer Berlin Heidelberg, 2006.

[14] Zhou, D.-X., "Universality of deep convolutional neural networks," Applied and Computational Harmonic Analysis, vol. 48, pp. 787–794, 2020, doi:https://doi.org/10.1016/j.acha.2019.06.004.

[15] Rosenblatt, F., "The perceptron: a probabilistic model for information storage and organization in the brain.," Psychological review, vol. 65, no. 6, p. 386, 1958.

[16] Hastie, T., Tibshirani, R., Friedman, J., Hastie, T., Tibshirani, R., y Friedman, J., "Overview of supervised learning," The elements of statistical learning: Data mining, inference, and prediction, pp. 9–41, 2009.

[17] Goodfellow, I., Bengio, Y., y Courville, A., Deep Learning. MIT Press, 2016. http://www.deeplearningbook.org.

[18] ichi Amari, S., "Backpropagation and stochastic gradient descent method," Neurocomputing, vol. 5, pp. 185–196, 1993, doi:https://doi.org/10.1016/0925-2312(93)90006-O.

[19] Bottou, L., "Stochastic gradient descent tricks," Neural Networks: Tricks of the Trade: Second Edition, pp. 421–436, 2012.

[20] Duchi, J. y Hazan, E., "Adaptive subgradient methods for online learning and stochastic optimization," Journal of Machine Learning Research, vol. 12, pp. 2121–2159, 2011.

[21] Tieleman, T. y Hinton, G., "Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude," COURSERA: Neural networks for machine learning, vol. 4, pp. 26–31, 2012.

[22] Kingma, D. P. y Ba, J., "Adam: A method for stochastic optimization," 2014, https://arxiv.org/abs/1412.6980.

[23] Werbos, P. J., "Beyond regression: New tools for prediction and analysis in the behavioral sciences. ph. d. thesis, harvard university, cambridge, ma, 1974," 1974.

[24] Rumelhart, D. E., Hinton, G. E., y Williams, R. J., "Learning representations by back-propagating errors," Nature, vol. 323, pp. 533–536, 1986, doi:10.1038/323533a0.

[25] Murtagh, F., "Multilayer perceptrons for classification and regression," Neurocomputing, vol. 2, pp. 183–197, 1991.

[26] Gardner, M. W. y Dorling, S. R., "Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences," Atmospheric environment, vol. 32, pp. 2627–2636, 1998.

[27] Hijazi, S., Kumar, R., y Rowen, C., "Using convolutional neural networks for image recognition," Cadence Design Systems Inc.: San Jose, CA, USA, vol. 9, p. 1, 2015.

[28] Collobert, R. y Weston, J., "A unified architecture for natural language processing: Deep neural networks with multitask learning," en Proceedings of the 25th International Conference on Machine Learning, ICML '08, (New York, NY, USA), p. 160–167, Association for Computing Machinery, 2008, doi:10.1145/1390156.1390177.

[29] Wang, W. y Gang, J., "Application of convolutional neural network in natural language processing," pp. 64–70, 2018, doi:10.1109/ICISCAE.2018.8666928.

[30] Abdel-Hamid, O., r. Mohamed, A., Jiang, H., Deng, L., Penn, G., y Yu, D., "Convolutional neural networks for speech recognition," IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 22, pp. 1533–1545, 2014, doi:10.1109/TASLP.2014.2339736.

[31] Hochreiter, S., "Untersuchungen zu dynamischen neuronalen netzen," 1991.

[32] Bengio, Y., Simard, P., y Frasconi, P., "Learning long-term dependencies with gradient descent is difficult," IEEE Transactions on Neural Networks, vol. 5, pp. 157–166, 1994, doi:10.1109/72.279181.

[33] Hochreiter, S., Bengio, Y., Frasconi, P., y Schmidhuber, J., "Gradient flow in recurrent nets: the difficulty of learning long-term dependencies," 2001.

[34] Lipton, Z. C., "A critical review of recurrent neural networks for sequence learning," CoRR, vol. abs/1506.00019, 2015, http://arxiv.org/abs/1506.00019.

[35] Hochreiter, S. y Schmidhuber, J., "Long short-term memory," Neural Computation, vol. 9, pp. 1735–1780, 1997, doi:10.1162/neco.1997.9.8.1735.

[36] Olah, C., "Understanding lstm networks," 2015, http://colah.github.io/posts/2015-08-Understanding-LSTMs/.

[37] Nikolaevich, M., "Machine learning modelling and feature engineering in seismology experiment,", doi:https://doi.org/10.3390/s20154228.

[38] Iaccarino, A., "Earthquake early warning system for structural drift prediction using machine learning and linear regressors," 2021, doi:10.3389/feart.2021.666444.

[39] Li, Z., "Machine learning seismic wave discrimination: application to earthquake early warning," 2018, doi:10.1029/2018GL077870.

[40] Kuyuk, H., "Real-time classification of earthquake using deep learning," 2018, doi:10.1016/j.procs.2018.10.316.

[41] Perol, T., "Convolutional neural network for earthquake detection and location," 2018, doi:10.1126/sciadv.1700578.

[42] Ristea, N. C. y Radoi, A., "Complex neural networks for estimating epicentral distance, depth, and magnitude of seismic waves," IEEE Geoscience and Remote Sensing Letters, vol. 19, pp. 1–5, 2022, doi:10.1109/LGRS.2021.3059422.

[43] Mousavi, S. M., Sheng, Y., Zhu, W., y Beroza, G. C., "Stanford earthquake dataset (stead): A global data set of seismic signals for ai," IEEE Access, 2019.

[44] Meier, A., "The gutenberg algorithm: Evolutionary bayesian magnitude estimates for earthquake early warning with a filter bank," 2015, doi:10.1785/0120150098.

[45] Kundu, A., Bhadauria, Y. S., Basu, S., y Mukhopadhyay, S., "Artificial neural network based estimation of moment magnitude with relevance to earthquake early warning," pp. 1955–1959, IEEE, 2017.

[46] Majstorović, J., Giffard-Roisin, S., y Poli, P., "Designing convolutional neural network pipeline for near-fault earthquake catalog extension using single-station waveforms," Journal of Geophysical Research: Solid Earth, vol. 126, p. e2020JB021566, 2021.

[47] Ochoa, L. H., Niño, L. F., y Vargas, C. A., "Fast magnitude determination using a single seismological station record implementing machine learning techniques," Geodesy and Geodynamics, vol. 9, pp. 34–41, 2018.

[48] Krischer, L., Megies, T., Barsch, R., Beyreuther, M., Lecocq, T., Caudron, C., y Wassermann, J., "Obspy: A bridge for seismology into the scientific python ecosystem," Computational Science Discovery, vol. 8, p. 014003, 2015.

[49] Cooley, J. W. y Tukey, J. W., "An algorithm for the machine calculation of complex fourier series," Mathematics of Computation, vol. 19, pp. 297–301, 1965.

[50] Kennett, B. L. N. y Engdahl, E. R., "Traveltimes for global earthquake location and phase identification," Geophysical Journal International, vol. 105, pp. 429–465, 1991, doi:10.1111/j.1365-246X.1991.tb06724.x.

[51] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Łukasz Kaiser, y Polosukhin, I., "Attention is all you need," Advances in neural information processing systems, vol. 30, 2017.