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

IMPLEMENTACIÓN DE MÉTODOS BASADOS EN DEEP LEARNING PARA LOCALIZACIÓN DE EVENTOS SÍSMICOS DE ORIGEN VOLCÁNICO

MEMORIA PARA OPTAR AL TÍTULO DE
INGENIERO CIVIL ELÉCTRICO

JORGE EMILIO CELIS MARÍN

PROFESOR GUÍA:
NÉSTOR BECERRA YOMA

MIEMBROS DE LA COMISIÓN:
JORGE WUTH SEPÚLVEDA
FRANCISCO RIVERA SERRANO

SANTIAGO DE CHILE
2021

RESUMEN DE LA MEMORIA PARA OPTAR
AL TÍTULO DE INGENIERO CIVIL ELÉCTRICO
POR: JORGE EMILIO CELIS MARÍN
FECHA: 2021
PROF. GUÍA: NÉSTOR BECERRA YOMA

IMPLEMENTACIÓN DE MÉTODOS BASADOS EN DEEP LEARNING PARA LOCALIZACIÓN DE EVENTOS SÍSMICOS DE ORIGEN VOLCÁNICO

La localización de eventos sísmicos de origen volcánico es una tarea de importancia para la evaluación de riesgos y para el estudio de las estructuras internas de un volcán. Los métodos más comúnmente usados en la actualidad se basan en la detección de fases de ondas sísmicas que, debido a la alta sensibilidad de la localización respecto a estas, mantienen una etapa de inspección visual por parte de expertos.

El presente trabajo expone el estudio y los resultados de la aplicación de métodos de redes neuronales profundas (*Deep Learning*, DL) para la localización de epicentro de eventos volcánicos provenientes del Volcán Chillán, sin la necesidad de detección de ondas (*end-to-end*). Las arquitecturas exploradas fueron redes convolucionales (CNN) y redes recurrentes, en particular, LSTM. Las señales de entrada a las redes fueron preprocesadas usando la Transformada de Fourier de Tiempo Reducido (STFT), que demostró ser un buen compromiso entre representación temporal y frecuencial.

Se prosigue con una comparación de los resultados, tanto entre las arquitecturas exploradas como contra un método de detección automático, consistente en una integración de un detector automático (*PhasePicker*) y un programa localizador en base a picado de ondas (HYPO71). Considerando como métrica de rendimiento el porcentaje de errores menores a 1 km LSTM consigue mejores resultados (49,351 %) que CNN (44,156 %) y, ambos, sobrepasan de manera considerable al método automático (1,37 %).

Dedicado a Sofía.

Tabla de Contenido

Índice de Tablas

Índice de Ilustraciones

Introducción

Los eventos volcánicos son fenómenos naturales que pueden tener importantes consecuencias para las comunidades y ciudades que viven en zonas cercanas a ellos. Un monitoreo de sus señales permite categorizar el estado de un volcán para implementar sistemas de alertas de acuerdo con la peligrosidad estimada y, por otro lado, sirve para mejorar el entendimiento de la estructura interna del volcán.

Chile cuenta con una extensa red de volcanes a lo largo de la Cordillera de los Andes y 90 de ellos son considerados activos y monitoreados por el Servicio Nacional de Geología y Minería (SERNAGEOMIN) mediante su Observatorio Volcanológico de los Andes Sur (OVDAS). El Volcán Chillán es el cuarto más activo y en este se han efectuado una serie de estudios para apoyar las tareas de monitoreo por parte de un grupo de estudios integrado por Universidad de la Frontera, OVDAS Temuco, Universidad de Santiago y Universidad de Chile. Dentro de los temas abordados están la clasificación de eventos y, lo que el tema de memoria aborda, la localización automática de estos eventos.

Los métodos tradicionales de localización son dependientes de la detección de inicios de onda y de un modelo de velocidad del terreno. Por su alta sensibilidad en los resultados, la tarea de detección continúa siendo monitoreada visualmente por expertos en una tarea demandante de tiempo. Para enfrentar el problema se propone la utilización de metodología *end-to-end* mediante arquitecturas de *Deep Learning* (DL) como una estrategia para disminuir la dependencia de variables específicas y de interacción humana.

El informe se organiza de la siguiente manera:

1. En la sección 1 se cubre la base teórica del problema (1.1, 1.2 y 1.3) y se hace un estudio de los métodos de localización tradicionales y el estado del arte (1.4).
2. En la sección 2 se detalla la implementación del modelo propuesto describiendo la base de datos, la metodología a usar, las arquitecturas consideras y el espacio de búsqueda de hiperparámetros. Luego, se describen en detalle el preprocesamiento y los tipos de arquitectura usadas.
3. Los resultados de lo anterior son expuestos en la sección 3, en donde se realiza un análisis y comparación con los métodos expuestos en la sección 1.
4. Por último, en la sección 4, se discuten las ventajas y limitaciones tanto del modelo como del proyecto, además de proponer trabajos futuros para mejorar el entrenamiento del modelo. Se finaliza con una conclusión que resume los resultados obtenidos y estas discusiones.

Capítulo 1

Marco Teórico

1.1. Ondas Sísmicas

Las ondas sísmicas son ondas mecánicas que viajan por la corteza terrestre originadas por grandes movimientos de tierra y liberación de energía, de acuerdo con la tectónica de placas u otros fenómenos como desprendimientos o fracturas en volcanes y explosiones artificiales. La primera gran división de categorías en el estudio de estas ondas está relacionada con la zona de propagación y, en ella, se identifican a las ondas internas y las superficiales.

Las ondas internas son las que se propagan desde el origen del movimiento (hipocentro) hasta la superficie y las superficiales son los movimientos que se suceden cuando las ondas internas llegan hasta la superficie. Para el contexto de detección y localización de eventos sísmicos, así como en el estudio del presente trabajo de título, las ondas con la información relevante, y que serán las estudiadas, son las ondas internas.

A su vez, las ondas internas se subdividen en las llamadas ondas Primaria (P) y Secundaria (S). Las ondas P corresponden a ondas longitudinales y las S a ondas transversales. Las formas de las soluciones de la ecuación de onda que dan origen a estas dos variantes predicen velocidades de propagación diferentes, siendo las ondas P más rápidas y por tanto primeras en ser detectadas (de ahí los nombres).

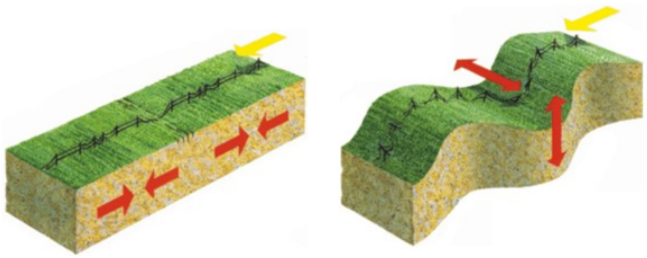

Figura 1.1: Ilustración de las ondas P (longitudinal, izquierda) y S (transversal, derecha)

Las ondas sísmicas son capturadas para su estudio por un sismógrafo. En general, un

modelo para la señal efectivamente capturada por un sismógrafo viene dada por la relación 1.1:

$$u(t) = s(t) * g(t) * \mathrm{i}(t) \tag{1.1}$$

En donde $u(t)$ corresponde a la señal capturada, $s(t)$ a la fuente sísmica, $g(t)$ a la atenuación producto del camino (*raypath*), $\mathrm{i}(t)$ es el efecto del sensor y $*$ es el operador convolución [4].

1.1.1. Ondas Sísmicas de Origen Volcánico

Un volcán es una formación geológica conectada a reservas de material fundido en las profundidades de la corteza terrestre. Debido a su conexión con material de alta temperatura, muchos de estos presentan actividad más o menos regular en donde se manifiestan diversos tipos de movimientos y liberación de energía. El material eyectado por el cráter se acumula y forma un edificio volcánico.

Las actividades de los volcanes se han clasificado según su naturaleza de origen y según sus características en los registros sismográficos, no necesariamente coincidiendo en la cantidad de clases. Los eventos de interés para este trabajo son los Volcano-Tectonic (VT) según la clasificación en [23], que sigue criterios de mecanismo de origen y es de las clasificaciones estándar.

Según esta clasificación, los VT son eventos cuyo origen obedece a la fractura de rocas producto de las presiones de las corrientes de magma. Al suceder la fractura, la presión acumulada se libera rápidamente generando ondas sísmicas tipo P y, en algunos casos, S, en las que se puede identificar un claro comienzo y son, por lo tanto, eventos cuyo registro permite utilizar métodos de localización similares a los de los sismos tectónicos.

Dependiendo de la profundidad, los eventos VT pueden subdividirse en VT-A y VT-B. Los VT-A corresponden a sucesos profundos (>2km) de relativa alta frecuencia (>5Hz) y en los que se suele detectar claramente ondas P y S. Por su parte, los VT-B se producen a poca profundidad (entre 1-2 km), tienen frecuencia dominante entre los 1 y 5 Hz y sus ondas P y S son relativamente lentas o emergentes, muchas veces sin poder identificar un claro inicio para la onda S.

Las señales provenientes de fuentes volcánicas son considerablemente más difíciles de procesar y caracterizar que las provenientes de sismos tectónicos. «Esto porque las fuentes sísmicas de origen volcánico envuelven dinámica de gases, fluidos y sólidos, los caminos de propagación son usualmente extremadamente heterogéneos, anisotrópicos y absorbentes, con topografía irregular e interfaces que incluyen rupturas de todas las escalas y orientaciones. Por lo tanto, la sismología volcánica es la más desafiante, requiriendo ingenio en el diseño de experimentos y en las interpretaciones de las observaciones.» [1].

1.2. Procesamiento de Señales

1.2.1. Dominio de la Frecuencia

Las transformaciones de una señal temporal para llevarla a una representación en el dominio de la frecuencia suelen servir para simplificar el estudio al descomponer los componentes de frecuencia según su aporte a la energía total, lo que permite concentrar el análisis de características en las bandas dominantes y descartar o filtrar componentes de ruido. La transformada canónica es la transforma de Fourier y su aplicación es un procedimiento estándar en el estudio de cualquier señal, se tenga o no presupuestado su tratamiento en este dominio, pues es esencial para la comprensión de las características de esta.

1.2.2. Transformada Discreta de Fourier (DFT)

Tanto la transforma de Fourier como la DFT se basan en la idea de que una señal en el tiempo se puede descomponer como la suma de una familia de funciones sinusoidales que forman una base ortogonal del espacio de funciones.

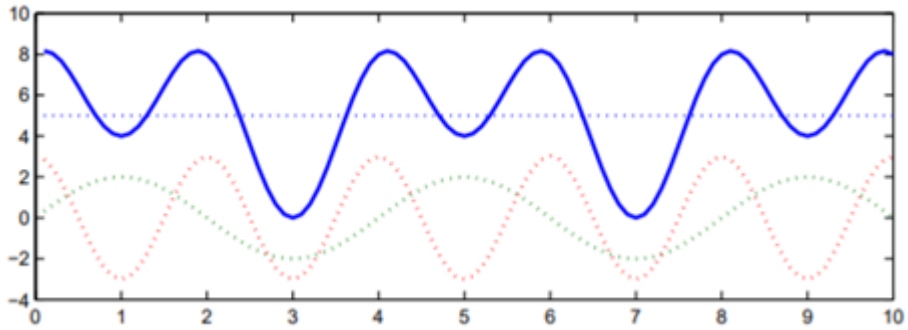

Figura 1.2: Ejemplo de descomposición de una señal (azul)

Una forma de describir el procedimiento de cálculo de la DFT es considerar a esta transformada como una medida de la correspondencia de la señal con una familia de funciones sinusoidales. Esta correspondencia se mide como el producto interno en un espacio de funciones y es la que determina el factor o escalar en la suma vectorial de las funciones base que sirven de constructor para la señal en estudio.

Dada a una señal de tiempo discreto de largo N ($x[n]$ con $n \in \{0, ..., N-1\}$), la componente de frecuencia normalizada $\frac{2\pi k}{N}$ se define según 1.2:

$$X[k] = \sum_{n=0}^{N-1} x[n] \cdot \mathrm{e}^{-\frac{\mathrm{i}2\pi}{N}kn} \tag{1.2}$$

1.2.3. Transformada de Fourier de Tiempo Reducido (STFT)

El análisis en el dominio de frecuencias es una poderosa herramienta para hacer explícitas propiedades de la señal en estudio, pero estas propiedades son representativas de toda la

señal y, son por tanto, no localizables directamente en el tiempo; no se sabe en qué momento tal o cual componente de frecuencia estuvo más activa. Esto es un problema para señales no estacionarias pues se pierde la capacidad de distinguir distintas etapas de estas.

Este problema es muy relevante en el contexto de localización de fuente emisora porque los métodos se basan explícitamente en la detección temporal de patrones y la diferencia de tiempo entre estos. Una solución para recobrar resolución temporal parcial es la introducción de la STFT.

La STFT es la aplicación de la DFT a una serie de ventanas de tiempo de la señal. Para conseguir el efecto de enventanado, la señal original es convolucionada por una función ventana (*windowing*) de forma tal que las muestras más allá de la ventana no contribuyan al cálculo de la DFT en cada iteración. Adicionalmente, es deseable que la función de enventanado tenga buenas propiedades de suavizado para evitar artefactos de alta frecuencia producto de cortar la señal. De esta manera, para una ventana centrada en la muestra 'm', la nueva DFT se calcula según 1.3:

$$X[k] = \sum_{n=0}^{N-1} x[n]w[n-m] \cdot \mathrm{e}^{-\frac{\mathrm{i}2\pi}{N}kn} \tag{1.3}$$

1.2.4. *Trade-Off* Tiempo-Frecuencia

Los bines de frecuencia abarcan un ancho de banda igual a $\frac{F_s}{N}$, siendo F_s la frecuencia de muestreo y N la cantidad de muestras. Esto se puede considerar como la «resolución en frecuencia», pues es la cantidad mínima en la que deben diferenciarse dos componentes de la señal original para poder ser detectadas como frecuencias diferentes en la DFT.

Por su parte, la «resolución en tiempo» se puede definir, directamente, como el ancho de la ventana pues es la diferencia, en tiempo, mínima para que dos efectos no estén siendo parte de la misma DFT.

Debido a que la resolución en frecuencia se relaciona de manera directa con la cantidad de muestras, pero la resolución temporal de manera inversa, sucede que en el análisis de STFT siempre se tiene un *trade-off* entre estas resoluciones.

Este *trade-off* debe ser estudiado caso a caso y es, por tanto, parte de los parámetros a estudiar para encontrar la representación óptima para el objetivo buscado. En la figura 1.3 se representa gráficamente este efecto.

1.3. Redes Convolucionales y Recurrentes

1.3.1. Red Neuronal Artificial (MLP)

Una unidad de neurona artificial se puede definir como una función no lineal sobre el producto de matrices, con una representación gráfica útil y que permite una composición de más unidades y capas de manera fácil. Para una entrada $x = [x_1, ..., x_n]$, una unidad queda

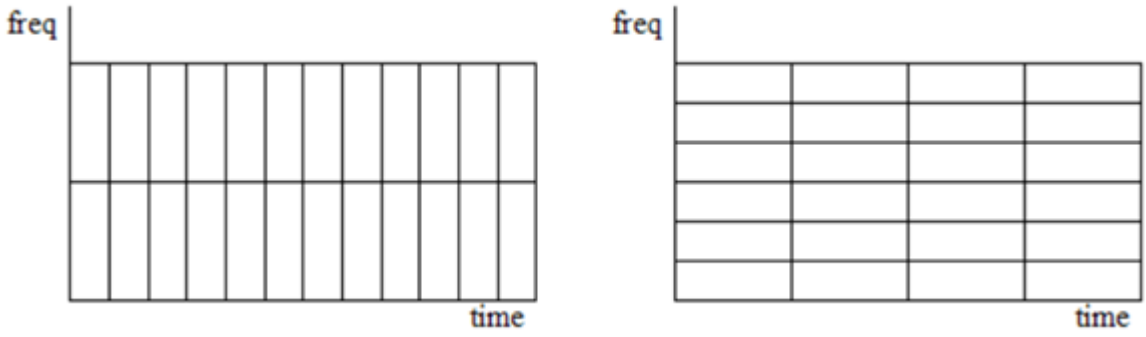

Figura 1.3: Representación del trade-off de resoluciones temporal y frecuencial para una STFT

definida por su vector de pesos $w = [w_1, ..., w_n]$, su valor de sesgo b y por su función de activación, en donde los pesos y el sesgo son parámetros entrenables (ver figura 1.4).

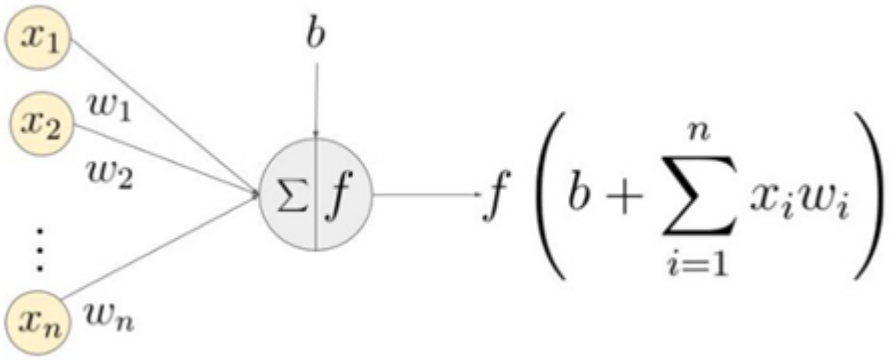

Figura 1.4: Representación gráfica de una neurona artificial

Que un parámetro sea entrenable significa que, para una determinada combinación de unidades y capas (arquitectura) y dadas una entrada y una función objetivo, estos serán los parámetros modificables encargados de optimizar la red bajo alguna métrica y siguiendo un proceso de descenso del gradiente.

El Teorema de Aproximación Universal demuestra que una red de una capa, con suficientes unidades y que incluya alguna no linealidad, es capaz de aproximar con error arbitrario a cualquier función continua y acotada que vaya desde R^n a R^m. No obstante lo anterior, este resultado teórico solo permite asegurar la existencia, con lo que el número de unidades requerido puede ser prohibitivamente grande y su entrenamiento puede resultar demasiado difícil.

Ante esta dificultad práctica surgen las arquitecturas con «capas ocultas» o profundas: Redes en las que capas de unidades son concatenadas formando una gran función compuesta, parametrizable y diferenciable por partes. Esto permite que la aproximación a una función se realice por etapas, lo que reduce el número total de parámetros requeridos. A las redes que incluyen capas de unidades, como la de la figura 2.5, se le conoce como Perceptrón multicapa

(MLP).

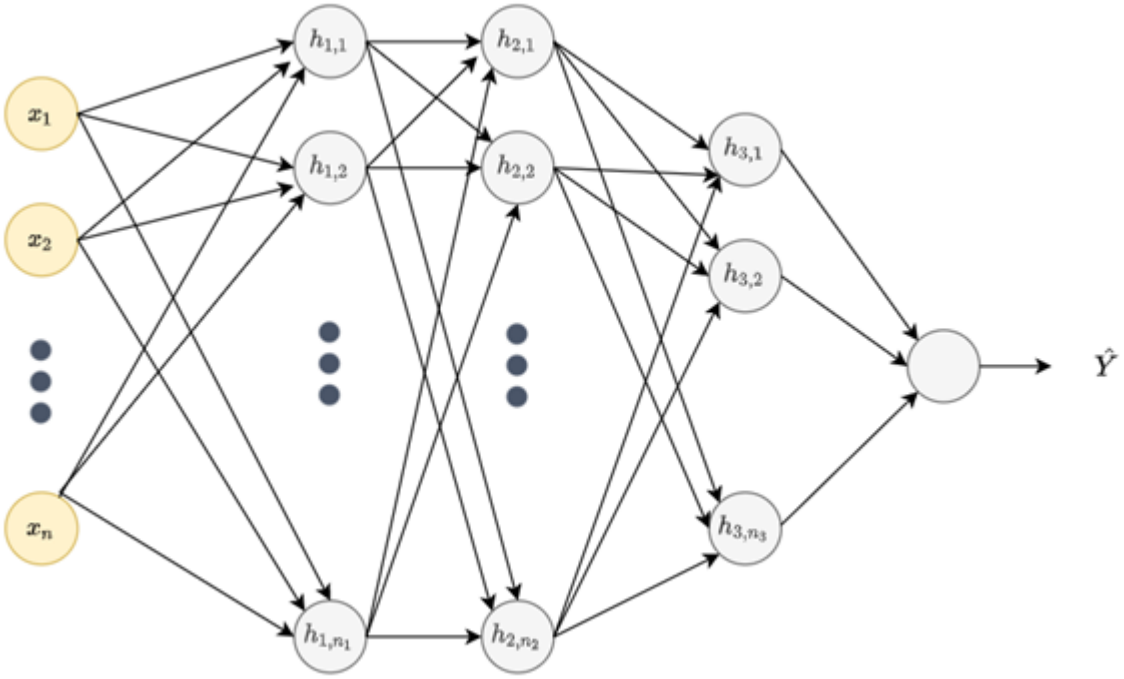

Figura 1.5: Representación gráfica de una red MLP

A pesar de su inspiración y pretensión inicial, las redes neuronales artificiales mantienen tan solo una analogía superficial con el funcionamiento básico de una neurona biológica. Muchos de los desarrollos posteriores a la introducción de la considerada primera red neuronal artificial, el Perceptrón, han ido alejando la analogía aún más en aras del funcionamiento práctico.

Como ejemplo de lo anterior se puede mencionar la adopción de la función de activación RELU por sobre la sigmoidal, a pesar de que la segunda fue propuesta por guardar mayor cercanía con los modelos de una neurona biológica o la implausibilidad biológica del algoritmo de aprendizaje *backpropagation*. Siguiendo a [11] «es mejor pensar en las redes neuronales como máquinas aproximadoras de funciones diseñadas para alcanzar la generalización estadística».

1.3.2. Redes Convolucionales

Un desarrollo que sí se puede considerar como un ejemplo exitoso de sinergia entre la inspiración biológica y las redes artificiales, son las Redes Neuronales Convolucionales (CNN, por sus siglas en inglés).

Los estudios de Hubel y Wissel sobre el funcionamiento de la corteza visual en mamíferos, en los que descubrieron que distintos arreglos de neuronas se especializaban para detectar patrones específicos en regiones limitadas del campo visual, que luego eran procesados siguiendo una conexión jerárquica, sirvieron de inspiración para la primera CNN llamada Neocognitron.

Tanto por la inspiración biológica mencionada como por su capacidad de lidiar eficiente-

mente con entradas de alta dimensionalidad gracias a su uso compartido de parámetros, las CNN han estado ligadas a aplicaciones de visión computacional obteniendo resultados considerablemente por sobre otros métodos tradicionales en áreas como reconocimiento de rostros, vehículos autónomos, reconocimiento de texto, etc. Es por esto que la mayoría de arquitecturas y aplicaciones, incluyendo el presente trabajo, trabajan con entradas tipo imágenes -2 dimensiones y uno o más canales- y en la siguiente sección se explicará la arquitectura de una red CNN para este tipo de entradas considerando un solo canal por simplicidad.

1.3.2.1. Arquitectura de una red CNN 2D

La CNN es una red secuencial (no recurrente) en donde la información de la capa de entrada es procesada consecutivamente por capas convolucionales, seguidas por funciones de activación y, opcionalmente, capas de *pooling* y/o regularizadoras. Generalmente, antes de la capa de salida se pasa a una representación unidimensional (*flattening*) para poder ser procesada por una capa densa que se conecta con la capa de salida.

Entre las técnicas de regularización más ampliamente utilizadas en la actualidad se encuentra el «*Dropout*», que consiste en asignar valor 0 («desactivar») aleatoriamente distintas conexiones o pesos de la red. El conjunto de conexiones desactivadas cambia en cada iteración. Con esto se logra una mayor robustez ante entradas anómalas y mejor generalización.

Una capa convolucional consiste en un número de matrices o filtros que son convolucionados discretamente con la imagen de entrada o la obtenida de la capa anterior. Al resultado de cada operación se le aplica una función de activación y es concatenado, formando una nueva imagen, de tantos canales como filtros, que será procesada por la siguiente capa.

1.3.2.2. Convolución Discreta

La convolución discreta (en rigor, lo que se define a continuación es el operador de correlación cruzada, pero por jerga se sigue usando convolución en este contexto) es un operador entre funciones y su producto es otra función. Para el caso de 2 dimensiones y dos funciones arbitrarias queda definida por 1.4.

$$I \circ F(x, y) = \sum_{j=-\infty}^{\infty} \sum_{\mathrm{i}=-\infty}^{\infty} F(\mathrm{i}, j) I(x + \mathrm{i}, y + j) \tag{1.4}$$

Para este contexto, las entradas y filtros se pueden considerar como funciones bidimensionales discretas con valores igual a 0 para todo elemento fuera de los radios definidos por el largo y ancho de estos. Así, si consideramos una imagen I y un filtro F de dimensiones impares (2N+1 x 2N+1) y de menor tamaño que la imagen, la ecuación se simplifica a 1.5:

$$I \circ F(x, y) = \sum_{j=-N}^{N} \sum_{\mathrm{i}=-N}^{N} F(\mathrm{i}, j) I(x + \mathrm{i}, y + j) \tag{1.5}$$

Dependiendo de la aplicación, el tratamiento de los puntos bordes de la imagen y el paso

de avance del filtro difieren. Un ejemplo gráfico para se presenta en la figura 1.6.

Figura 1.6: Ejemplo de cálculo de una convolución

La operación de correlación puede considerarse una medida de similitud; el filtro recorre la imagen y se activa más en las zonas donde la imagen presenta el patrón del filtro. El uso de múltiples capas (arquitecturas profundas) permite que capas sucesivas aprovechen el procesamiento de las anteriores y se puedan detectar patrones cada vez más complejos (con un mayor contenido semántico) y que abarcan áreas mayores.

Volviendo a la analogía biológica, la operación de convolución se puede homologar al funcionamiento de múltiples neuronas con campos receptivos locales. Cada filtro de convolución representa el patrón de especialización de un tipo de neurona y cada desplazamiento del filtro, durante la convolución, representa la activación de una neurona de ese tipo procesando una región diferente de los datos de entrada.

1.3.3. Redes Recurrentes

Las redes MLP están bien adaptadas para trabajar con datos de tipo tabular, es decir, presentados como un vector de largo fijo de dimensión igual a la cantidad de variables. Cada entrada tiene asociado pesos que son entrenados siguiendo este orden que debe ser respetado para que la red interprete correctamente los datos nuevos.

Procesar datos secuenciales como series de tiempo o lenguaje natural presenta la dificultad de enfrentarse a entradas de largo variable y en donde, dependiendo del problema, el orden puede no importar. Las redes recurrentes enfrentan este problema utilizando una estrategia de parámetros compartidos y agregando una entrada que es función de un estado que depende de todas las entradas anteriores.

Como ejemplo, para una RNN de una capa, la unidad de la figura 1.7 se repite tantas veces como largo tenga la secuencia de entrada $x_1 \dots x_T$. En todas las etapas para los productos

matriciales se usan los mismos pesos. La operación de esta unidad queda resumida por la ecuación 1.6

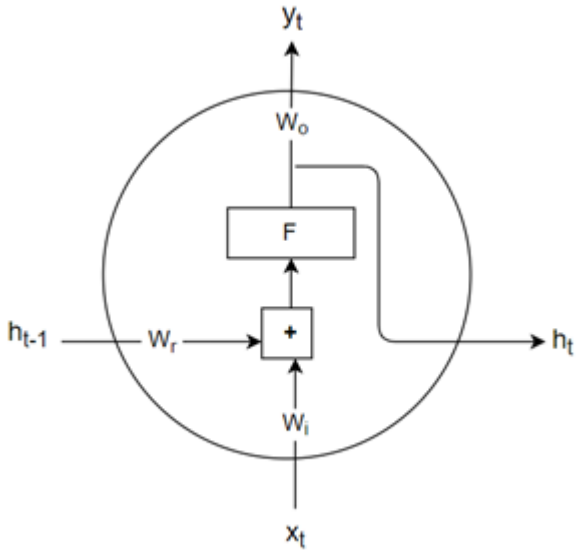

Figura 1.7: Esquema de una RNN de una capa para un paso de tiempo

$$h_t = \mathrm{F}\left(W_r h_{t-1} + W_{\mathrm{i}} x_t + b\right); \quad y_t = W_o h_t \tag{1.6}$$

La definición de h_t es explícitamente recursiva pues depende directamente de h_{t-1}. Por lo tanto, h_{t-1} representa a la memoria de la red y x_t es la nueva información a procesar. Estas dos entradas juntas pueden modelar comportamientos que requieren contexto de entradas pasadas. Y_t es la salida para el tiempo t, esta puede ser igual a h_t o puede pasar por otra capa.

1.3.3.1. Desvanecimiento del Gradiente

Para el entrenamiento de una RNN se aprovecha la propiedad de equivalencia entre una red MLP con pesos compartidos y una RNN extendida en el tiempo para poder usar el procedimiento estándar de descenso del gradiente del error. Sin embargo, las redes recurrentes, como la mostrada, tienen problemas prácticos para aprender relaciones de largo plazo entre las entradas: el llamado Desvanecimiento del Gradiente y su contraparte Explosión del Gradiente.

Según [24] este problema les impide aprender correctamente dependencias con más de 10 pasos de tiempo. Para entender este problema se procede a estudiar el gradiente respecto a los pesos de la entrada recurrente W_r:

Asumiendo que el error se calcula respecto a la salida del tiempo «t» dada una referencia «y» se tiene $E = E(y, y_t)$, con lo que el cálculo del gradiente para los pesos de la matriz de recursión se puede expresar según ecuación 1.7:

$$\frac{\partial E}{\partial W_R} = \sum_{\mathrm{i}=0}^{t} \frac{\partial E}{\partial y_t} \frac{\partial y_t}{\partial h_t} \frac{\partial h_t}{\partial h_\mathrm{i}} \frac{\partial h_\mathrm{i}}{\partial W_R} \tag{1.7}$$

De los factores de los términos de la sumatoria en ecuación 1.7, $\frac{\partial h_t}{\partial h_\mathrm{i}}$ tiene un comportamiento problemático; es inestable y puede tender a ser cada vez más pequeño o cada vez más grande al agregar pasos de tiempo. En la ecuación 1.8 se desarolla el término.

$$\frac{\partial h_t}{\partial h_\mathrm{i}} = \frac{\partial h_t}{\partial h_{t-1}} \frac{\partial h_{t-1}}{\partial h_{t-2}} \cdots \frac{\partial h_{\mathrm{i}+1}}{\partial h_\mathrm{i}} = \prod_{k=\mathrm{i}}^{t-1} \frac{\partial h_{k+1}}{\partial h_k} \tag{1.8}$$

Replicando el análisis de [8], pero simplificando para ilustrarlo, sea h_t y W_r escalares (RNN de una unidad). El término de la pitatoria en 1.8 queda expresado, a su vez, por otra pitatoria (1.9), en la que queda explicitado su componente problemática (1.10) :

$$\frac{\partial h_{k+1}}{\partial h_k} = W_r F' \left(W_r h_k + W_\mathrm{i} x_{t+1} + b \right) \tag{1.9}$$

$$\prod_{k=\mathrm{i}}^{t-1} \frac{\partial h_{k+1}}{\partial h_k} = W_r^{(t-\mathrm{i})} \prod_{k=\mathrm{i}}^{t-1} F' \left(W_r h_{k-1} + W_\mathrm{i} x_k + b \right) \tag{1.10}$$

Como la derivada de la función de activación es acotada, la expresión 1.10 queda dominada por el factor $W_r^{(t-\mathrm{i})}$. Si este factor es menor a 1, el gradiente se desvanece, de lo contrario crece exponencialmente.

El problema de la explosión del gradiente puede ser solucionado satisfactoriamente con métodos de regularización, en particular, normalizando Wr. Es el problema de desvanecimiento que demostró ser especialmente difícil de tratar sin recurrir a cambios de diseño.

1.3.4. Long-Short Term Memory (LSTM)

Con el objetivo de superar los problemas producidos por el desvanecimiento del gradiente fue ideada la red LSTM [12]. En esta se agregan funciones que permiten modelar explícitamente la selección de información relevante y la eliminación de información irrelevante. Para lograr esto la red cuenta con 2 estados de memoria que se modulan entre ellos y se introdujeron bloques o compuertas con tareas específicas (ver figura 1.8).

Para entender el funcionamiento de la LSTM, se estudiará el cálculo de un paso de tiempo. La información de la red hasta ese instante está contenida en los estados C_{t-1} y h_{t-1} que, siguiendo la notación, son llamados estado de la celda y estado escondido, respectivamente. Las dimensiones de los estados son iguales y quedan definidas por la cantidad de unidades. Como esquema general se tiene que h_{t-1} junto con x_t son usados para actualizar al estado de la celda y esta, por su parte, es usada para actualizar al estado escondido.

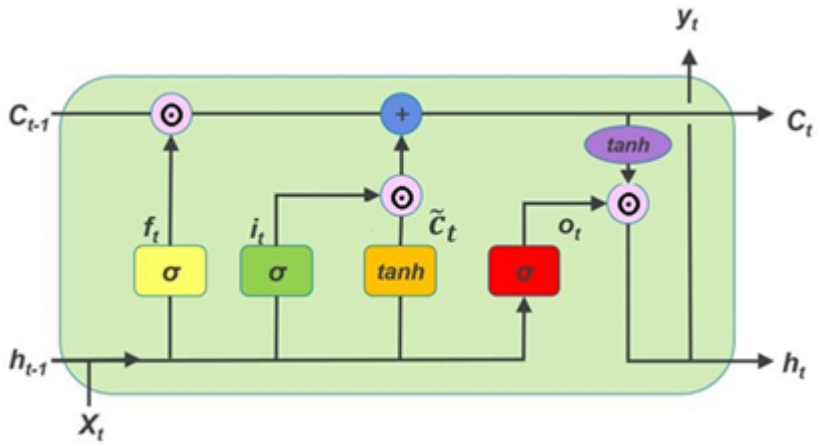

Figura 1.8: LSTM de una capa para un paso de tiempo

1.3.4.1. Aprendiendo a Olvidar

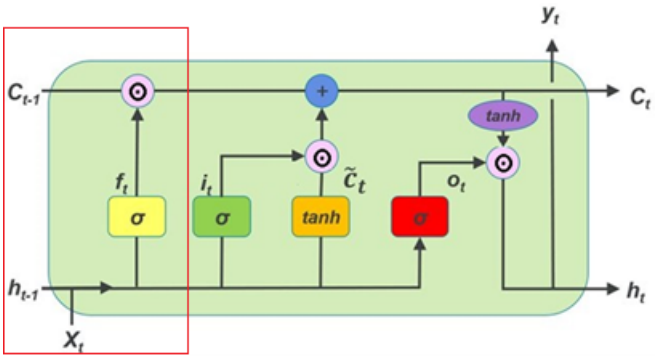

Figura 1.9: Celda LSTM, compuerta del olvido en recuadro rojo

La memoria de la red está modelada por los valores de los estados y el olvidar información, en este contexto, se equipara a llevar a cero a componentes de estos vectores. En este sentido, la función que se busca con la compuerta de olvido tiene que poder ponderar estas componentes de acuerdo con el nuevo contexto que indicará cuáles mantienen relevancia y cuáles no.

Para lograr esta dinámica, la compuerta de olvido procesa la entrada x_t junto con h_{t-1} para generar un vector $f_t \in (0,1)^{D_h}$, que luego es multiplicado punto a punto con C_{t-1} (ecuación 1.11)

$$f_t = \sigma\left(W_f\left[h_{t-1}; x_t\right] + b_f\right) \tag{1.11}$$

1.3.4.2. Seleccionando Nueva Información

Con el contexto h_{t-1} y la entrada x_t se genera un vector de estado $\widetilde{C}_t$ candidato que es inmediatamente ponderado por una compuerta similar a la de olvido, pero con el objetivo opuesto de ponderar según relevancia.

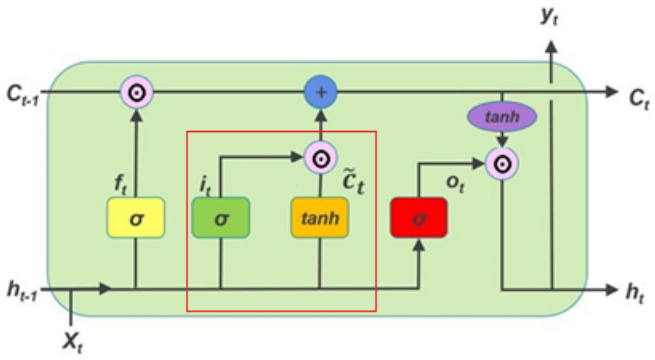

Figura 1.10: Celda LSTM, compuerta de entrada en recuadro rojo

$$\widetilde{C}_t = \tanh\left(W_c\left[h_{t-1}; x_t\right] + b_c\right) \tag{1.12}$$

$$\mathrm{i}_t = \sigma\left(W_{\mathrm{i}}\left[h_{t-1}; x_t\right] + b_{\mathrm{i}}\right) \tag{1.13}$$

La función de la compuerta de entrada i_t , que selecciona lo relevante, puede parecer redundante con la compuerta de olvido f_t , que trata de mantener solo lo relevante y, de hecho, en algunas aplicaciones de LSTM se usan versiones en que esto se explicita de la forma $f_t = 1 - \mathrm{i}_t$. Lo anterior puede servir para reducir la cantidad de parámetros a ser aprendidos, pero la versión más general descrita es la más usada y la que ha sido programada en la librería *Tensorflow Keras* utilizada en el presente trabajo.

1.3.4.3. Nuevo Estado de Celda

Con C_{t-1} ponderado por la compuerta de olvido se tiene lo que se quiere preservar del pasado y $\widetilde{C}_t$ representa la nueva información que se quiere codificar en el estado de celda. Para aprovechar ambos vectores se utiliza la operación de suma vectorial.

$$C_t = C_{t-1} \odot f_t + \widetilde{C}_t \odot \mathrm{i}_t \tag{1.14}$$

1.3.4.4. Compuerta de Saldia

Con C_t calculado se procede a calcular el nuevo estado oculto, h_t. El estado oculto es una transformación no lineal del estado de celda ponderado punto a punto por la salida de una capa sigmoidal con entradas h_{t-1} y x_t.

$$o_t = \sigma\left(W_o\left[h_{t-1}; x_t\right] + b_o\right) \tag{1.15}$$

$$h_t = \tanh\left(C_t\right) \odot o_t \tag{1.16}$$

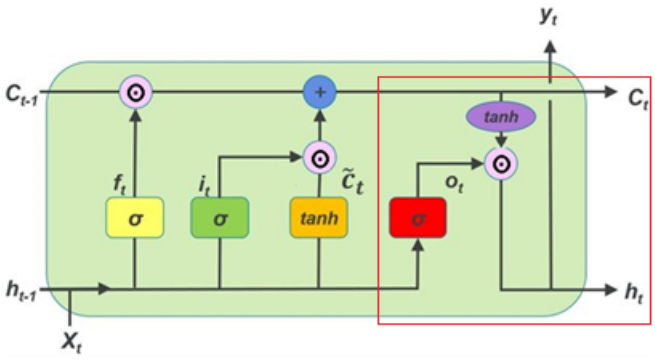

Figura 1.11: Celda LSTM, compuerta de salida en recuadro rojo

1.4. Métodos de localización Clásicos y Estado del Arte

La localización de eventos sísmicos puede ser abordada de diversas formas dependiendo de la disponibilidad y calidad de recursos (cantidad de sismógrafos utilizados, calidad de la señal, capacidad de detectar inicio de ondas P y S, compatibilidad de los registros entre estaciones, el modelo de velocidad utilizado y el rango de distancias en estudio entre otras variables) y del tipo de modelos con lo que se pretende trabajar.

Los métodos que se revisarán en esta sección se dividen en 3 grupos: Métodos Clásicos, Métodos de *Machine Learning* y Métodos de *Deep Learning*.

• Las estrategias clásicas incluyen desde métodos gráficos y directos a métodos automáticos que utilizan optimización clásica. Este tipo de estrategia requiere de un conocimiento explícito de un modelo físico del problema.

• En *Machine Learning* se incluirán los métodos clásicos de ML y redes neuronales en donde la entrada sea un vector de características seleccionado en base a conocimiento experto.

• Se considera abordar el problema con metodología DL o *end-to-end* a utilizar arquitecturas profundas de redes neuronales en donde las entradas son las señales mismas o representaciones en frecuencia de ellas.

Debido a que la cantidad de literatura específica para localización de eventos volcánicos VT es relativamente pequeña, la revisión de métodos abarca dos contextos: localización de sismos tectónicos (en donde se aprovecha la compatibilidad del análisis con los eventos tipo VT por contar con P detectable) y localización eventos volcánicos, en donde se incluyen localización de eventos correspondientes a otras clasificaciones (como temblores o eventos de periodo largo) pues los métodos aplicables a estos suelen ser aplicables a VT por explotar características presentes en ambos tipos (no así a la inversa).

1.4.1. Métodos Clásicos de Localización

1.4.1.1. Métodos Gráficos o Directos

Los métodos gráficos están actualmente obsoletos por contar con estrategias más robustas que lidian de mejor manera con las incertezas del modelo, aprovechando el poder de cálcu-

lo de las herramientas computacionales, pero una revisión de estos es útil para un primer acercamiento a la complejidad del problema y conocer la cantidad de variables (y por tanto estaciones) mínimas requeridas, aún en los modelos más simplificados.

El problema de localización de una fuente sísmica se puede considerar, en principio, un problema determinístico cuyo comportamiento obedece a las ecuaciones de onda derivadas de la mecánica clásica. La incerteza se atribuye a falta de conocimiento sobre el tipo de terreno que recorre la onda y a errores o distorsiones en los instrumentos de medición.

Método de Círculos

Asumiendo eventos con detecciones de comienzos de ondas P y S para al menos 3 estaciones (eventos tipo VT-A con buena calidad de señal, por ejemplo) se puede triangular una posición calculando la distancia del epicentro a cada estación como función de la diferencia en tiempos de llegada de las ondas ($\Delta T = T_s - T_p$). Por ejemplo, en el caso más simplificado, asumiendo velocidades v_p y v_s constantes, se tiene para una estación i con i $\in \{1, 2, 3\}$ su distancia se puede calcular siguiendo la ecuación 1.17:

$$\Delta T = \frac{\mathrm{d_i}}{v_s} - \frac{\mathrm{d_i}}{v_p} \tag{1.17}$$

$$\mathrm{d_i} = \Delta T \frac{v_p v_s}{v_p - v_s} \tag{1.18}$$

Con las distancias calculadas se conocen las ubicaciones posibles para cada estación y,la intersección de estos lugares geométricos da, por tanto, la ubicación del epicentro (ver figura 1.12). En la práctica, debido a la inexactitud de la detección de inicios de ondas y, más importante, a la diferencia respecto a un modelo de velocidades más realista, lo que se obtiene como intersección es un área de la que se debe decidir como escoger el punto candidato (centro de masa, por ejemplo).

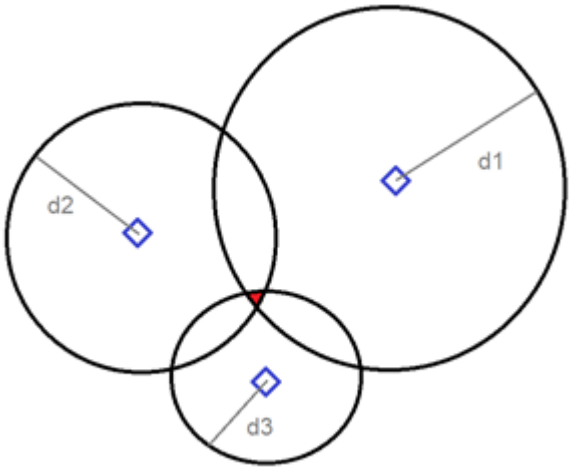

Figura 1.12: Representación de método de círculos

Método de Hipérbolas

En el caso de contar solo con detecciones de inicio de ondas P (eventos tipo VT-B, por ejemplo) se puede proceder de manera análoga, pero utilizando la diferencia de tiempos de llegadas P relativas entre estaciones. Con dichos arribos queda determinada la diferencia de las distancias de dos estaciones al epicentro y, con esto, queda definido un lugar geométrico de posibles epicentros formando una hipérbola cuyos focos son las dos estaciones. Un ejemplo se puede encontrar en [14], en donde lo usaron para localizar desprendimientos de glaciares por tratarse de un problema en donde la onda S está ausente.

Regresión lineal sobre Tp VS Ts-TP (diagrama de Wadati)

Para usar los métodos anteriores se requiere de un modelo de velocidad. El Diagrama de Wadati permite hacer una estimación de la relación entre v_p y v_s cuya proporción es requerida para resolver la ecuación 2.16. La aproximación más simplificada es asumir una razón constante, lo que se traduce en encontrar la regresión lineal de los pares T_p *vs* $T_s - T_p$.

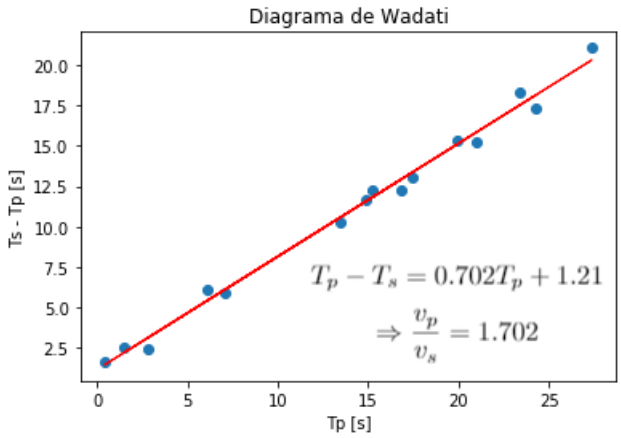

Figura 1.13: Ejemplo de Diagrama de Wadati

1.4.1.2. Métodos iterativos (mínimos cuadrados sobre linealización – HYPO71)

Los métodos iterativos se basan en la búsqueda del mejor ajuste a un modelo utilizando la linealización del problema inverso según el error cuadrático y el avance por gradiente. Uno de los más usados es el algoritmo HYPO71 [15] cuyo modelo se basa en [9]; una aplicación del método Gauss-Newton.

En particular, HYPO71 usa los tiempos de viaje registrados para las ondas P y S junto con un modelo de velocidad para calcular los errores o residuos entre el tiempo observado y calculado.

Definiendo a las coordenadas del hipocentro candidato como $r_o = (x_o, y_o, z_o)$, las coordenadas de una estación 'i' como $r_\mathrm{i} = (x_\mathrm{i}, y_\mathrm{i}, z_\mathrm{i})$ y al tiempo de origen candidato como t_o el modelo de velocidad se pude expresar de manera general utilizando la ecuación 1.19:

$$T_{\text{calc}} = T\left(t_o, x_o, y_o, z_o, x_{\text{i}}, y_{\text{i}}, z_{\text{i}}\right) = t_0 + \int_{r_o}^{r_{\text{i}}} \frac{1}{v(r)} \mathrm{d}r \tag{1.19}$$

En donde $v(r)$ es la velocidad de la onda como función de la posición $r = (x, y, z)$. Es decir, en esta función se explicita el modelo de velocidad del terreno en estudio.

Sean T_{obs} y $\widehat{X_k}$ los vectores de arribos observados y de variables estimadas en la iteración k, respectivamente. El paso δX para calcular la nueva estimación sigue la forma 1.20:

$$\frac{\partial T^{-1}}{\partial X}\left(T_{\text{obs}} - T_{\text{calc}}\right) = \delta X \tag{1.20}$$

Iniciar la iteración requiere de una primera estimación de ubicación y tiempo de origen no demasiado lejos de la solución. Una opción usada es considerar un punto cercano a la estación con las primeras detecciones de ondas, aunque no se puede asegurar que el método converja.

A pesar de continuar siendo de los modelos más usados y de mantenerse como referencia estándar, estos métodos presentan limitaciones que es importante conocer pues estas son heredadas por los métodos de ML y DL, cuyos rendimientos dependen crucialmente de la calidad de los datos y etiquetas.

En [3], [5] y [13] se mencionan problemas fundamentales y prácticos sobre este tipo de métodos. Por una parte, no siempre se pueden asumir como buena aproximación los supuestos necesarios para optimización por mínimos cuadrados y, más importante, la linealización del problema es una mala aproximación para casos altamente no lineales y en donde se tengan varios mínimos locales. Esto es especialmente relevante para el contexto de eventos volcánicos por lo mencionado en la sección 1.1.1, pues los modelos de velocidad para estos casos necesitan ser más complejos.

1.4.1.3. Método de Búsqueda en Rejilla

La búsqueda exhaustiva en un espacio discretizado es otro método que, al igual que los iterativos, utiliza un modelo explícito del problema y la solución es encontrada como el mínimo respecto a una función de error. A mayor capacidad computacional mayor es la capacidad de aumentar la resolución de la discretización y, por lo tanto, menor incerteza en la obtención del mínimo verdadero.

Aunque la utilización de ajuste a los tiempos de llegada, siguiendo un modelo de velocidad, es la forma mayormente usada, para casos en que, por las características de las ondas, sea difícil conseguir picados P y S, se pueden aprovechar otras funciones de ellas.

Por ejemplo, en [7] se estudia el uso de un modelo de decaimiento de amplitud para localizar eventos volcánicos tipo temblor y de periodo largo por carecer ellos de arribos de onda claros, y se plantea la búsqueda como encontrar el menor error definido por 1.22:

$$A(r) = A_0 \frac{\mathrm{e}^{-Br}}{r} \tag{1.21}$$

$$\mathrm{Err} = \sqrt{\frac{\sum \left(A_\mathrm{i} - A_\mathrm{i}^{obs}\right)^2}{\sum \left(A_\mathrm{i}^{obs}\right)^2}} \tag{1.22}$$

Como ventaja a los métodos iterativos se tiene que se evitan problemas en la solución del problema inverso linealizado (invertivilidad de la matriz, mínimos locales, etc). Para evitar el exceso de requerimiento computacional se pueden utilizar muestreo aleatorio o búsquedas mediante algoritmos evolutivos [3].

1.4.1.4. Correlación

Cuando no se tienen tiempos de llegada claros, sea por alto componente de ruido o porque la causa de la señal es un mecanismo emergente o continuo, como temblores en volcanes, una opción es usar métodos de correlación cruzada entre estaciones.

La correlación es una medida de similitud entre señales como función del desfase temporal de una respecto a la otra, para señales provenientes de dos estaciones «a» y «b» con un desfase «j» se define como:

$$C_{ab}(j) = \sum_{\mathrm{i}=1}^{N-j} a(\mathrm{i}) b(\mathrm{i}+j) \tag{1.23}$$

De tratarse de señales similares o 'correlacionadas' el valor de $C_{ab}(j)$ tendrá valores altos o bajos dependiendo del desfase aplicado. De tratarse de señales no correlacionadas (ruido, por ejemplo) el valor de la correlación es bajo para todo desfase. En el caso de una señal que es capturada por sensores separados por una distancia, la diferencia entre los tiempos de viajes será reconocida como el desfase que produce el primer peak positivo.

Con la diferencia en tiempos de llegada calculados, se puede estimar la localización, valiéndose de un modelo de velocidad, de igual manera que en los métodos que usan los tiempos de llegada.

En [18] usan doble correlación cruzada entre pares de estaciones para detectar los desfases de señales de temblores volcánicos. Con estos desfases triangulan usando método de elipses. El uso de doble correlación lo justifican por tener mayor redundancia, lo que permite menor sensibilidad al ruido. Reportan errores de «unos pocos km» para pruebas con datos simulados.

1.4.2. Métodos de Machine Leaerning

En la literatura existe una cantidad bastante mayor de trabajos en detección automática de las ondas P y S (que luego pueden ser procesadas por otro software para estimar localización) que en estimación de localización directamente. Aunque se puede considerar que los

primeros, efectivamente, realizan una localización automática, dividiendo el problema y luego integrando las soluciones en una rutina, el enfoque del presente proyecto es la estimación de localización de manera directa y el énfasis estará en estos.

Cómo único ejemplo de sistema de picado automático integrado a software de localización se expone a [10], pues es un sistema que utiliza redes neuronales para la detección y en su estudio de resultados hacen uso de HYPO71 para poder dar una estimación tanto del error en el picado como en el resultado de la localización, usando este picado. La base de datos consistió de señales de 3 componentes captadas por un promedio de 6,19 estaciones.

En dicho trabajo se aplican funciones de los estadísticos varianza, *skewness* y *kurtosis* (definidos en ecuaciones 1.24) como descriptoras de una señal. Se estudia a cada estación de manera independiente (4 descriptores para detección de onda P y 6 para detección de onda S).

$$\begin{aligned} &\text{Var } = \sigma^2 = \tfrac{1}{N+1}\textstyle\sum_{\mathrm{i}=1}^{N}(x-\bar{x})^2 \text{ ; Skew } = \left|\tfrac{1}{N+1}\textstyle\sum_{\mathrm{i}=1}^{N}\left(\tfrac{x_\mathrm{i}-\bar{x}}{\sigma}\right)^3\right| \\ &\text{Kurt } = \tfrac{1}{N+1}\textstyle\sum_{\mathrm{i}=1}^{N}\left(\tfrac{x_\mathrm{i}-\bar{x}}{\sigma}\right)^4 - 3;\ \text{Integ } = \text{ Skew } \cdot \text{ Kurt } \left|\tfrac{\mathrm{d}(\text{ Skew })}{\mathrm{d}t}\cdot\tfrac{\mathrm{d}(\text{ Kurt })}{\mathrm{d}t}\right| \end{aligned} \tag{1.24}$$

En un primer paso se descartan ventanas consideradas ruido por tener un bajo valor de kurtosis y, luego, se calculan los descriptores para cada ventana restante. Con esto se generan 8 series de tiempo normalizadas para cada estación (4 para canal vertical y 4 para canal horizontal) en las que cada muestra se corresponde con la muestra central de la ventana que la origina.

Teniendo estas series de descriptores, se considera como candidato a onda P a la primera muestra cuya varianza supere un umbral determinado empíricamente. La búsqueda de P se refina en la vecindad de ese primer candidato: a cada muestra de la vecindad se le identifica con un vector compuesto por los descriptores de esa muestra y sus vecinas en un radio de 10 muestras.

Este vector de características es procesado por una red neuronal perceptrón de una capa con una salida en el rango [0,1]. La muestra que dé la mayor salida es considerada como el nuevo candidato a inicio de onda P.

Un procedimiento similar se realiza para la detección de la onda S, pero se agregan dos descriptores y requirieron perceptrón de dos capas.

El sistema permite detectar el 89 % de las ondas P y el 67 % de las ondas S, lo que les permitió realizar estimaciones de localización en el 92 % de los casos. Los errores reportados para picado siguen una distribución de $0{,}00 \pm 0{,}07[s]$ para P y de $0{,}00 \pm 0{,}18[s]$ para S. Por su parte, procesando estos picados en HYPO71 se obtiene errores de $-0{,}18 \pm 0{,}77[km]$ para longitud, de $0{,}10 \pm 0{,}62[km]$ para latitud y de $0{,}1 \pm 2{,}0[km]$ para profundidad.

Los relativamente buenos resultados, considerando la utilización de perceptrón de 1 y 2 capas (arquitecturas poco profundas), sugiere que los descriptores seleccionados (ecuaciones 2.22) son muy atingentes al problema.

En [20] Se desarrolló un sistema pensado para detección y alarma temprana de sismos, utilizando solo una estación de 3 canales. Se estimó la distancia epicentral mediante regresión por soporte vectorial (SVMR), con un total de 25 descriptores; 12 relacionados a la estimación de magnitud, 9 basados en trabajos previos para la estimación de distancia al epicentro y 4 relacionados con la detección del ángulo azimutal. Los errores obtenidos fueron de 10.9 km en promedio.

1.4.3. Métodos usando Deep Learning

En [22] se estudia el uso de redes convolucionales temporales (1D) para el problema de detección y localización de eventos sísmicos (no volcánicos) usando una estación de 3 canales. La localización es tratada como un problema de clasificación al dividir el área de estudio en 6 *clusters* de aproximadamente 100 km2 cada uno. La cantidad y forma de los *clusters* fue seleccionada en base a *k-means* en el conjunto de entrenamiento. Alcanzan una correcta identificación de cluster para el 74.5 % de los casos.

En [17] se aprovechan las señales de múltiples sensores de 3 canales para la localización de eventos (no volcánicos). La base de datos se conforma de cerca de 3000 eventos captados de un enjambre de sismos originados en las cercanías de una falla geológica en West Bohemia. La distribución geográfica es muy concentrada, por lo que la zona de estudio es un área de unos 2 x 3 km2.

La entrada a la red son las señales concatenadas formando una matriz de N x T, siendo N la cantidad total de señales (3 por cada estación) y T la cantidad de muestras para el tamaño de ventana escogido. La arquitectura propuesta consiste en capas convolucionales 1D que procesan información en el tiempo seguidas por capas convolucionales 2D. Estas capas 2D tiene por objetivo combinar la información temporal de las distintas estaciones. Se aplica Dropout después de cada capa y, además, se aplica un «Dropout de estaciones». El Dropout de estaciones consiste en dejar en cero las entradas de estaciones al azar, en este caso con una probabilidad de 5 %. Reportan un error de menos de 200m para el 86 % de los casos.

Adicionalmente, al estudiar las respuestas de las capas a distintas entradas se dieron cuenta de una propiedad explotable para aprovechar de realizar una etapa de detección: Al ser entrenada solo con señales con presencia de eventos la primera capa termina especializándose en detectar patrones característicos de este tipo de señales y, por lo tanto, su nivel de activación es una medida de la similitud de nuevas entradas con estos patrones de entrenamiento.

La métrica utilizada para cuantificar esta discriminación es la energía de la primera capa (A) normalizada por la energía de la entrada (I), a la que llaman «energía de activación de la primera capa», definida por fórmula 1.25:

$$l = \frac{\sqrt{\sum A^2}}{\sqrt{\sum I^2}} \tag{1.25}$$

Si bien no fue necesario entrenar la red para esta tarea, se debe fijar un umbral para funcionar como detector. Con un umbral de 1,8 reportan una tasa de falsos positivos de 3,5 %.

En [25] también se crea un desarrollo en base a redes convolucionales, pero practicando una estrategia diferente para determinar la localización: En vez de tratarlo como un problema de regresión, se realiza una discretización del terreno de búsqueda (latitud, longitud y profundidad) y la red es entrenada para estimar la distribución de probabilidad de localización en dicho espacio, eligiendo la rejilla con mayor probabilidad como la candidata.

La entrada a la red son las señales de 30 estaciones de 3 canales para una ventana de 2048 muestras, organizadas de la forma 30 x 2048 x 3. Para obtener una estimación de la distribución se aprovecha como capa de salida a la última capa convolucional, la que se construye de manera que tenga dimensiones iguales al espacio de búsqueda.

La arquitectura se puede dividir en dos secciones en serie: la primera formada por capas convolucionales y capas *pooling* para reducir la dimensionalidad y buscar representar concisamente la información relevante y, la segunda, compuesta por capas convolucionales y capas *upsampling* para llegar a las dimensiones deseadas por la discretización.

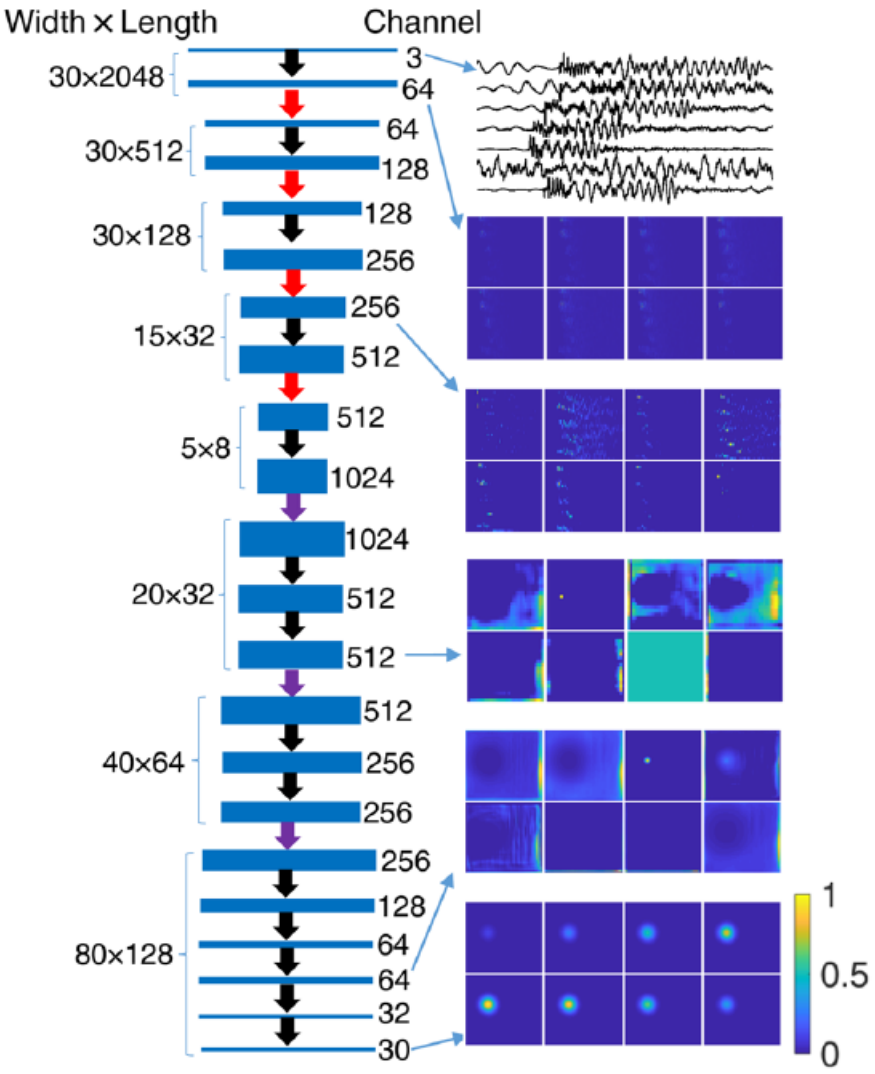

Figura 1.14: Arquitectura utilizada por [20]. Flechas negras indica convolución, Flechas rojas Maxpooling y Flechas negras Upsampling (permiso bajo licencia Creative Commons).

El terreno en estudio es de dimensiones 320 x 270 x 12 km3 y la discretización es de 80 x 128 x 30 (dimensiones de la última capa convolucional), lo que deja una resolución de 3.5km, 2 km y 0.4 km para latitud, longitud y profundidad, respectivamente. El error promedio reportado es de 4.9 km para el epicentro y de 1.0 km para la profundidad.

La forma de modelar la estimación de la localización, mediante una distribución de probabilidades, presenta algunas ventajas respecto a una regresión. Se pueden aplicar criterios para determinar si el *peak* de la distribución es fiable, por ejemplo, se pueden rechazar candidatos si su vecindad es considerablemente de menor probabilidad o exigir un valor mínimo para demostrar confidencia en la solución.

En particular, una característica que explotan en este paper, es determinar un umbral para la localización candidata, bajo el cual, se considera que no se tiene certeza suficiente. Esto, explican, sucede en al menos dos tipos de eventos: cuando las señales no presentan un patrón claro ya sea por excesivo ruido o por no contener un evento (con lo que se puede cumplir un rol de detector) y cuando los eventos se localizan por fuera de los límites de la discretización.

Otro enfoque probabilístico es presentado en [21], en donde utilizan redes bayesianas para la localización de eventos sísmicos tectónicos registrados en la base de datos global STEAD [19] mediante el procesamiento de señales de una estación de 3 canales (distintas estaciones son usadas para distintos eventos).

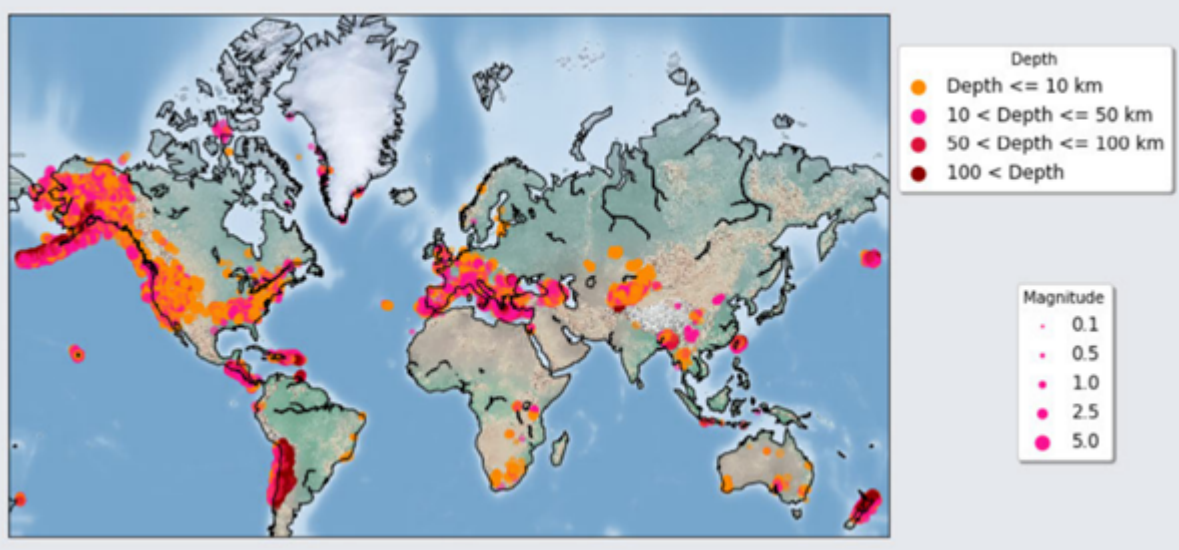

Figura 1.15: Distribución de los eventos de la base de datos STEAD [22] (permiso bajo licencia Creative Commons)

Las redes Bayesianas son redes neuronales que modelan explícitamente la naturaleza probabilística de las inferencias en modelos que aprenden de datos. Mientras las redes clásicas aprenden pesos con los que se computa la respuesta de la red como un 'mejor estimado', las redes bayesianas aprenden los parámetros de distribuciones para cada peso, típicamente media y desviación estándar al usar priors Gaussianos. Con estas distribuciones se puede calcular en cascada la incerteza de la respuesta final da la red.

Se sigue una estrategia de 3 pasos para la localización: estimación de distancia a la estación, estimación del ángulo azimutal respecto a la estación y cálculo de la localización, teniendo esos dos datos junto con la localización de la estación usada. Para las estimaciones de distancia y ángulo se entrenan dos redes bayesianas distintas. Esta metodología permite evitar la limitación de la aplicación del modelo a los límites de los datos de entrenamiento cuando se trabaja con localización absoluta y normalización de coordenadas y, por lo tanto, permite aprovechar una cantidad mucho mayor de eventos de entrenamiento y prueba.

La base de datos STEAD contiene 1,050,000 registros de sismógrafos de 3 canales asociados con unos 450,000 terremotos (ver figura 1.15), de los cuales fueron seleccionados 150,000 registros siguiendo como criterio que los epicentros estuvieran a menos de 110 km de la estación, que los registros tuvieran una relación señal a ruido de al menos 25 dB y en los que sus componentes estuvieran correctamente alineadas.

La entrada para la red destinada a la estimación de distancias es una ventana de 6 segundos (6000 muestras), filtrada entre 1 y 45 Hz, que contiene las ondas P y, para cada canal del sismógrafo, además, se le agrega un cuarto canal con información binaria en donde se etiqueta con 1 el periodo entre las llegadas P y S. Con esto las dimensiones de entrada son de 6000 x 4. Se menciona que este cuarto canal no es fundamental para lograr estimar la distancia pero que su inclusión es para convergencia más rápida en el entrenamiento.

La arquitectura seleccionada para esta red es una CNN temporal (1-D) de 11 capas con convolución seguidas por 2 capas densas. Las capas convolucionales tienen como función de activación a RELU y para la aplicación del filtro se utiliza la técnica de dilatar, en una magnitud creciente conforme avanzan las capas, para tener un campo receptivo mayor sin explotar la cantidad de parámetro.

Un filtro dilatado en magnitud 'd' a partir de un filtro F_1 1-D se define como:

$$F_{\mathrm{d}}(j) = \begin{cases} F_1(\mathrm{i}) & \text{si } j = \mathrm{i} + (\mathrm{d}-1)(\mathrm{i}-1) \\ 0 & \sim \end{cases} \tag{1.26}$$

La red estimadora del ángulo azimutal tiene dos entradas paralelas que se concatenan posterior a la capa *Flattening*, para luego ser procesadas en conjunto por dos capas densas. Una de las entradas es una ventana de 1.5 segundos (150 muestras) de los 3 canales, tal que los inicios de ondas P de las 3 estén contenidos (dimensiones 150 x 3). Esta entrada es procesada por una serie de 4 grupos de capas, en donde cada grupo está compuesto por una capa convolucional, una capa *Dropout* y una capa *Maxpooling*.

Paralelamente, se tiene como entrada una matriz de 7 x 3 compuesta por matriz de covarianza, los valores propios y vectores propios derivados de la señal de 3 componentes. El procesamiento es una sola capa convolucional. La salida son las estimaciones del coseno y el seno del ángulo azimutal. Esto último se ideó para poder entrenar con una función de pérdidas bien definida, pues el espacio de ángulos no permite la norma Euclidiana.

Los errores promedio obtenidos de esta manera fueron de 7.3 km para la estimación de epicentro y de 6.7 km para profundidad.

1.5. Picado Automático *PhasePicker*

Una opción para la localización automática es la integración de un detector de ondas a un algoritmo de localización, basado en estos tiempos de llegada. Debido a la importancia para la comparación de resultados del presente proyecto, se presenta al algoritmo PhasePicker [16], utilizado por OVDAS como apoyo a la detección manual.

PhasePicker modela a la señal de un canal como la respuesta de un oscilador de masa unitaria amortiguado de la forma:

$$m\ddot{u} + c\dot{u} + ku = -m\ddot{u}_{\mathrm{g}} \tag{1.27}$$

En donde u es el desplazamiento relativo y u_g el desplazamiento de tierra. Estos desplazamientos son los valores obtenidos de los sismógrafos.

Las métricas para la caracterización de la señal son las medidas de energía para este sistema (ecuación 1.28). Estas se obtienen de la integración respecto a u de la ecuación 1.27.

$$E_{\mathrm{K}} + E_{\zeta} + E_{\mathrm{S}} = E_{\mathrm{I}} \tag{1.28}$$

En donde E_I ,la energía total del sistema, es igual a la suma de E_K ,la energía cinética, E_ζ la energía de amortiguamiento (la energía disipada dada cierta trayectoria u) y E_S, la energía de estrés elástico. La derivada respecto al tiempo (la potencia) de E_ζ es la medida que el algoritmo monitorea para decidir la detección de la llegada de una onda.

Se reportan los errores respecto a la referencia manual al aplicar sobre 2 bases de datos, obteniendo 0,01 $\pm$ 0,41 $[s]$ y 0,01 $\pm$ 0,18 $[s]$. Estos resultados son comparados con los obtenidos con métodos de detección mediante ratio de promedios móviles de corto y largo plazo (STA/LTA) y con métodos autoregresivos, reportando mejores resultados con PhasePicker.

1.5.1. Implicancia para Error en Cálculo de Distancias

Como una manera de tener una primera aproximación del error en la estimación de distancias como efecto del error en el picado, se procederá a modelar el error en distancias calculadas según ecuación (2.16):

Los errores reportados por el paper son aproximadamente normales e, incorporando como supuesto que los errores de detección de P y S son independientes, se puede obtener la distribución de la diferencia de erros usando la siguiente propiedad de distribuciones normales:

$$\begin{array}{c} X_1 \sim N\left(\mu_1, \sigma_1^2\right); \quad X_2 \sim N\left(\mu_2, \sigma_2^2\right) \\ \Rightarrow \quad X_2 - X_1 \sim N\left(\mu_2 - \mu_1, \sigma_1^2 + \sigma_2^2\right) \end{array} \tag{1.29}$$

Con esto, la diferencia de errores sigue una distribución: $E_s - E_p \sim N(0, 0{,}3362)$

Considerando lo anterior, junto con la ecuación (2.16) y asumiendo valores estándar de velocidades de ondas ($v_p = 6{,}1[km/s]$ y $v_p = \sqrt{3}v_s \approx 1{,}7v_s$), el error de estimación de distancia desde una estación es una versión escalada de la diferencia de errores de picado:

Capítulo 2

Implementación

2.1. Base de Datos

Las señales sismográficas utilizadas para este estudio corresponden a registros de eventos VT obtenidos de una red de sismógrafos destinados a monitorear el Volcán Chillán (ver figura 2.1). Fue utilizado solo el canal de eje vertical y la tasa de muestreo es de 100 Hz. Su adquisición es responsabilidad del Observatorio Volcanológico Andes Sur (OVDAS).

Cada evento cuenta con la identificación visual por parte de expertos de los inicios de onda P y, de ser posible, S. Estas detecciones o «picados» de ondas es la información que, junto a un modelo de velocidad, es dada de entrada al software HYPO71 que entrega las localizaciones consideradas ground truth para este trabajo.

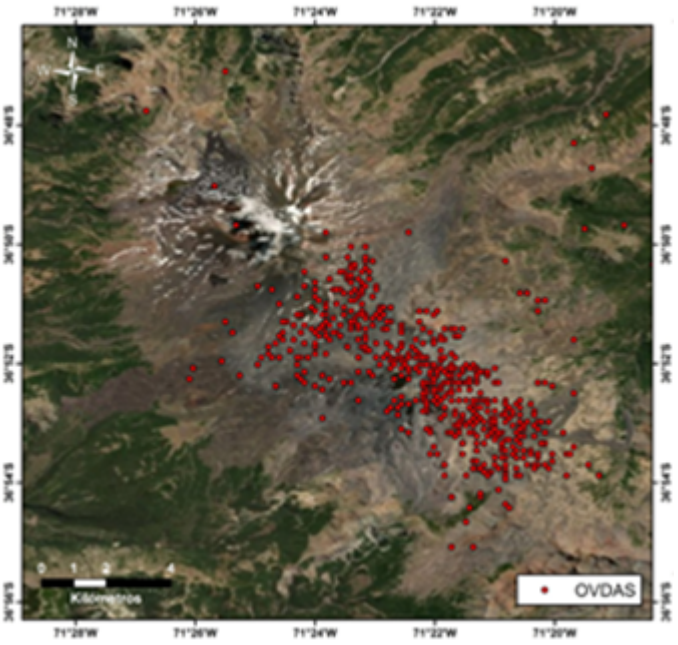

Figura 2.1: Distribución geográfica de eventos de base de datos Volcán Chillán

La base de datos (BD) se nutre de 3 periodos correspondientes a los años 2015, 2019 y 2020. La red se compone de 10 estaciones, pero no todos los eventos cuentan con señales en todas ellas y la BD del 2015 no cuenta con 2 sensores presentes en las del 2019-2020. Debido a esto y para poder compatibilizar las entradas a la red se optó por escoger un subconjunto de estaciones en las que la mayoría de eventos estuviera registrado y limitar la BD a eventos en que así fuera.

De esta forma las estaciones para el entrenamiento y prueba se redujeron a 3 y la cantidad de eventos que pudieron ser usados quedó en 1532 que se distribuyen en un área cuyos límites quedan definidos por un rectángulo de lados 24.43 km en latitud y 31.41 km en longitud. Cabe destacar que esta cantidad de estaciones es la mínima necesaria para triangular en el caso de trabajar con estaciones de un canal.

2.2. Metodología

Teniendo el ground truth como objetivo, se plantearon y entrenaron múltiples modelos de DL (LSTM y CNN) para la localización de eventos. Los resultados de estas arquitecturas se comparan con el ground truth usando como criterio el % de aciertos. Para efectos de este estudio, fue definido por parte del OVDAS como «acierto» a toda predicción con error menor a 1 km.

Una segunda etapa considera la comparación con un método semiautomático que consiste en la detección de los picados de onda mediante el software PhasePicker [16] para luego ser procesados por el software HYPO71. Las detecciones automáticas se realizan procesando solo las 3 estaciones usadas por las arquitecturas.

2.3. Arquitecturas y Optimización

Con el conocimiento cualitativo y cuantitativo a priori sobre el tipo de señal con el que se trabaja, se escogen arquitecturas que, en principio, sean capaces de lidiar con el nivel de complejidad adecuado y que tengan facilidad con el tipo de datos a procesar.

Dentro de las características de la señal se puede mencionar su no estacionariedad, su relativo bajo SNR por ser señales volcánicas y la alta dependencia entre estados para el cálculo de la localización del evento que la origina (los tiempos relativos de llegada entre ondas P y S y entre estaciones).

Este comportamiento temporal hace que una red LSTM sea considera una opción adecuada por ser capaz de aprender relaciones temporales de largo plazo y de olvidar información innecesaria, como el ruido. La elección de una LSTM dentro del universo de redes recurrentes se justifica por la cantidad de pasos de tiempo involucrados (mínimo 25, lo que descarta a las RNN simples por el problema del desvanecimiento del gradiente) y porque numerosos estudios han demostrado que estas redes siguen dando resultados similares o superiores a otras arquitecturas modernas [2].

Por su parte, las redes convolucionales han sido usadas con éxito en diversas aplicaciones de procesamiento de voz al aplicarlas sobre espectrogramas y, de manera más reciente, en

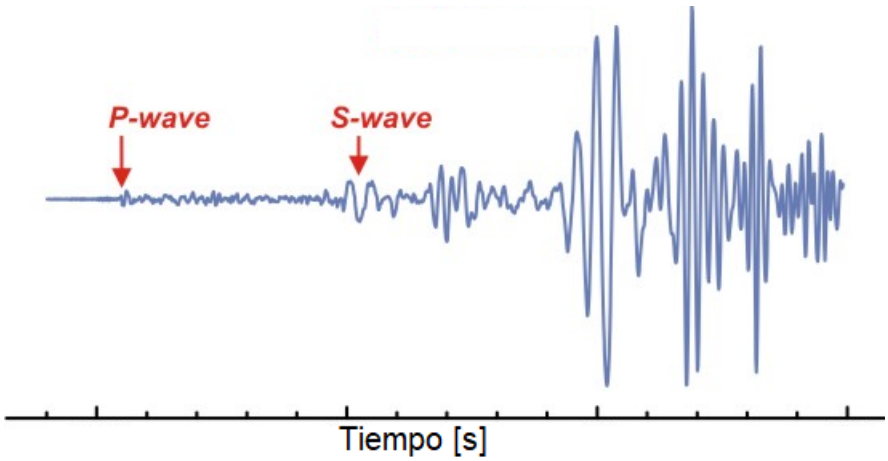

Figura 2.2: Ejemplo de señal sísmica explicitando su dependencia entre estados temporales

aplicaciones de detección, clasificación y localización de señales sísmicas y volcánicas utilizando convolución 1D o 2D sobre STFT, por lo que se decidió probar con arquitecturas CNN clásicas como fue propuesto con éxito para el caso de clasificación por [6].

Además, en [2] realizan un extenso estudio comparativo entre arquitecturas CNN y LSTM para problemas de modelado de secuencias y concluyen que las CNN logran resultados iguales o mejores mientras mantienen las ventajas prácticas de una CNN, dentro de las cuales se mencionan el paralelismo explotable por las GPU (entrenamientos más rápidos), campos receptivos variables y gradientes más estables (mayor estabilidad durante el entrenamiento), por lo que una comparación de estas dos arquitecturas es un ejercicio interesante y debatido en la literatura.

2.3.1. Hiperparámetros

Como en toda investigación en *Deep Learning*, parte ineludible del proceso de generación de modelos es la iteración exhaustiva sobre conjuntos de hiperparámetro, es decir, para cada tipo de arquitectura explorado se prueban configuraciones dentro de un universo de diseños plausibles. Tanto los mejores resultados como las diferencias en general dan pistas de qué tipo de arquitectura responde mejor a la naturaleza de las señales con las que se trata. Dentro de la búsqueda de hiperparámetros se incluyen tanto parámetros de diseño de la red como del preprocesamiento. En la tabla 2.1 se expone una lista de las variables utilizadas en la exploración.

Para criterio de OVDAS, la métrica de interés es el porcentaje de eventos con localización estimada con un error menor a 1 km, sin embargo, este tipo de métrica no es directamente aplicable para entrenamiento de una red pues no es diferenciable. Por ello se utiliza como *proxy* a la métrica Error Absoluto Promedio (MAE).

Para todo modelo se utilizó el optimizador ADAM y las funciones de activación fueron RELU para las redes convolucionales y tangente hiperbólica para las redes LSTM. Esto último debido a que la implementación optimizada GPU de la librería *TensorFlow Keras* tiene esta función fija.

Debido a la naturaleza probabilística de la inicialización y entrenamiento de los pesos de una red, cada configuración fue entrenada al menos 3 veces y sus % de aciertos promediados antes de comparar.

Preprocesamiento	**CNN**	**LSTM**
Largo de ventanas	Cantidad de capas CNN	Cantidad de capas LSTM
Traslape de ventanas	Filtros por capa CNN	Unidades por capa LSTM
Puntos FFT	Tamaño filtros	Cantidad de capas densas
Bandas de frecuencia usadas	Capas Pooling	Unidades por capa densa
Tipos de normalización	Cantidad de capas densas	Learning rate
	Unidades por capa densa	% Dropout
	% Dropout	
	Learning rate	

Tabla 2.1: Lista de Hiperparámetros

2.4. Preprocesamiento

Como fue anticipado por la sección de Marco Teórico, el preprocesamiento de la señal consiste en llevarla a una representación mixta entre temporal y frecuencial mediante la STFT. En la figura 2.3 se muestra como ejemplo las señales en el tiempo de un evento para las 3 estaciones empleadas y su representación en STFT para ventanas de 64 muestras (0.64 segundos) y traslape de 75 %.

Parte importante de la optimización del modelo tiene que ver con el *trade-off* entre la resolución temporal y la frecuencial, siendo ambas funciones del largo de ventana usado. La resolución temporal también puede ser mejorada manejando el traslape entre ventanas consecutivas, lo que, si bien no empeora la resolución frecuencial, tiene consecuencias en el largo de la secuencia y, por lo tanto, en la facilidad de procesamiento para la red.

Posterior al cálculo de la STFT se procede a la normalización. Esta es necesaria tanto por motivos de estabilidad y convergencia en el entrenamiento de redes como para hacer compatibles los registros de las estaciones utilizadas.

Primero se extrae el módulo del valor complejo (como está visualizado en figura 2.3) para trabajar con una función proporcional a la energía por banda y, esperando una mejor representación de las diferencias de energía, se aplica el logaritmo natural punto a punto.

La normalización consiste en llevar a media cero y desviación estándar = 1 (Z-score) y se efectúa por separado para cada banda de cada estación de cada evento: Considerando NFFT puntos para el largo de las ventanas y el logaritmo del módulo de la STFT de una estación queda definido como $S = \left[b_0; \ldots; \frac{b_{NFFT}}{2}\right]$ en donde $b_{\mathrm{i}} = [b_{\mathrm{i},0}, \ldots, b_{\mathrm{i},NVentanas}]$ es el vector de componentes de frecuencia de una banda 'i' a lo largo de las Nventanas. Luego, se tendrá que S normalizado corresponde a ecuación 2.1:

$$\bar{S} = \left[\overline{b_0}; \ldots; \overline{\frac{b_{NFFT}}{2}}\right] \quad ; \quad \bar{b}_l = \frac{b_{\mathrm{i}} - \mu_{\mathrm{i}}}{\sigma_{\mathrm{i}}} \tag{2.1}$$

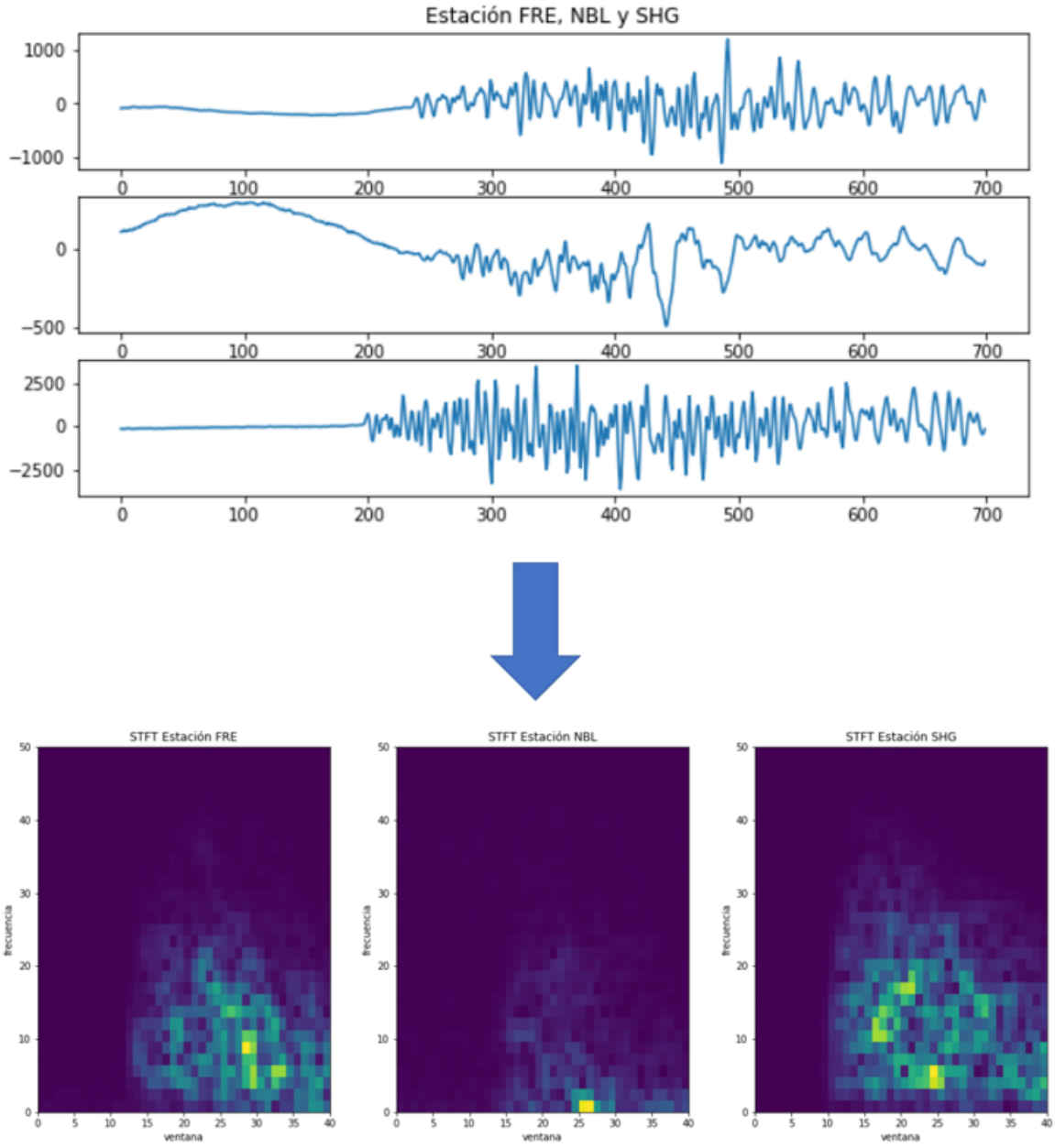

Figura 2.3: Esquema del preprocesamiento: Señales en el tiempo (Arriba) son transformadas a su representación en STFT (N=64, Traslape = 75 %)

Por último, del STFT normalizado se probaron distintas cantidades de bandas de alta y baja frecuencia eliminadas, para evitar estar agregando ruido al procesamiento (ver figura 2.4).

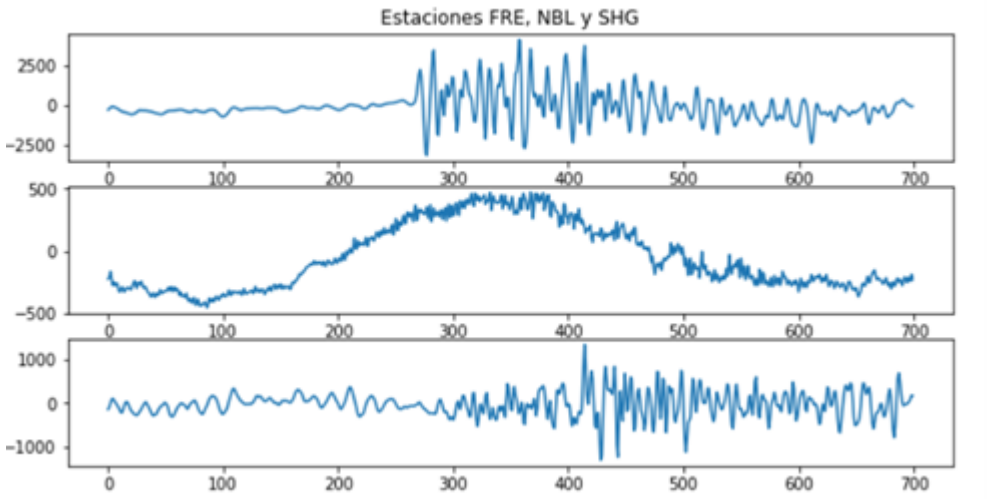

Figura 2.4: Ejemplo de señal con ruidos de alta y baja frecuencia

2.5. Arquitecturas CNN

Para las redes convolucionales las STFT normalizadas de las 3 estaciones forman una entrada tipo imagen de dimensiones descritas por la ecuación 2.2

$$(\text{ Alto } \times \text{ Ancho } \times \text{ Canales }) = \left(\frac{NFFT}{2} + 1 - \text{ nbines } \times \frac{T - W}{W * SP} + 1 \times 3 \right) \tag{2.2}$$

En donde *nbines* es la cantidad de bines que se excluyeron en el último paso del preprocesamiento. Un esquema general para las arquitecturas CNN usadas se ilustra en figura 2.5. El uso de puntos suspensivos representa la cantidad variable de capas.

Debido a que uno de los métodos de localización involucra el procesamiento de la diferencia entre tiempos de llegada de las ondas P y S, se agrega una restricción al espacio de búsqueda: el campo receptivo debe ser por lo menos lo suficientemente amplio en la dimensión temporal como para abarcar 1.5 segundos (cota superior para la diferencia de tiempo de llegada, $\Delta T = T_s - T_p$, para la base de datos).

Se entrenaron arquitecturas con distintas cantidades de capas convolucionales, capas de *pooling*, capas densas y porcentaje de *Dropout* para las capas densas. La exploración de tamaños de filtros fue limitada a tamaños 3x3 y 5x5, pues es lo adecuado para el tamaño de las imágenes (del orden de 30x40, variable dependiendo de los parámetros NFFT y porcentaje de traslape).

El tamaño relativamente pequeño de la imagen de entrada limita también a la cantidad de capas *pooling* que se pueden aplicar. Estas variaron desde cero a 3.

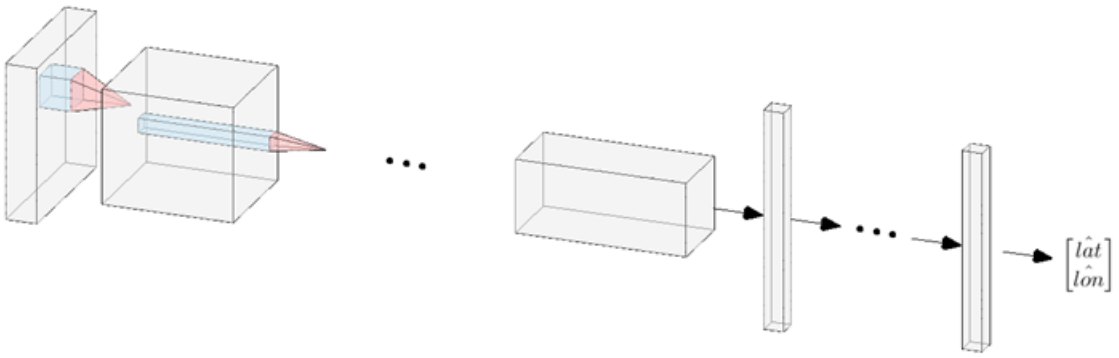

Figura 2.5: Tipo de arquitectura CNN usada. Puntos suspensivos indican cantidad variable de capas

2.6. Arquitecturas LSTM

Para las redes LSTM las 3 STFT normalizadas representan una serie de tiempo de largo $\frac{T-W}{W*SP} + 1$, en donde cada elemento es un vector compuesto de las bandas de frecuencia concatenadas de las 3 estaciones, es decir, de dimensión $3 \times (\frac{NFFT}{2} + 1 - nbines)$.

Aunque, como en toda red neuronal, una capa es, en teoría, suficiente para aproximar cualquier función, la utilidad de agregar más capas se explica en la teoría por la reducción progresiva en la representación de la entrada (lo que permite reducir la cantidad de parámetros requeridos en total) y, en este caso, se presume por la naturaleza de los métodos de clásicos en donde el cálculo es en 2 pasos: primero detectar las ondas y luego computar de acuerdo con algún modelo.

El esquema general de las arquitecturas usadas se ilustra en figura 2.6, en donde la cantidad de capas LSTM y de capas densas son variables.

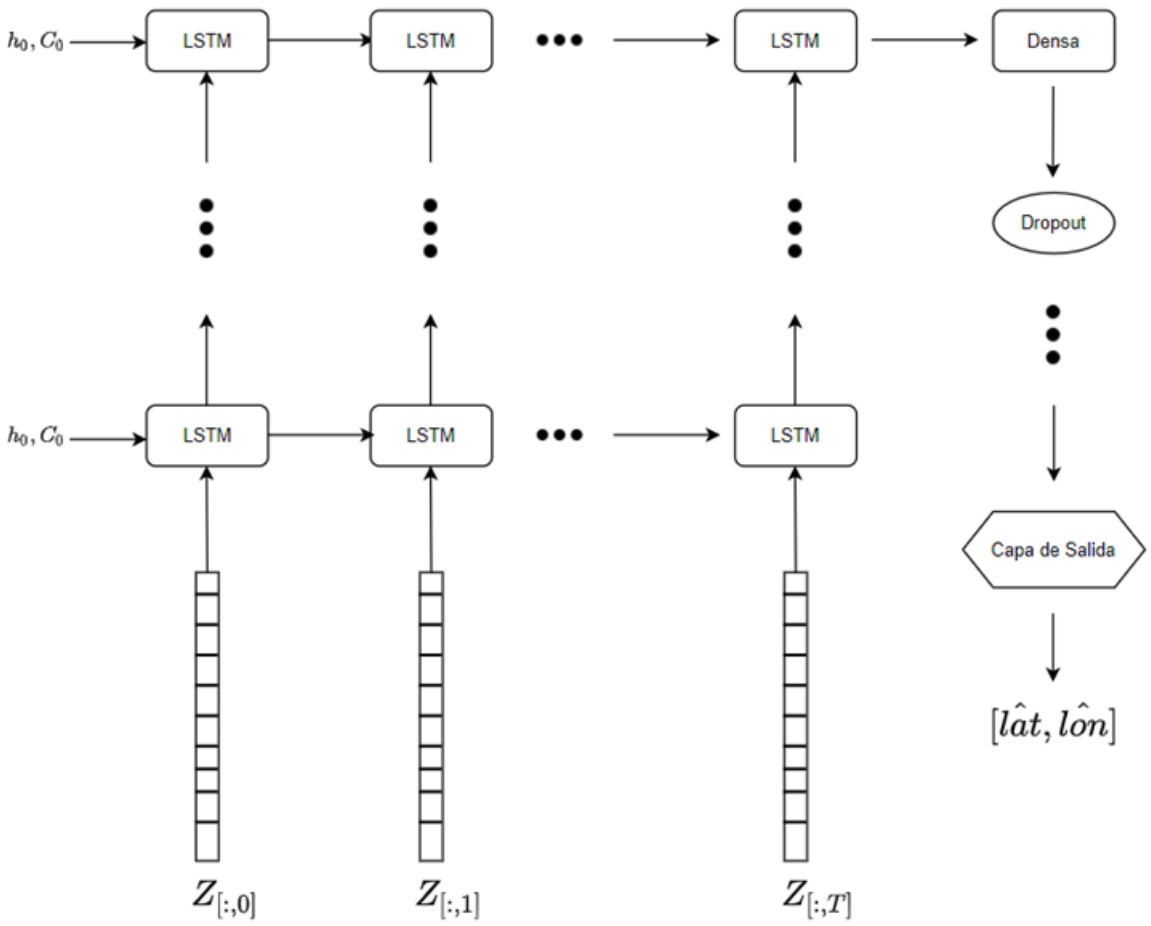

Figura 2.6: Tipo de arquitectura LSTM usada. Puntos suspensivos indican cantidad variable de capas

Capítulo 3

Resultados

Se partirá por listar los hiperparámetros de preprocesamiento y de arquitectura que lograron mejor resultado, luego se expondrán los resultados para 3 modelos: CNN, LSTM y detección automática + HYPO71. Este último corresponde a la aplicación del algoritmo *PhasePicker* para la detección de ondas y la utilización de estos picados como entrada para el algoritmo HYPO71, el mismo usado para las referencias con picado manual.

El criterio para determinar una predicción como acertada es tener un error respecto a la referencia menor a 1 km. En cada caso primero se visualizarán el error en km en latitud y longitud de cada evento de prueba (todos los aciertos están contenidos en el círculo unitario). Luego, se mostrará un histograma de frecuencias para distintos rangos de error, en donde las dos primeras columnas sumadas determinan el porcentaje de aciertos.

3.1. CNN

Del proceso de iteración sobre el espacio de hiperparámetros se extrae la arquitectura con mejor promedio de aciertos. Las tablas 3.1 y 3.2 resumen las hiperparámetros con mejor resultado para las redes CNN y, por su parte, en las figuras 3.1 y 3.2, se muestran el gráfico de dispersión de errores y el histograma de rangos de errores, respectivamente:

Parámetros de Preprocesamiento CNN	**Valor**
Largo de ventanas	64
Traslape de ventanas	50 %
Puntos FFT	64
Rango de frecuencia usadas	[1,5625 – 42,1875]
Normalización	Z score

Tabla 3.1: Parámetros de Preprocesamiento CNN

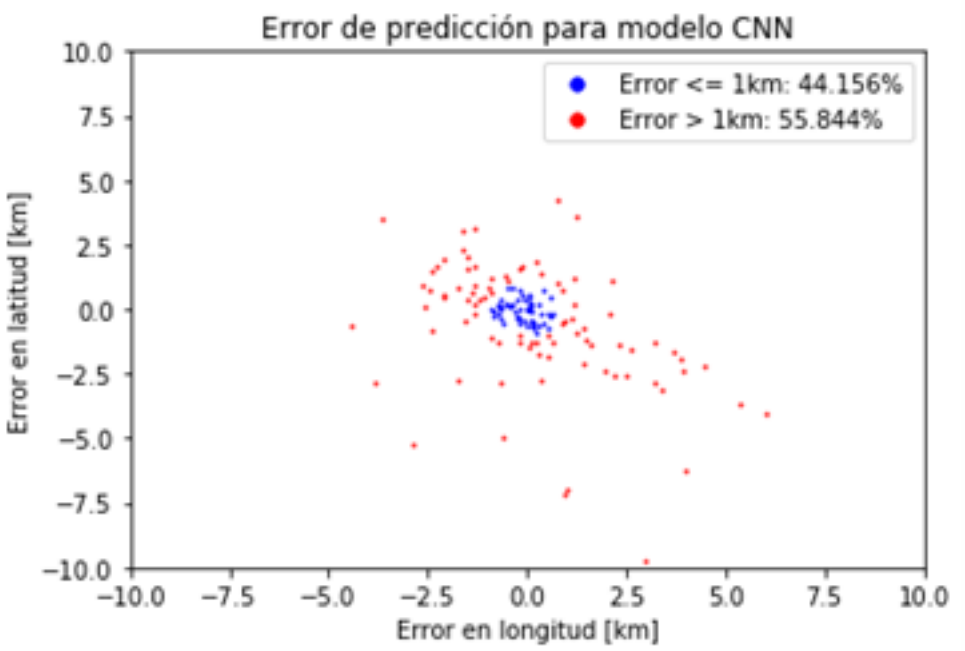

Figura 3.1: Error CNN

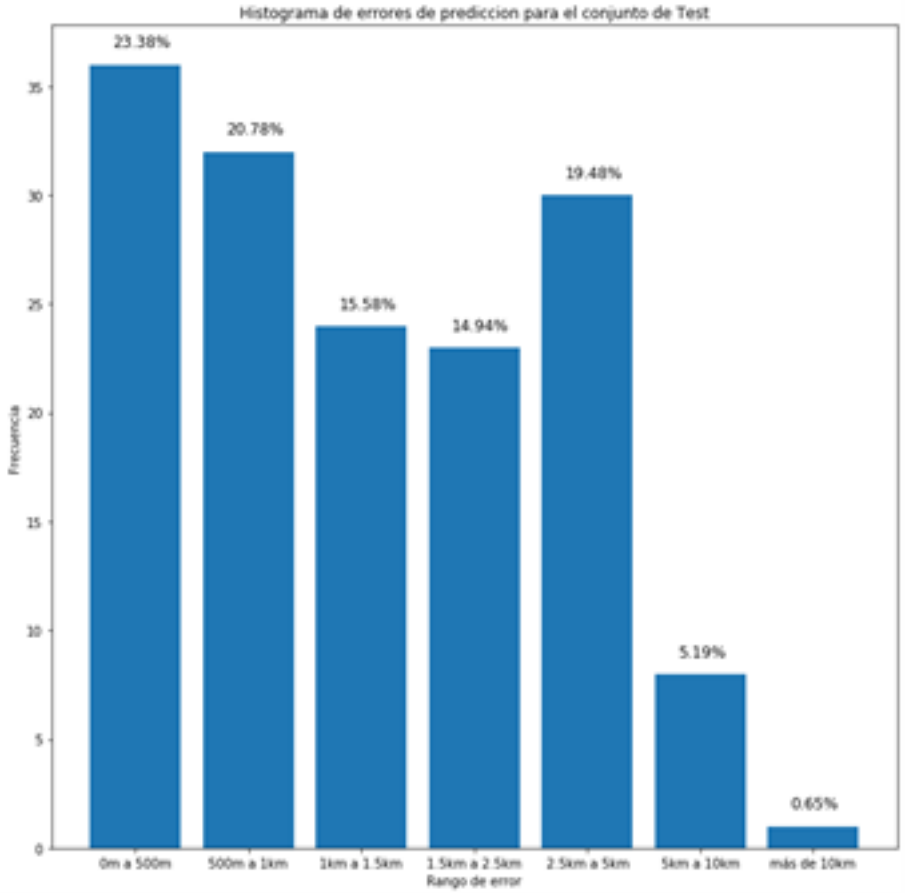

Figura 3.2: Frecuencia de errores por rango para CNN. Error promedio = 1.75 km, mediana = 1.18 km, percentil95 = 5.05 km

Hiperparámetros red CNN	Valor
N° Capas Convolucionales	4
Filtros por Capa	256
Tamaño Filtros	3 x 3
Capas Pooling	Sin Pooling
N° Capas Densas	2
Unidades por Capa Densa	64
Dropout	10 %
Learning Rate	0,001
Batch Size	32

Tabla 3.2: Hiperparámetros CNN

3.2. LSTM

Para el caso de la red LSTM, los hiperparámetros de preprocesamiento y de arquitectura están listados en las tablas 3.3 y 3.4, respectivamente. Los gráficos de dispersión de error y el histograma de rango de errores se muestran en las figuras 3.3 y 3.4.

Parámetros de Preprocesamiento LSTM	Valor
Largo de ventanas	50
Traslape de ventanas	60 %
Puntos FFT	64
Rango de frecuencia usadas	[3,125 – 50]
Normalización	Z score

Tabla 3.3: Parámetros de preprocesamiento LSTM

Hiperparámetros red LSTM	Valor
N° Capas	4
Unidades por Capa	256
N° Capas Densas	2
Unidades por Capa Densa	64
Dropout	10 %
Learning Rate	0,001
Batch Size	32

Tabla 3.4: Hiperparámetros LSTM

3.3. Picado Automático + HYPO71

En el caso de este método no hubo proceso de optimización de parámetros pues estos estuvieron determinados por las configuraciones establecidas por el equipo de la Universidad de la Frontera.

Los resultados mostrados corresponden a los eventos para los cuales HYPO71 logró converger dependiendo de la cantidad y calidad de las detecciones logradas por PhasePicker (ver

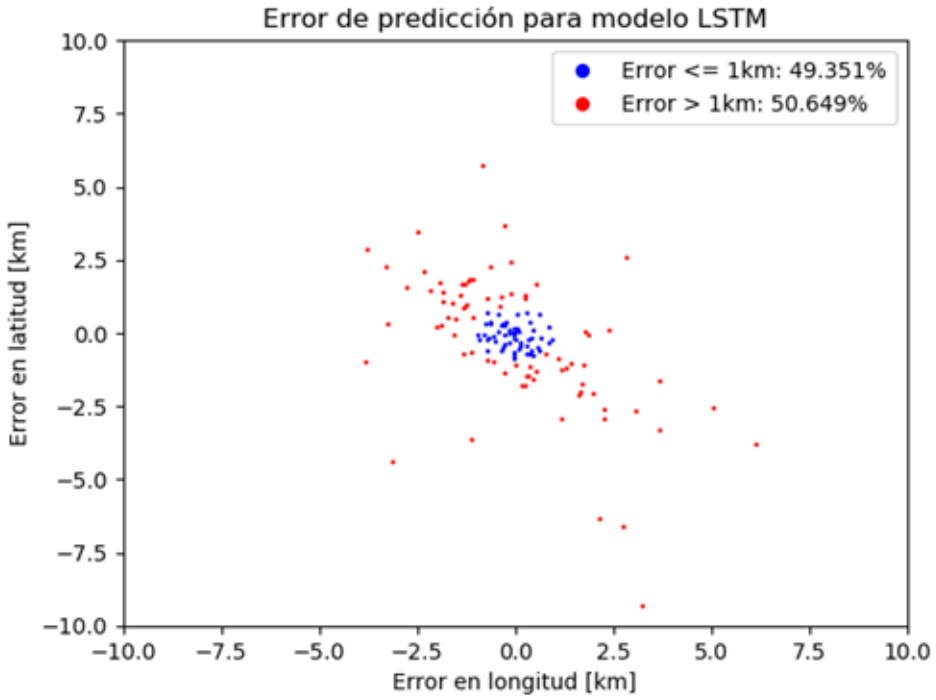

Figura 3.3: Error LSTM

Tabla Resumen			
	% de Aciertos	Mediana	Percentil 95
CNN	44,156	1,18 km	5,05 km
LSTM	49,351	1,02 km	4,16 km
PhasePicker + HYPO71	1,734	7,52 km	957,58 km

Tabla 3.5: Resumen resultados

figuras 4.5 y 4.6 para gráfico de dispersión y histograma de rango de errores, respectivamente).

3.4. Análisis de Resultados

3.4.1. Sobre el Porcentaje de Aciertos

Primero, se constata que LSTM alcanza mejores resultados que las arquitecturas CNN y que, a su vez, ambas obtienen resultados considerablemente mejores que el método de PhasePicker + HYPO71.

La diferencia entre LSTM y CNN es no despreciable y demuestra que, al menos en el conjunto de arquitecturas e hiperparámetros explorados, la modelización secuencial de la LSTM es más adecuada para el tratamiento de este problema. Esto podría considerarse lo esperado por ser las LSTM diseños especializados para problemas secuenciales, pero, como fue mencionado, la supremacía de las redes recurrentes por sobre las CNN para modelamiento de secuencias está cuestionada [2] y, además, la literatura de DL para localización de eventos sísmicos está dominada por aplicaciones de CNN temporales.

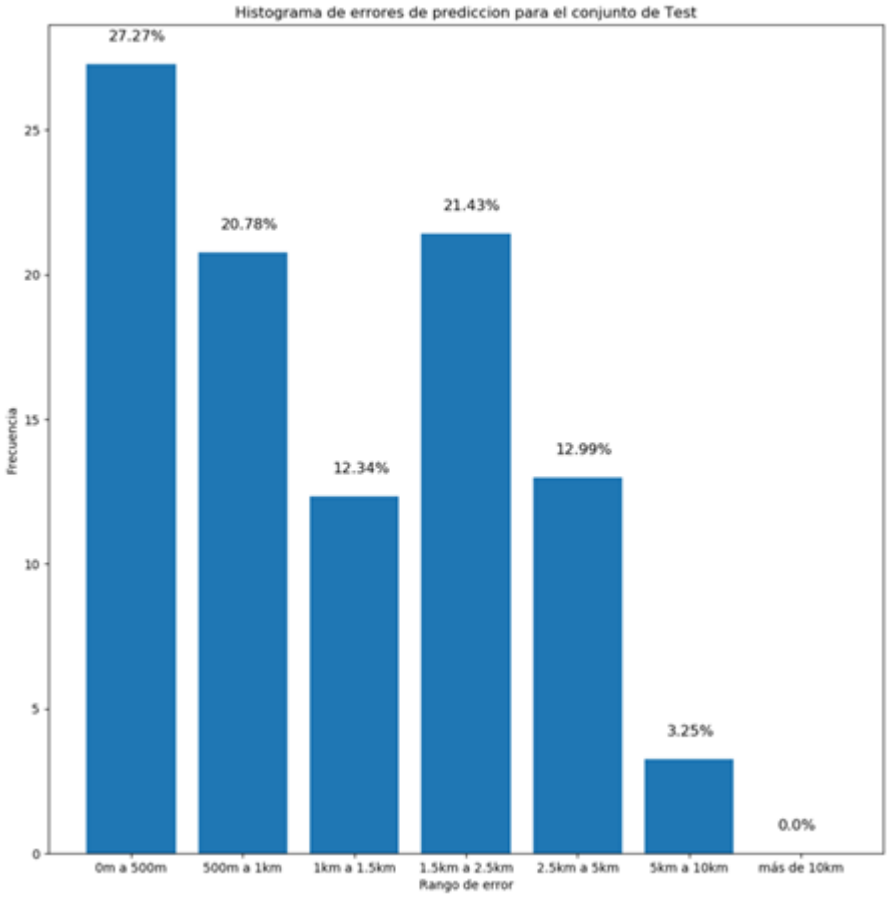

Figura 3.4: Frecuencia de errores por rango para LSTM. Error promedio 1.45 km, mediana 1.02 km, percentil 95 4.16 km

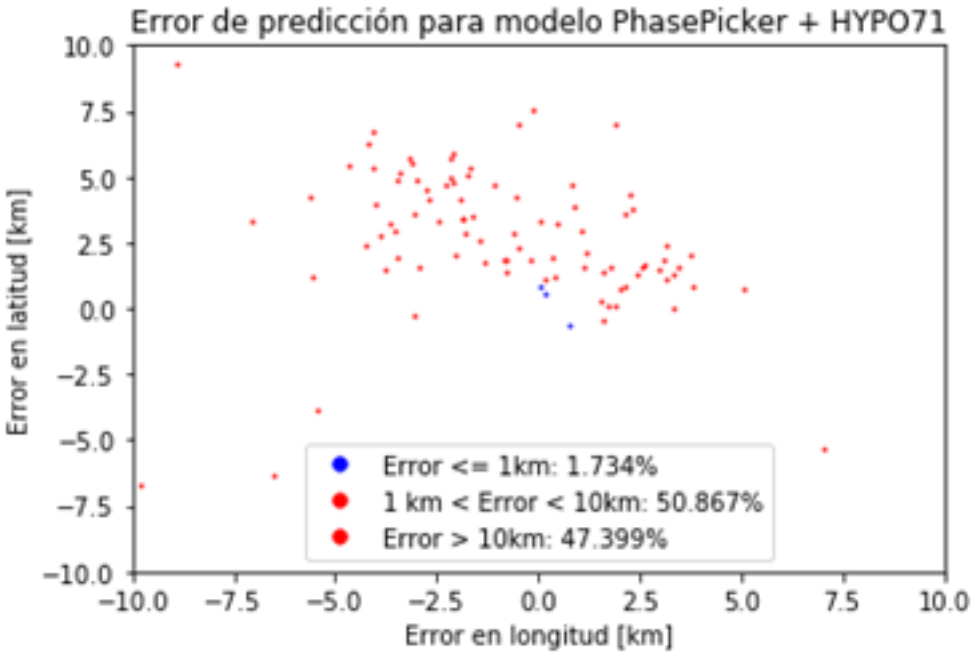

Figura 3.5: Para facilitar visualización se muestran en el gráfico solo los con error menor a 10 km

Por su parte, la gran diferencia respecto al método de picado automático es tal que se puede considerar que este último método fracasa en la tarea con el 1,73 % de acierto alcanzado. Es más, estos resultados son calculados considerando solo a los casos en donde el método HYPO71 logra converger, lo que ocurre para un 21,85 % de los eventos.

Los resultados de HYPO71 tienen 3 problemas:

- Porcentaje de aciertos muy bajo
- Eventos con errores extremos
- Eventos con localización indeterminada

El bajo porcentaje de aciertos es relativo a la métrica considerada para este contexto, en donde el área de estudio es relativamente pequeña, sin embargo, hay que tener en cuenta que HYPO71 tiene la capacidad de estimar distancias mucho mayores por lo que es esperable que el margen de error sea mayor en términos absolutos (más en detalle en siguiente subsección).

El problema de errores extremos es considerablemente grave pues, aunque la mediana está en el orden de magnitud del de las redes, la frecuencia de errores mayores a 10 km se mantiene no despreciable hasta llegar a alrededor de los 700 km (ver figura 3.7).

Sobre los eventos indeterminados: para que HYPO71 logre converger se deben cumplir algunas condiciones que, ante errores considerables del picado, no se cumplen. Como fue descrito en la sección 2.4.1.2, para que un método de optimización iterativa mediante linealización del problema inverso opere debe partir con un primer estimado no demasiado lejos del

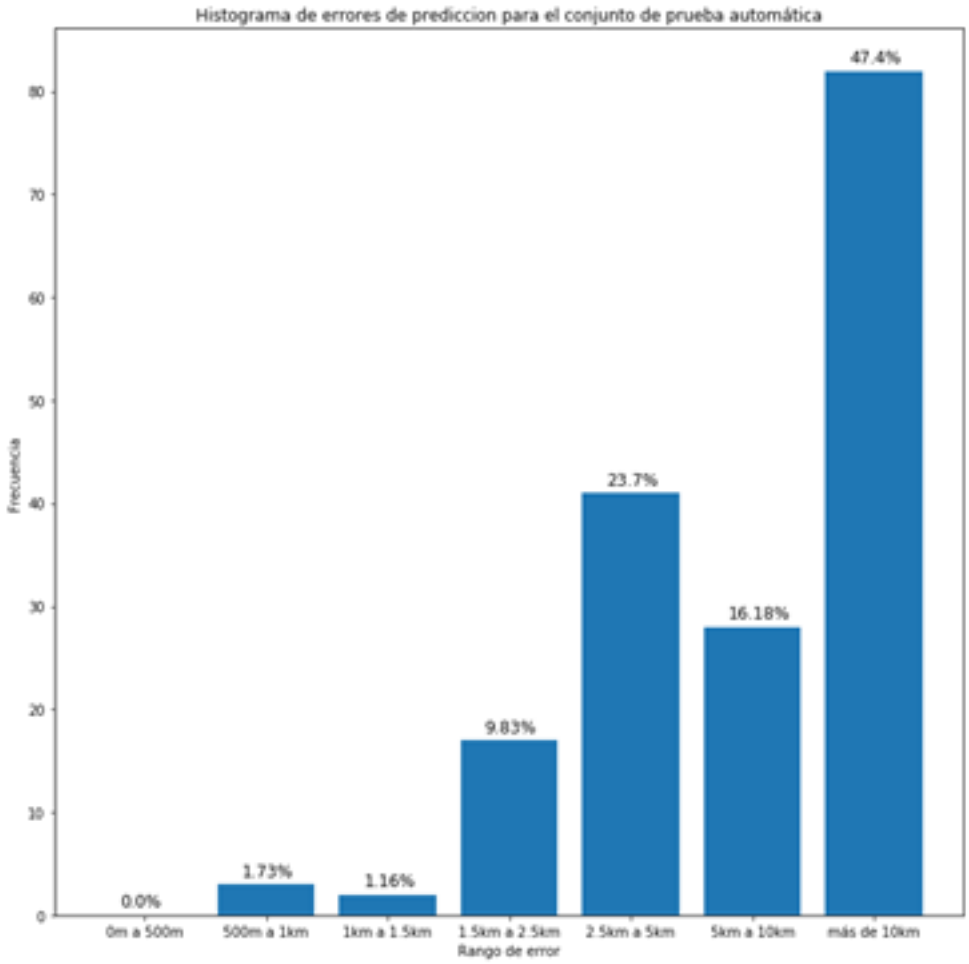

Figura 3.6: Frecuencia de errores por rango para *PhasePicker*+HYPO71

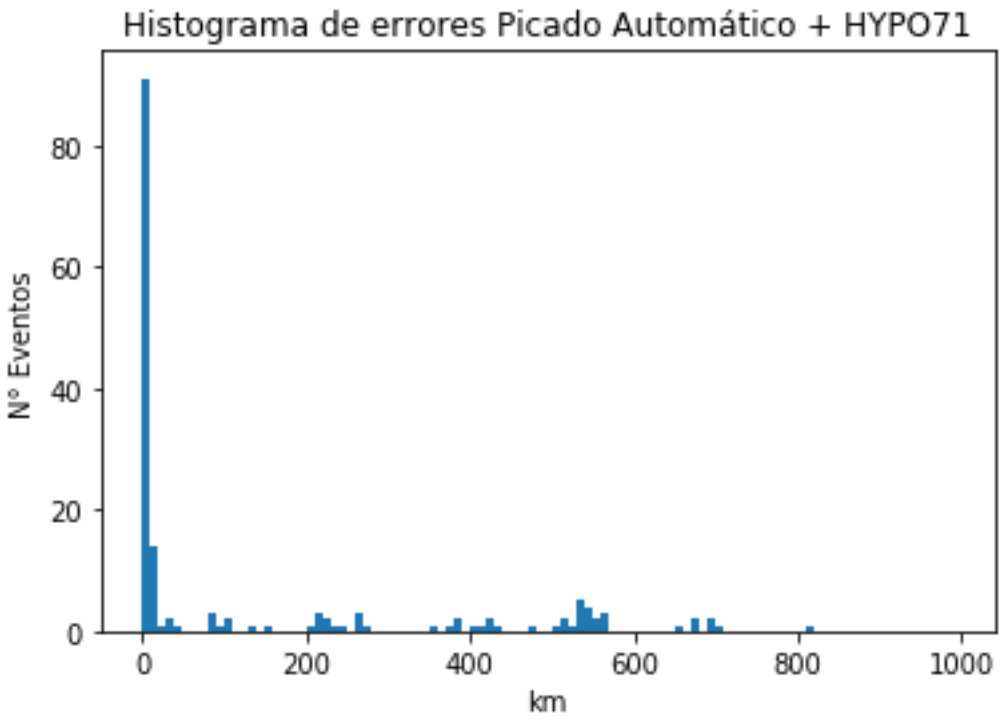

Figura 3.7: Histograma de errores para Picado Automático + HYPO71 con bines de 10 km

mínimo, este suele escogerse como un lugar cercano a la primera estación en captar la señal. Si la posición inicial no es adecuada el método puede quedar atrapado en mínimos locales con un error alto o no converger.

Por lo tanto, los errores de localización no solo no son directamente proporcionales a los errores de distancia estimada para cada estación por la naturaleza no lineal del problema, además, existen combinaciones de tiempos de llegada para los que el problema es indeterminable, lo que da cuenta de la sensibilidad de HYPO71 a errores en el picado. En particular, y por simplicidad para ilustrar, si consideramos el método de círculos, es claro que, dado un error suficientemente grande respecto a los radios verdaderos, es posible que los círculos no tengan intersección. Esto cobra mayor relevancia en la base de datos tratada en donde las distancias son pequeñas y el error, por tanto, más significativo.

Resumiendo, las respuestas del método automático se pueden dividir en 3 conjuntos: casos en donde no hay convergencia, casos en donde hay convergencia, pero el error es extremo, y casos en donde hay convergencia y el error es moderado (menor a 10 km). La alta proporción de eventos sin convergencia y de eventos con error extremo demuestran que el picado automático con PhasePicker fue poco adecuado para tratar con la dificultad de las formas de onda de la base de datos y refuerza la hipótesis de que los eventos de origen volcánicos son de más difícil tratamiento (esto al comparar con los resultados reportados por la referencia [16] y lo analizado en sección 1.5)

3.4.2. Sobre la capacidad de tratar outliers y de aplicar a otro contexto

A pesar de la clara superioridad de las redes sobre el picado automático, hay que tener en cuenta que esto se da en un contexto muy acotado; las redes están entrenadas con datos cuyas coordenadas están contenidas en un área definida. Aunque, dependiendo del conjunto de pesos, una red tiene la capacidad de entregar resultados que sobrepasen estos límites, durante el entrenamiento toda combinación de pesos que entregue predicciones fuera de ellos recibirá penalización que puede crecer indefinidamente, por lo que la tendencia es a que las predicciones se restrinjan al sector en donde el error está acotado.

En este sentido, la red tendría problemas en poder predecir nuevos eventos que se aparten de los confines de los datos de entrenamiento, mientras que HYPO71 permite trabajar con datos que pueden ser originados muy por fuera de los límites de entrenamiento (mientras se pueda considerar apropiado mantener el modelo de velocidad local).

De hecho, las predicciones de la red terminan no solo estando contenidas en el rectángulo definido por las localizaciones extremas, sino que, además, tienen como un primer estimado una distribución no homogénea en él, de lo que se infiere que la red aprende no solo de las formas de onda sino que también de las zonas más probables. Esto limita su aplicación a otros volcanes, para los cuales se deberá entrenar con sus propios datos históricos.

Para estudiar este efecto se experimentó generando señales de ruido Gaussiano y graficando la respuesta de la red. Por definición, el ruido Gaussiano no contiene información relevante para la determinación de una localización, por lo que la distribución de las localizaciones estimadas de esta forma se puede interpretar bayesianamente como el «prior» que la red utiliza para la predicción. Este prior es actualizado si se encuentra nueva información útil en las señales.

En la figura 3.8 se muestran las localizaciones de referencia del conjunto de entrenamiento (arriba izquierda), las localizaciones de referencia del conjunto de test (arriba derecha), las localizaciones estimadas cuando las entradas son ruido (abajo izquierda) y las estimaciones para el conjunto de test (abajo derecha). Primero se constata que los conjuntos de entrenamiento y de prueba tienen distribuciones similares.

Que las localizaciones estimadas para señales reales se concentran en el cluster de mayor probabilidad es lo esperado y no demuestra que la red esté dando más peso a las localizaciones históricas que a la forma de la señal; evidentemente, si la red cumple con su tarea de predecir correctamente la distribución del conjunto de prueba, que es la misma que la de entrenamiento, se replicará dicha distribución.

Pero, por otro lado, si se analiza la distribución de las localizaciones entregadas por la red cuando la entrada es ruido Gaussiano, se aprecia que se replica la distribución del cluster de mayor probabilidad del conjunto de entrenamiento. Esto indica que, efectivamente, la red tiene un prior o distribución por defecto que depende de las localizaciones de entrenamiento y que aplica cuando no tiene nueva información. De esto se desprende que hay un porcentaje de aciertos que se debe atribuir a el azar de que un punto estimado dentro del cluster esté a menos de 1 km de un evento de él.

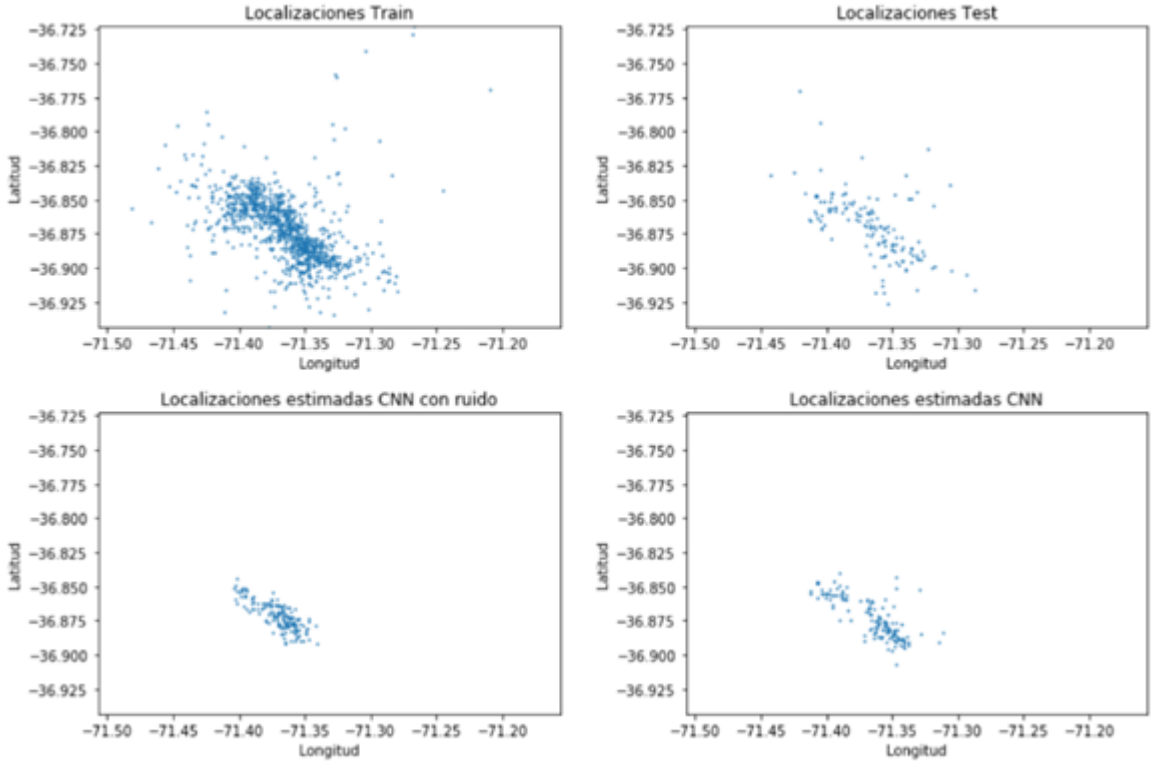

Figura 3.8: Localizaciones de referencia para train y test (arriba) y estimaciones para entrada con ruido y entradas reales (abajo)

Para tener una medida de la contribución del azar al porcentaje de acierto total se medirá el porcentaje de aciertos de la red CNN considerando las localizaciones del conjunto test con entradas generadas con ruido Gaussiano. Este resultado será comparado con 2 experimentos:

• Porcentaje de aciertos considerando una predicción fija igual al centro de masa de las localizaciones del conjunto de entrenamiento

• Porcentaje de distancias menores a 1 km entre los puntos del conjunto de test y puntos escogidos al azar en un rectángulo de lados iguales a los límites de la zona de estudio (24.43 km en latitud y 31.41 km en longitud).

En el caso de la red CNN para entrada Gaussiana estándar (media cero y desviación estándar unitaria), se obtiene un porcentaje de aciertos promedio de 9.09 %. En la figura 3.9 se ilustra una de las realizaciones (se escogió una con porcentaje de aciertos cercano a la media), donde se puede ver que persiste el patrón de errores siguiendo una distribución semejante a la geográfica.

Por su parte, el porcentaje de acierto fijando como predicción al centro de masa de las localizaciones del conjunto de entrenamiento es igual a 9,92 % y, por último, para el porcentaje de distancias menores a 1 km se realizó un experimento con datos aleatorios uniformemente distribuidos en un rectángulo de lados igual a los límites de la zona en estudio, dando como resultado en torno al 0.4 %

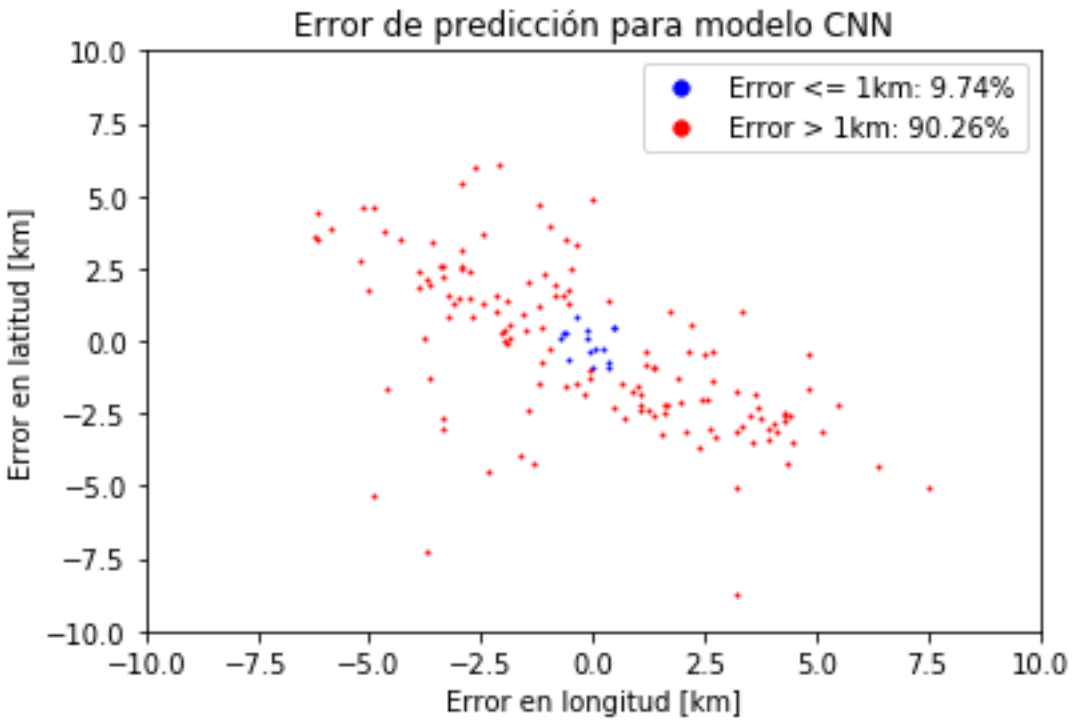

Figura 3.9: Error de predicción de modelo CNN para entrada ruido Gaussiano y localizaciones objetivo del conjunto Test

Si la red no tuviera memoria sobre la distribución geográfica y solo basara su predicción en las formas de las señales, se esperaría un % de aciertos, cuando las entradas son ruido, más cercano al experimento aleatorio, pero, al contrario, su porcentaje de aciertos es muy similar al de la predicción con centro de masa como punto fijo. Esto indica que, efectivamente, parte del ajuste de la red consistió en memorizar las zonas con mucha probabilidad de ocurrencia.

La mejora en el porcentaje de aciertos respecto a las entradas con ruido demuestra que la red es capaz actualizar la distribución cuando logra extraer información útil de los eventos evaluados. Esto se debe tanto a que la distribución actualizada permite predicciones fuera del cluster (visible en la figura 3.8) como a que las predicciones dentro del cluster son más certeras.

De lo anterior, es evidente que para intentar una aplicación a otro volcán, la red debe ser entrenada con datos que provengan de dicho volcán, mientras que PhasePicker+Hypo71 es de aplicación más general (aunque se debe ajustar el modelo de velocidad; un modelo genérico es capaz de entregar resultados).

Como corolario de la no aplicabilidad a otro volcán se debe mencionar que la capacidad de mantener el nivel de aciertos para eventos futuros depende de la representatividad de la base de datos respecto a la actividad del Volcán Chillán más allá del horizonte de los 3 años estudiados.

Capítulo 4

Discusión

4.1. Dificultades del Problema y de la Base de Datos

Como fue mencionado en el Marco Teórico, el procesamiento de señales de origen volcánico suele ser más desafiante que el de sismos de origen tectónico por tener mecanismos de origen más enrevesados y menor relación señal a ruido por la magnitud relativamente baja de los eventos volcánicos. Por esto último, la distancia a las estaciones debe ser más pequeña (en las cercanías del volcán), provocando que las llegadas de ondas estén más juntas haciendo más difícil el picado y más grave relativamente los errores en este.

A lo anterior, se le debe sumar que para el presente proyecto se contó con una base de datos fragmentada por no contar con señales en distintas estaciones para distintos eventos (lo que disminuyó las estaciones compartidas para entrenar y el número de eventos a usar) y a que la base de datos de OVDAS consistía únicamente del canal vertical. Lo anterior es importante pues, por la forma de propagación de las ondas P y S (transversal y longitudinal, respectivamente), estas son captadas con mayor facilidad por un eje de movimiento o por otro.

Las ondas P concentra su energía en el eje vertical (el usado en este proyecto) mientras que la onda S es visible mayormente en uno de los dos ejes horizontales. Como fue estudiado en la sección de métodos de localización, la capacidad de detectar ambos inicios de onda es uno de los elementos que permiten resolver el problema, por lo que contar solo con la componente vertical limita la información útil y dificulta el problema.

Otra limitación de información útil sucede por no contar con los modelos de los sismógrafos. Con ellos se podría separar el efecto del sensor para poder comparar las amplitudes relativas de las señales según las distintas distancias al foco y, como se vio en la sección de Métodos de Localización, esta información es en principio suficiente para un modelo de localización.

Respecto a la literatura, la cantidad de trabajos sobre eventos de origen volcánico es mucho menor que la de sismos, pero dado que este trabajo se limitó a eventos volcánicos tipo VT, en donde al menos las ondas P son detectables, se usaron referencias de ambos tipos. Esto hay que tenerlo en consideración al momento de comparar.

Debido a estas dificultades, las comparaciones con los métodos expuestos en la sección de Estado del Arte no son directas ya sea por la calidad de la base de datos, la cantidad de estaciones usadas, la cantidad de canales disponibles o por el origen de los eventos estudiados.

4.2. Ventajas del Método

Las redes neuronales no son un modelo explícito en el sentido de tener una relación funcional clara derivada de principios conocidos, pero sus mapeos internos mediante transformaciones no lineales y pesos entrenados se pueden considerar un modelo implícito del problema y, como fue visto en la sección de Métodos de Localización, hay gran diversidad de herramientas y principios por los cuales se puede llegar a la localización de la fuente (detección de ondas más modelo de velocidad, decaimiento de amplitud, correlación, uso de estadísticos, etc.).

Una de las pretensiones de las redes profundas es poder, data mediante, aprovechar las distintas piezas de información que usan estos modelos y poder combinarlas de manera óptima para los datos presentados, por lo que la diversidad en los tipos de solución del problema se puede considerar una característica explotable por DL que, además, le confiere mayor robustez por alimentarse, potencialmente, de más de una representación.

Por otro lado, los métodos que utilizan como entrada los tiempos de llegada de las ondas son altamente dependientes de la correcta detección de estos tiempos. Es por ello que el picado manual continúa siendo parte del trabajo de los geofísicos a pesar de contar con algoritmos de picado automático. El no necesitar tiempos de llegada se considera una ventaja, sobre todo en el contexto de eventos volcánicos en donde es más difícil la detección y los errores en estas tienen mayor peso relativo por las distancias implicadas.

Por último, Las redes neuronales en general cuentan con la ventaja de que su tiempo de ejecución es reducido.

4.3. Limitaciones del Método

La principal limitación del método es que su aplicación directa está restringida a los límites definidos por los puntos extremos de la base de datos. Esto tanto porque la red aprende de los patrones específicos al Volcán Chillán como producto de la normalización de coordenadas. De querer replicar el modelo para otras regiones se debe entrenar con datos de dichas regiones y, aunque los hiperparámetros obtenidos en este trabajo deben ser considerados como un buen punto de referencia y de partida para los experimentos, no se puede asegurar que estos mismos sean la mejor combinación para otros datos.

Una segunda limitación importante es que las entradas a la red son ventanas de 7 segundos en donde se sabe existe un evento. Si bien esto es una condición mucho menos exigente que requerir los tiempos de llegada de ondas, de todas formas, para una aplicación online, se requeriría de un detector de eventos. Ligado a lo anterior, como la red no se encarga de detectar un evento, al procesar una señal de ruido la red de igual manera entregaría una estimación espuria de una localización no existente, por lo que se debe tener la precaución de asegurarse de procesar un evento efectivo para la correcta interpretación.

Por último, como las redes fueron entrenadas con un ground truth obtenido mediante picado manual y procesamiento en HYPO71, los errores de estos procesos limitan la capacidad de aprender del fenómeno real. La función objetivo a minimizar encuentra su óptimo en lograr replicar HYPO71.

4.4. Trabajo Futuro

4.4.1. Sobre Cómo Superar las Limitaciones

Del análisis de los modelos estudiados en la sección 1.4, se puede proponer explorar el uso de algunas de sus arquitecturas para superar algunas de las limitaciones mencionadas en la sección 4.3.

• Para evitar que la aplicación del modelo quede confinada dentro de los límites de la zona de entrenamiento, una opción es entrenar redes con función objetivo como en [20] o [21], en donde se estima la distancia y ángulo a una estación. Al no buscar localización absoluta, sino que relativa a la estación utilizada, esto permite usar distintas estaciones y regiones.

Aunque la eficiencia del modelo puede verse mermada al exponerlo a regiones no exploradas durante el entrenamiento, al menos no es una imposibilidad metodológica el intentar su aplicación. Además, este mismo tipo de libertad de usar distintas regiones permite poder aprovechar grandes bases de datos como [19]. Lamentablemente, no se tiene conocimiento de una base de datos global semejante para el caso de eventos con origen volcánico.

• Para enfrentar el problema de que la red dará resultados espurios si se le entrega como entrada señales que no contienen eventos, se puede intentar una modelación que incluya la modelación de la incerteza de la respuesta como fue en [25] o [21]. En este sentido, un nivel alto de incerteza podría interpretarse como que probablemente no se está procesando una señal sísmica. Otra opción es explorar un desarrollo como en [17], aprovechando que las redes implícitamente aprenden la distribución de las entradas con presencia de evento y la poca activación general es un indicador de entrada anómala.

• Respecto a la base de datos, aunque se escapa del alcance de este trabajo, en [11] se mencionan los problemas con los métodos de localización mediante optimización del problema inverso mediante mínimos cuadrados (HYPO71) y utilizan algoritmos genéticos para encontrar localizaciones mejor ajustadas. En especial, para eventos de origen volcánico que requieren de modelos de velocidad de mayor complejidad (haciendo de la linealización una peor aproximación) es aconsejable la búsqueda de un mínimo global en base a búsquedas exhaustivas del problema directo, aprovechando el alto poder de cómputo en programas paralelizables [7].

4.4.2. Sobre la Cantidad de Estaciones

Una limitación muy fuerte fue la decisión de trabajar con las 3 estaciones que más se repetían (de 10 en total) como activas entre todos los eventos, esto con el fin de contar con una data sin entradas vacías. Esto redujo la información aprovechable en cada evento y la cantidad de eventos total, pues de igual forma existían eventos con alguna de estas 3

estaciones apagadas.

En este sentido, se considera que una estrategia potencialmente más fructífera y realista sería trabajar con todas las estaciones. Se espera que las redes debiesen poder ser capaces de ser lo suficientemente robustas como para manejar este tipo de perturbaciones.

Además, aún con la reducción de datos, se mantuvieron problemas con estaciones con señales saturadas. Estos eventos no fueron eliminados por no tener un criterio claro para ello; la saturación era a distintas amplitudes y en distintas ventanas de tiempo (un ejemplo es presentado en la figura 4.1).

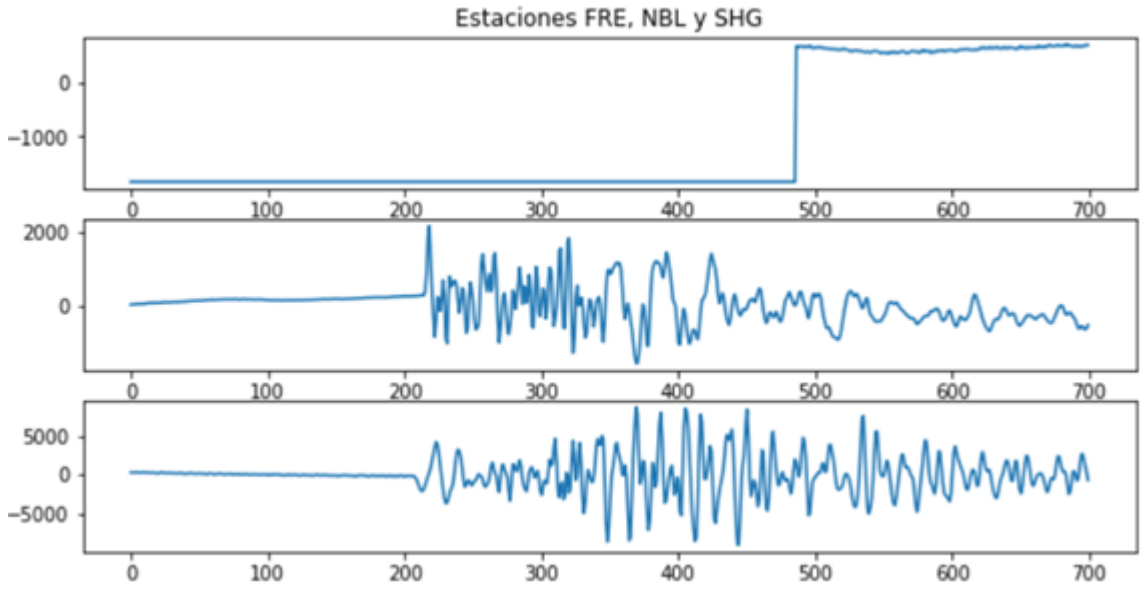

Figura 4.1: Ejemplo de evento con estación saturada

4.4.3. Data Aumentada

La cantidad de eventos totales con los que se pudo entrenar es relativamente baja y esto reduce la capacidad de generalización de cualquier red neuronal. Lamentablemente, la cantidad de recursos para el estudio de señales de origen volcánico es mucho más limitada que el de eventos sísmicos en general. Una estrategia que se suele usar es recurrir a algún tipo de incrementación artificial de la base de datos o «data aumentada». Estas estrategias pueden ser desde agregar perturbaciones como ruido aditivo, desfases, etc. a métodos más elaborados en donde se realiza una extrapolación a partir de las características estadísticas de las señales.

Para el caso de las señales sísmicas con detección de ondas P y S, una estrategia explorable es aprovecharse de que la diferencia de los tiempos de llegada ($\Delta T = T_s - T_p$) es proporcional a la distancia del foco a la estación. Con esta relación se pueden generar eventos nuevos a partir de eventos reales al considerar una nueva localización cercana a la original, siguiendo como estrategia modificar la señal de acuerdo con estas relaciones.

Conclusión

Los 2 tipos de arquitectura de Deep Learning estudiadas, CNN y LSTM, demuestran capacidad de resolver el problema de localización considerablemente mejor que la detección automática mediante integración de programa de picado automático (PhasePicker) y programa de localización en base a picados (HYPO71).

Los resultados de este último método lo hacen calificar como inadecuado para el procesamiento de señales complejas como lo son las de origen volcánico. Esto por la alta sensibilidad a la calidad del picado por parte del método de linealización iterativa del problema inverso y optimización en base a mínimos cuadrados.

Este resultado debe considerarse en el contexto de eventos originados en el Volcán Chillán y captados en un rango de 24,43 x 31,41 km2, una aplicación de estas arquitecturas para otras zonas requiere de un entrenamiento con datos propios de ellas. Por su parte, PhasePicker+HYPO71 puede ser adaptado más fácilmente a otras áreas, modificando el modelo de velocidades de ser necesario.

Respecto a la comparación entre las 2 clases de redes, las LSTM demostraron poder modelar de mejor manera que las CNN el problema de localización. Esto se podía esperar por ser las LSTM redes especialmente diseñadas para el procesamiento de información secuencial y dependencias en el tiempo, como lo es fuertemente este problema.

Bibliografía

[1] Keiiti Aki. *Volcanic Seismology.* IAVCEI Proceedings in Volcanology, Vol. 3. Springer-Verlag, Berlin, 1982.

[2] Kolter J. y Koltun V Bai, S. *An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling.*

[3] P. et al. Berger. *New locations of volcano-tectonic earthquakesunder Popocatépetl Volcano applying a Genetic Search Algorithm.* Geofísica Internacional (2011) 50-3: 319-340, 2011.

[4] Klinge K. y Wendt S. Bormann, P. *Data Analysis and Seismogram Interpretation, Capítulo 13 de New Manual of Seismological Observatory.* 2002.

[5] Lomnitz C. *Three Theorems of Earthquake Location.* Bull. Seis. Soc. Am., 96, 306 312.Doi:10.1785/0120050039., 2006.

[6] M. et al Curilem. *Using CNN to classify spectrograms of seismic events from Llaima volcano.* 2018.

[7] Cornejo-Surez et al. *Using Parallel Computing for Seismo-Volcanic Event Location Based on Seismic Amplitudes.* 2018.

[8] Pascanu et al. *On the Difficulty of Training Recurrent Neural Networks.* 2013.

[9] L. Geiger. *Probability method for the determination of earthquake epicenters from the arrival time only.* 1912.

[10] P. Gentili, S. y Bragado. *A neural-tree-based system for automatic location of earthquakes in Northeastern Italy.* Journal of Seismology, Springer Verlag, 2006, 10 (1), pp.73-89. 10.1007/s10950- 005-9001-z. hal-00647262., 2006.

[11] Bengio Y. y Courville A. Goodfellow, I. *Deep Learning.* MIT Press, 2016.

[12] J. Hochreiter, S. y Schmidhuber. *Long Short-Term Memory.* Neural Comput., vol. 9, no. 8, pp. 1735–1780, 1997.

[13] J. L. Husen, S. y Hardebeck. *Earthquake location accuracy, Community Online Resource for Statistical Seismicity Analysis.* 2010.

[14] M et al. Jeffrey Mei. *A Two Station Seismic Method to Localize Glacier Calving*. 2016.

[15] Lee W. H. K. and J. C. Lahr. *HYPO71 (REVISED) - A computer program for determining hypocenter, magnitude, and first motion pattern of local earthquakes*. U.S. Geol. Surv. Open-File Report 75-311, 100 pp, 1975.

[16] E. Kalkan. *An Automatic P-Phase Arrival-Time Picker*. Bulletin of the Seismological Society of America, Vol. 106, No. 3, pp., 2016.

[17] Petersen G. Vasyura-Bathke H. Kriegerowski, M. and M. Ohrnberger. *A Deep CNN for localization of clustered earthquakes based on multistation full waveforms*. Seismological Research Letters Volume 90, Number 2A March/April 2019, 2019.

[18] K. L. et al. Li. *A double-correlation tremor-location method*. Geophysical Journal International, 208(2), 1231–1236, doi:10.1093/gji/ggw453, 2017.

[19] Sheng Y. Zhu-W. y Beroza G. Mostafa, S. *STanford EArthquake Dataset (STEAD): A Global Data Set of Seismic Signals for AI*. 2019.

[20] Niño L.F. Ochoa, L.H. and C.A. Vargas. *Fast estimation of earthquake epicenter distance using a single seismological station with machine learning techniques*. DYNA, 85(204), pp. 161-168, March, 2018, 2018.

[21] Bayesian-Deep-Learning Estimation of Earthquake Location From Single-Station Observations. *S. Mostafa, S. y Beroza, G.* IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING, VOL. 58, NO. 11, NOVEMBER 2020 8211, 2020.

[22] Gharbi M. Perol, T. and M. Denolle. *Convolutional Neural Network for Earthquake Detection and Location*. Science Advances ISSN 2375 - 2548, 2018., 2018.

[23] J. Wassermann. *Volcano Seismology Capítulo 13 de New manual of seismological observatory practice (NMSOP)*. 2002.

[24] Staudemayer R y Rothstain E. *Understanding LSTM a tutorial into Long Short Term Memory Recurrent Neural Networks*. 2019.

[25] X. et al. Zhang. *Locating earthquakes with a network of seismic stations via a Deep learning method*. 2020.