Caracterización de eventos microsísmicos: procesamiento y algoritmos

Geof. Germán Ismael Brunini García

Director: Dr. Danilo R. Velis
Codirector: Dr. Juan I. Sabbione

Facultad de Ciencias Astronómicas y Geofísicas
Universidad Nacional de La Plata

Tesis presentada para optar por el grado de
Doctor en Geofísica

La Plata, Argentina
Agosto de 2023

Dedicatoria

A Cris, Milo e Indi, por su amor incondicional y apoyo constante.

Agradecimientos

Quiero agradecer especialmente a mis directores, Danilo y Juani, quienes me guiaron y ayudaron en cada uno de los momentos claves de este viaje. Sin ellos esta Tesis, sencillamente, no hubiese sido posible. Pero más importante, quiero agradecerles porque me extendieron la mano en ese momento en el que más la necesitaba.

Quiero agradecer a mis compañeros y amigos de Geofísica Aplicada, que hicieron de cada día una aventura llena de diversión y entretenimiento. Fueron el entretenimiento constante y necesario.

A la facultad, por darme el espacio y los recursos necesarios para completar esta tarea. También quiero agradecer al CONICET por su apoyo financiero, sin el cual este doctorado no hubiera comenzado.

Por último, quiero agradecer profundamente a mi familia, Cris, Milo e Indi, que son el motor y el apoyo más importante de todos.

Resumen

La microsísmica comprende al conjunto de técnicas y métodos orientados al procesamiento de las señales registradas durante un tratamiento de estimulación hidráulica de un reservorio de hidrocarburos. El objetivo principal de este procesamiento es el de extraer, de las trazas sísmicas registradas durante el monitoreo, la información necesaria para comprender las propiedades del yacimiento que está siendo estimulado. En este sentido, resulta imprescindible contar con técnicas modernas y eficientes destinadas a extraer la mayor cantidad de información posible de estas trazas. En esta Tesis se presentan un conjunto de estrategias diseñadas específicamente para el procesamiento de datos microsísmicos de pozo. Para comenzar, se introduce un método que permite comparar diferentes estrategias de filtrado de ruido obteniendo medidas cuantitativas sobre su desempeño. Para ello, se construyen histogramas de polarización a partir de las señales antes y después del filtrado, y se analizan sus diferencias. Los resultados encontrados indican que el método constituye una estrategia viable para cuantificar el impacto de diferentes algoritmos de filtrado. Luego, se presenta un método heurístico (*differential evolution*) para la determinación de las coordenadas espaciales de los eventos microsísmicos registrados durante una fractura hidráulica. El mismo está basado en un procedimiento no-determinístico, desarrollado específicamente para resolver problemas de tipo no-lineal. Además, se compara su desempeño con el que logran otros algoritmos de localización, tales como *particle swarm optimization* y *very fast simulated annealing*. La aplicación de dicho método resulta en localizaciones con precisiones satisfactorias. Por último, se muestra el diseño e implementación de una red neuronal en el problema de la inversión del tensor momento. Se encuentra que utilizando únicamente la posición del evento y las amplitudes relativas de la señal, la red es capaz de recuperar el tensor momento del mismo con altos niveles de confianza. Todas las herramientas utilizadas para obtener los resultados presentados en este documento fueron mayormente desarrolladas y/o implementadas por códigos en lenguaje Julia, Python y, en menor medida, Fortran. El objetivo final de los algoritmos presentados es, por un lado, brindar alternativas útiles para el procesamiento de las señales microsísmicas con aplicación al monitoreo de los procesos fracturación hidráulica, y por otro, contribuir en la evaluación de la calidad de los resultados de métodos ya existentes.

Índice general

Lista de acrónimos

AI Inteligencia artificial (AI, por sus siglas en inglés). Ver página p. 61
ANN Redes neuronales artificiales (ANN, por sus siglas en inglés). Ver página p. 129
BEP Barriles equivalentes de petróleo. Ver página p. 3
BPF Filtro pasabanda clásico (BPF, por sus siglas en inglés). Ver página p. 14
CNN Red neuronal convolucional (CNN, por sus siglas en inglés). Ver página p. 61
DE Evolución diferencial (DE, por sus siglas en inglés). Ver página p. 41
DNN Red neuronal profunda (DNN, por sus siglas en inglés). Ver página p. 59
EIA *U.S Energy Information Administration*. Ver página p. 4
EMD Descomposición en modos empíricos (EMD, por sus siglas en inglés). Ver página p. 23
EWA *Exponentially weighted averages*. Ver página p. 142
HLs Hidden layers. Ver página p. 134
IAPG Instituto Argentino del Petróleo y del Gas. Ver página p. 2
IEA *International Energy Agency*. Ver página p. 4
IL *Input layer*. Ver página p. 66
IMF Funciones de modo intrínseco (IMF, por sus siglas en inglés). Ver página p. 23
ITM Inversión de tensor momento. Ver página p. 7
LTU *Linear Threshold Unit*, o también TLUs. Ver página p. 66
MAM Método de Allen modificado. Ver página p. 54
ML *Machine learning* o aprendizaje automático. Ver página p. 61
MTI Inversión de tensor momento (MTI, por sus siglas en inglés). Ver página p. 7
OL *Output layer*. Ver página p. 66
PSO *Particle swarm optimization*. Ver página p. 41
QBtu Cuadrillones de Btu. Ver página p. 4
QF Factor de calidad (QF, por sus siglas en inglés). Ver página p. 16
RHRT Transformada hiperbólica de Radon de dominio restringido (RHRT, por sus siglas en inglés). Ver página p. 14
SA *Simulated annealing*. Ver página p. 43
SGD Descenso de gradiente estocástico (SGD, por sus siglas en inglés). Ver página p. 141

SRV Volumen de reservorio estimulado. Ver página p. 7

SVD Descomposición por valores singulares (SVD, por sus siglas en inglés). Ver página p. 14

TFH Tratamiento de fracturación hidráulica. Ver página p. 5

TM Tensor momento. Ver página p. 7

TWh Teravatio-hora. Ver página p. 4

VFSA *Very fast simulated annealing*. Ver página p. 41

VTI Isotropía transversal respecto del eje vertical (VTI, por sus siglas en inglés). Ver página p. 59

Capítulo 1

Introducción

1.1. Marco histórico y geológico

El origen de la industria hidrocarburífera en Argentina se remonta a los últimos años del siglo XIX, cuando la "Compañía Mendocina de Petróleo" puso en producción el primer campo petrolero en el área de Cacheuta, Mendoza, creando simultáneamente la primera tubería de petróleo, que conectaba la boca de pozo con la ciudad de Mendoza (Yrigoyen, 1993). A pesar de este antecedente, el nacimiento de la era petrolífera en el país suele considerarse a mediados de la primera década del siglo XX, cuando una perforación en busca de agua perforó un yacimiento de petróleo a una profundidad de aproximadamente 530 metros. El evento ocurrió en Comodoro Rivadavia en el año 1907 (Yrigoyen, 1993; Carbone et al., 2020b,a). Paralelamente, existían indicios de yacimientos en otras regiones del país (Stratta, 2013). Así, en los años venideros, la industria se expandió y comenzaron las exploraciones y explotación del recurso en otras zonas, como por ejemplo, la cuenca de Neuquén. Con el correr de los años, la industria continuó con su crecimiento a lo largo y ancho del territorio nacional, logrando producir hidrocarburo de forma exitosa en numerosas cuencas del país. Actualmente, existen seis principales cuencas productivas que comprenden una superficie total aproximada de 550000 km^2 (Barredo and Stinco, 2013). Las mismas reciben los nombres de: cuencas Paleozoica y Cretásica, ambas situadas en el extremo norte del país, cuenca Cuyana, ubicada mayormente entre las provincias de San Juan y Mendoza, cuenca Neuquina, principalmente en la zona de la provincia de Neuquén y alrededores, cuenca del Golfo de San Jorge, en las provincias patagónicas de Chubut y Santa Cruz, y finalmente la cuenca Austral, en el extremo sur del país. Según informes de la Secretaría de Energía de la Nación, la totalidad de las cuencas hidrocarburíferas produjeron un total de 29.3 Mm^3 (Mm^3: millones de metros cúbicos) de petróleo y 47.02 Mm^3 de gas durante el año 2018. Del total de la producción de petróleo, el 89 % se adjudica a las dos principales cuencas del país. Específicamente, un estimado de 13.5 Mm^3 corresponde a

la cuenca del Golfo de San Jorge, representando el 46 % de dicha producción, seguido de la cuenca Neuquina, de la cual se extrajo un volumen cercano a 12.57 Mm^3, lo que representa un 43 %.

En cuanto al gas, la cuenca Neuquina es la principal productora de este recurso, produciendo un total estimado de 28.4 Mm^3, por lo que es responsable de un 60.4 % del valor total de gas extraído en el país. Su ubicación geográfica puede apreciarse en la Figura 1.1 (izquierda).

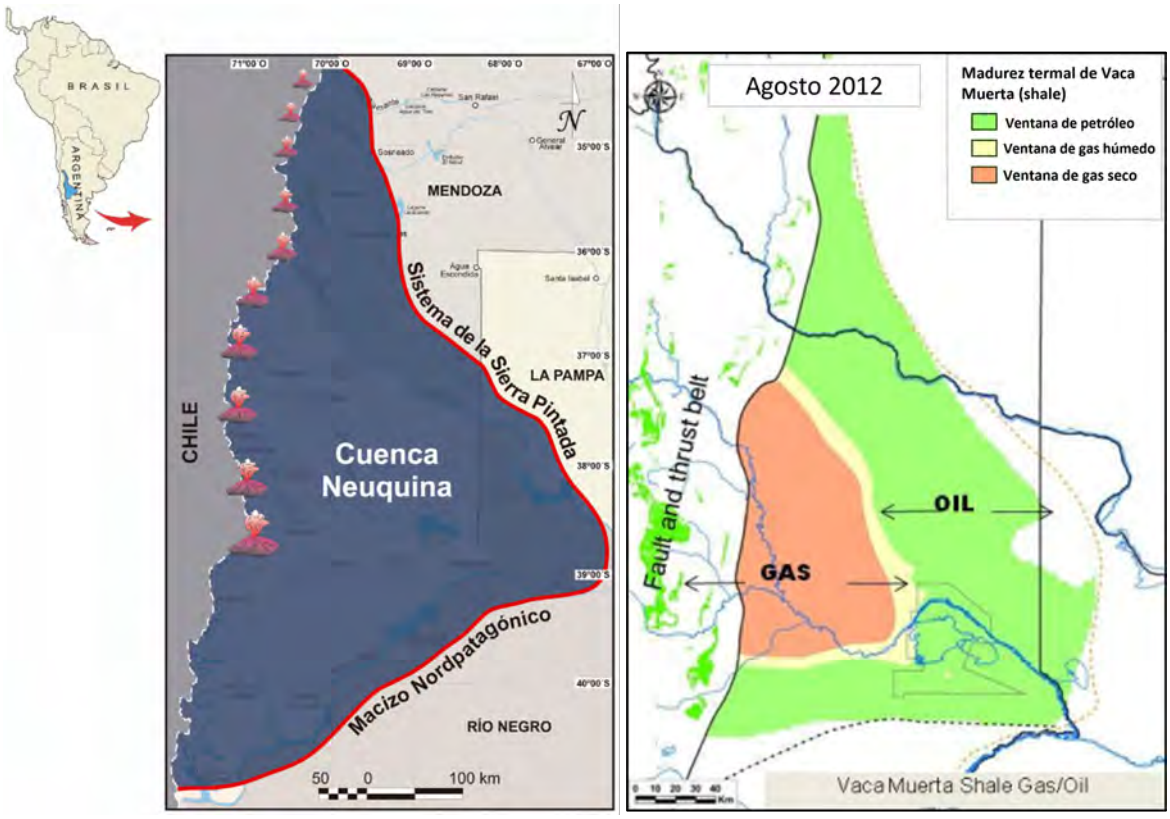


Figura 1.1: Ubicación geográfica de la cuenca Neuquina (izquierda) y la formación Vaca Muerta (derecha). Esta se encuentra mayoritariamente en la cuenca Neuquina. Fuentes: imágenes extraídas de Otharán (2020) (izquierda), y Lanusse et al. (2012) (derecha), respectivamente.

Más recientemente, y de acuerdo a los datos suministrados por el IAPG (Instituto Argentino del Petróleo y del Gas), la producción de petróleo y gas durante el año 2020 alcanzó los 27.9 y 45.1 Mm^3, respectivamente. En términos productivos, dicho instituto indica que durante el mismo año, aproximadamente el 54 % de la matriz energética del país estaba constituida por gas natural, mientras que el 31 % correspondía a la utilización de petróleo. Esto es, un total del 85 % de la energía requerida depende de los recursos hidrocarburíferos que se explotan, exportan e importan en el país. De estos números se desprende que más

de la mitad de dicha matriz depende directamente de la balanza comercial del gas. Por ello, la cuenca Neuquina es, actualmente, el principal foco de los esfuerzos productivos del país.

Desde el punto de vista geológico, la cuenca Neuquina es de tipo sedimentaria marina. Su formación tiene sus orígenes entre los períodos del Triásico tardío, cuando la región estaba dominada por eventos extensionales de tipo *rifting* (Ortiz et al., 2016), abarcando el período del Jurásico inferior, y extendiéndose hasta el Cretásico inferior. Entre estos dos últimos períodos, la región fue inundada por el océano en reiteradas ocasiones, generando la depositación de sedimentos alternadamente y dando origen a varias de las rocas madres y reservorios no-convencionales que la componen (Barredo and Stinco, 2013; Romero-Sarmiento et al., 2017). Desde el punto de vista petrolero, las unidades más importantes de la cuenca Neuquina son la formación "Los Molles", la formación "Vaca Muerta", y por último, la formación "Agrio". De estas tres, Vaca Muerta se destaca como la roca madre más importante de Argentina, con un espesor variable de entre 20 y 450 m (Stinco and Barredo, 2014) y desde hace cerca de dos décadas, considerada el reservorio no-convencional de hidrocarburos más relevante de la región (Ortiz et al., 2016). Su extensión geográfica comprende casi la totalidad de la superficie de la cuenca Neuquina. La misma puede apreciarse en la Figura 1.1 (derecha). Esta formación se define, principalmente, como *shale-oil* y *shale-gas*, es decir, petróleo inmerso en esquistos y gas de lutita. El contenido de material orgánico presente en dichas arcillas es elevado, lo que la sitúa como un prospecto petrolero por excelencia. En contraparte, este tipo de rocas se caracterizan por tener una permeabilidad extremadamente baja. Esto es, aun cuando la misma tiene una porosidad capaz de alojar hidrocarburos, los poros que forman la matriz de la roca no se encuentran conectados, lo que no permite el desplazamiento interporal del fluido, impidiendo así que el recurso pueda ser extraído por métodos convencionales o en volúmenes económicamente rentables (Holditch and Madani, 2010). Para el caso particular de Vaca Muerta, por ejemplo, la permeabilidad puede registrar valores de entre 140 a 213 nD (nano-Darcy) (Romero-Sarmiento et al., 2017). En cuanto a sus reservas, en 2012 la compañía YPF estimó una cantidad de 22500 millones de barriles equivalentes de petróleo (BEP), mientras que la Agencia de Información Energética informó números que ascendían hasta los 27000 M de BEP, aumentando así las reservas declaradas para ese momento en un factor de diez (AIE, 2013). En la actualidad, portales oficiales[*] sitúan al reservorio en el segundo lugar mundial en importancia como recurso no-convencional de gas y en cuarto lugar como recurso no-convencional de petróleo. En términos económicos, el estado nacional comunica inversiones, ya sea anunciadas o en curso, por un total de 21.168 millones de dólares norteamericanos.

[*] https://www.argentina.gob.ar/economia/energia/vaca-muerta/inversiones
https://www.ypf.com/desafiovacamuerta/Paginas/vaca-muerta.html

Por muchos años, este tipo de rocas no-convencionales, con alto potencial hidrocarburífero pero muy baja permeabilidad, no atrajo la atención de la industria debido a los altos costos de producción, asociados principalmente a su extracción. En cambio, los esfuerzos eran concentrados en la explotación de aquellos reservorios que presentaban las condiciones necesarias para extraer el recurso mediante técnicas de recuperación primaria. Esto es, reservorios donde la porosidad y permeabilidad fuesen suficientes para que la propia presión de la roca empuje el fluido hasta la boca de pozo con poca o nula necesidad de estímulos externos (Holditch and Madani, 2010). Naturalmente, el crecimiento de la mayoría de las economías mundiales, cuyas matrices productivas se desarrollaron dependiendo mayoritariamente de la explotación de recursos fósiles, trajo aparejado una mayor demanda de hidrocarburos. Bajo este panorama, este escenario de producción hidrocarburífera abocado al desarrollo de los reservorios convencionales pronto comenzaría a ser insuficiente para abastecer la demanda mundial de energía. En efecto, de acuerdo a la *International Energy Agency* (IEA), la generación global de electricidad en el año 2019 fue estimada en unos 18508.37 TWh* (teravatio-hora), de los cuales 747.17 TWh (4.04 %) y 6346.00 TWh (34.29 %) correspondieron a la utilización petróleo y gas, respectivamente. Para el caso específico de Estados Unidos, y según la *U.S Energy Information Administration* (EIA, 2019), la matriz energética marcó un consumo total de 100.293 QBtu (QBtu: cuadrillones de Btu**), donde 80.358 QBtu (80.12 %), 8.452 QBtu (8.42 %), y 11.350 QBtu (11.31 %) correspondieron al uso de combustibles fósiles, energía nuclear, y renovables, respectivamente, mientras que la producción energética arrojó valores de 101.403 QBtu, o análogamente, 81.425 QBtu (80.29 %), 8.452 QBtu (8.33 %), y 11.527 QBtu (11.36 %). Una matriz que, al igual que Argentina y el resto de los países del mundo, depende claramente de la industria hidrocarburífera. Asimismo, informa que el consumo de energía mundial para el 2050, considerando fuentes de tipo nuclear, carbón, gas natural, petróleos y otros líquidos, y energías renovables, está pronosticado a crecer en un 50 % respecto de los niveles actuales. Solo la demanda de petróleo se estima que crecerá desde aproximadamente 100 Mb/d (Mb/d: millones de barriles diarios) en 2019, a unos 103 Mb/d en 2025 (IEA, 2020), considerando el escenario pos-COVID19.

1.2. Factibilidad y microsísmica

En vistas de la creciente demanda de hidrocarburos para abastecer las necesidades energéticas globales, durante las últimas dos décadas, la industria petrolera aumentó considerablemente su interés en la explotación de los reservorios no-convencionales. En parte, este cambio de paradigma productivo puede atribuirse a que la explotación de hidrocar-

*equivalente a 10^{12} vatios-hora.

**Btu: unidad térmica británica.

buros en reservorios convencionales ha comenzado a mostrar señales de dificultad para abastecer la mencionada demanda (Ali and Aftab, 2020). En la región, la formación Vaca Muerta apareció en escena como el principal productor de gas no-convencional de Latinoamérica. A la par con el interés de la industria en este tipo de reservorios, surgió la necesidad de diseñar y aplicar estrategias que fuesen económicamente rentables, y a su vez, técnicamente viables. Actualmente, el desarrollo de este tipo de reservorios demanda inversiones considerablemente mayores a las requeridas para la explotación de reservorios convencionales. Esto se debe principalmente a la baja permeabilidad de la roca en este tipo de yacimientos, lo que naturalmente implica que la extracción del recurso sea más lenta y dificultosa. Como consecuencia, la cantidad de perforaciones que deben realizarse para que estos yacimientos sean económicamente rentables es notablemente superior a las requeridas en situaciones convencionales (Holditch and Madani, 2010). En general, este tipo de reservorios son explotados con pozos de tipo horizontales, en geometrías "multipozos", de larga longitud, y fracturados lateralmente en toda su extensión (Saldungaray and Palisch, 2012). Lo que se busca es maximizar el volumen de roca de la cual extraer los fluidos. Además, debido a que este tipo de formaciones se caracteriza por poseer muy baja permeabilidad, estos reservorios presentan un bajo nivel de flujo de hidrocarburos, por lo que su producción depende de la presencia de fracturas que faciliten la movilidad de dichos fluidos. Cuando dichas fracturas no se presentan en forma natural, deben crearse de forma artificial. La generación artificial de fracturas es una tarea compleja y altamente costosa, que se conoce comúnmente como "tratamiento de fracturación hidráulica" (TFH). La idea principal de esta técnica implica la inyección de fluidos a altas presiones a través de las bocas de los pozos a estimular. Estas presiones deben ser suficientemente altas para que el fluido logre romper la estructura comprimida de la roca, generando las fracturas deseadas. Durante un TFH, los fluidos utilizados son combinados con arenas y/u otros agentes de sostén con el fin de generar el apuntalamiento de las "microfracturas" generadas. El objetivo de este apuntalamiento es evitar que los intersticios y "caminos" generados por las fracturas vuelvan a cerrarse, facilitando el flujo y la extracción de los hidrocarburos. Asimismo, los fluidos suelen contener químicos que reducen la fricción y facilitan la lubricación del espacio fracturado (Gupta et al., 2021). Esto último es necesario ya que la red de fracturas artificiales generado por el TFH no logra alterar la matriz de permeabilidad global de la roca lo suficiente como para posibilitar el transporte del hidrocarburos por largas distancias. En términos económicos, la eficiencia de un proceso de TFH está íntegramente relacionada a la producción futura del pozo (Castano et al., 2010). Por ello, una vez fracturado, el potencial hidrocarburífero del pozo estará ligado al volumen que fue efectivamente estimulado por la fractura. Para calcular este volumen es necesario conocer la distribución global de la red de fracturas, así como también la geometría de las fracturas individuales. Considerando que estos procesos de fracturación se llevan a cabo a centenas o

miles de metros de profundidad, y bajo condiciones de presión y temperaturas extremas, la medición directa de estas cantidades resulta imposible con las técnicas y tecnologías disponibles a la fecha. Al momento, la única manera de obtener estas cantidades es infiriendolas mediante mediciones indirectas. En este sentido, la sísmica global proporciona el marco teórico necesario para obtener información de la geometría de fracturas. Sin embargo, y considerando que las fracturas generadas por un proceso de TFH tienen magnitudes con varios órdenes de magnitud menor que las manifestadas en eventos globales, la técnica se adaptó a pequeñas escalas en lo que se conoce como "microsísmica". Se refiere con este nombre (o HFM: *hydro-fracture monitoring*), al conjunto de técnicas orientadas al monitoreo espacio-temporal de las microfracturas generadas durante un TFH (Kendall et al., 2011; Goodway, 2012). La técnica consiste en el despliegue de "micrófonos" (Kamata et al., 2008) especialmente diseñados para detectar las señales sísmicas que se producen cuando se fractura la roca y que viajan a través del medio circundante. Estos sensores, llamados "geófonos" son desplegados en pozos aledaños a la zona estimulada (Wu et al., 2016), o en arreglos de superficie (Duncan Peter and Eisner, 2010; Baig et al., 2012) ubicado sobre el terreno inmediatamente superior a la región que está siendo fracturada. En cualquier caso, el objetivo es registrar la vibración del terreno a fin de distinguir los arribos sísmicos generados por una microfractura. Tal como se hace en la sismología global, durante un monitoreo microsísmico se aplican estrategias y algoritmos a fin de detectar los arribos de las fases de onda y, posteriormente, para localizar los epicentros de estas microfracturas o "eventos microsísmicos". Dicho esto, la variedad de técnicas de detección y localización de microsismos ha evolucionado y crecido a la par de la explotación de estos reservorios, y en la actualidad, existen innumerables estrategias para abordar estos problemas, algunas de las cuales no se derivan de la sismología global. Algunos ejemplos de algoritmos de detección y localización microsísmica pueden leerse en Drew et al. (2005); Michaud and Leaney (2008); Zimmer (2010); Song et al. (2010); Sabbione and Velis (2012); Velis et al. (2015); Sabbione et al. (2014); Mousavi et al. (2016); Akram et al. (2017); Chen (2018); Huang et al. (2018); Binder and Tura (2020); Qu et al. (2020). En general, la aplicación de cualquiera de estas técnicas arroja buenos resultados en términos de detección y/o localización. Sin embargo, es importante mencionar que un típico ambiente de monitoreo microsísmico está caracterizado por una baja relación señal-ruido. En otras palabras, las amplitudes de los arribos suelen estar enmascarados total o parcialmente por el ruido de fondo. Así, el éxito de estas técnicas está ligado al contenido de ruido de la señal y/o de nuestra capacidad para filtrarlo. Por ello, previo o simultaneo a la detección, es usual aplicar técnicas de filtrado de ruido. Para esta tarea, suelen aplicarse estrategias tradicionales como filtros pasabanda, o también algunas más complejas. Entre estas últimas se encuentran algoritmos basados en descomposiciones en modos empíricos (Torres et al., 2011; Wu and Huang, 2009; Bekara and Van der Baan, 2009; Gómez and Velis, 2016), transformadas diseñadas específicamente

para el filtrado de ruido (Pinnegar and Eaton, 2003; Sabbione et al., 2013a, 2014; Zhang and van der Baan, 2018), descomposiciones por valores singulares (Chen and Sacchi, 2014; Velis et al., 2015), u otras (Schimmel and Gallart, 2007; Eisner et al., 2008).

Luego del filtrado de ruidos y la detección de eventos microsísmicos, estos últimos deben ser localizados. La localización de cada evento suele visualizarse como un punto individual en un espacio 3D. En el mejor de los casos, la coordenada espacial de cada evento está acompañada de una elipsoide de incerteza o error (Maxwell, 2009a). Los factores que más influyen en el tamaño de este elipsoide suelen ser el contenido de ruido de la señal y errores en el modelo de velocidades utilizado en el procesamiento, entre otros. El conjunto total de eventos localizados suele formar una "nube" de eventos. Esta nube es un producto importante para la microsísmica, ya que permite dimensionar, a grandes rasgos, el tamaño (ancho, alto y largo) de la fractura. Conocer las dimensiones resulta esencial para estimar el volumen de reservorio estimulado (SRV, por sus siglas en inglés), así como la geometría de la red de fracturas. Más aun, un estudio más profundo sobre las señales individuales puede brindar información sobre el tipo de fractura que dio origen al microsismo. En general, estos eventos se originan como deslizamientos de cizalla donde se libera energía y suelen ocurrir en planos de debilidades pre-existenes de la roca, que se deforman y/o desplazan al sufrir cambios en la tensión y presión poral durante el fracturamiento hidráulico (Cipolla et al., 2012; Rutledge and Phillips, 2003). El cálculo y estudio de los mecanismo de fractura de la roca se denomina "inversión de tensor momento" (ITM o MTI, por sus siglas en inglés). El mecanismo focal o tensor momento (TM) es uno de los productos más valiosos de la microsísmica. Combinar esta información con los datos de fractura (cantidad y tipo de fluido inyectado, presiones, tipos de propante utilizados, etc) permite reducir la incerteza en el cálculo del SRV. Claramente, la microsísmica es una herramienta que permite caracterizar la fractura hidráulica, y entre sus aplicaciones, permite actualizar los modelos geomecánicos del subsuelo, mejorar los diseños de tratamiento y "completación" de los pozos (Cipolla et al., 2010), así como también contribuye a mejorar la eficiencia de futuros TFH en dicho reservorio. Sin embargo, es cuando el monitoreo microsísmico se realiza en tiempo real cuando la utilidad de esta técnica tiene mayor relevancia. En este caso, la microsísmica es una herramienta fundamental para distinguir riesgos y evitar accidentes de varias índoles. Por ejemplo, el monitoreo en tiempo real puede brindar la información necesaria para detener el TFH si se detecta la activación de una falla natural (Maxwell et al., 2009). En general, este tipo de eventos es indeseable, pudiendo causar la pérdida del pozo a nivel productivo. En el peor de los casos, la activación de una falla puede incluso proveer una conexión entre los fluidos de fractura (altamente contaminantes) con un reservorio de agua natural (Downie et al., 2010). También, puede servir para prevenir posibles conexiones entre la fractura y pozos adyacentes, lo que en la industria se conoce como *fracture-hit* (Gupta et al., 2021).

Como vemos, la microsísmica resulta esencial tanto para controlar el proceso de inyección como para mapear el desarrollo del reservorio (Eisner et al., 2011a). El conocimiento de los diversos mecanismos asociados a las fracturas generadas tiene un impacto inmediato en el diseño de las estrategias del proceso de inyección hidráulica, así como en el conocimiento de las condiciones geológicas y el comportamiento geomecánico del reservorio de interés (Van Der Baan et al., 2013a). Esta Tesis tiene como finalidad la de crear estrategias y métodos con aplicación directa en la microsísmica, y contribuir, como objetivo final, a aumentar el conocimiento de esta disciplina en el país.

1.3. Antecedentes y motivación

El monitoreo microsísmico se ha convertido en una herramienta importante no solo para evaluar y estudiar los tratamientos de fractura hidráulica (Eisner et al., 2010a; Maxwell et al., 2010), sino también para evaluar procesos subterráneos relacionados, por ejemplo, con la exploración geotérmica, y la vigilancia minera y de yacimientos (Warpinski, 2009). Además, el análisis de señales microsísmicas ha encontrado aplicación en muchos estudios y procesos de la industria petrolera (Maxwell and Urbancic, 2001; Kendall et al., 2011). Los estudios de microsísmica tienen como objetivo detectar, localizar y, eventualmente, caracterizar las microfracturas generadas por los fenómenos antes mencionados (Van Der Baan et al., 2013b). La magnitud de estos eventos suele ser muy pequeña, con valores generalmente negativos (Grechka and Heigl, 2017), y a menudo exhibiendo amplitudes comparables a las amplitudes del ruido de fondo (Maxwell, 2005; Shemeta and Anderson, 2010), lo que representa un gran desafío para los algoritmos de detección. A su vez, la incertidumbres en el picado de tiempos de arribo en escenarios de baja relación S/R afectan significativamente la precisión en la estimación del hipocentro (Akram and Eaton, 2016). Simultáneamente, los monitoreos microsísmicos suelen ser registrados mediante geometrías de adquisición que no son favorables para la aplicación de algunas de las técnicas de inversión más utilizadas en el campo de la geofísica de reservorios. Además, tanto el modelo de velocidades, como la geometría del subsuelo, no suelen conocerse con precisión y deben ser calibrados o modelados adecuadamente. Más aún, puesto que el medio de propagación de las ondas suele ser particularmente complejo, la posibilidad de inferir información precisa y confiable sobre las características geomecánicas de las fracturas es bastante limitada (Eisner et al., 2009, 2011b; Poliannikov et al., 2012). La combinación de estos factores conduce, generalmente, a una gran incertidumbre en la estimación de los diversos parámetros de interés. Dicho de otra manera, la información que contienen las ondas generadas por los microsismos está contaminada por ruido de diferente naturaleza y es incompleta. Por estas razones, el procesamiento de las señales microsísmicas con el objetivo de detectar y determinar los hipocentros de los eventos en el dominio espacio-tiempo (Zimmer and Jin, 2011) y la ca-

racterización geométrica y geomecánica de las estructuras involucradas (Chorney et al., 2012), representa un gran desafío científico y tecnológico (Eisner et al., 2011b; Maxwell, 2011; Van Der Baan et al., 2013a; Kendall et al., 2019).

Dicho esto, la obtención de información acerca del tipo de fracturas que se generan durante un proceso de fracturación hidráulica, o de la orientación de las mismas, resulta esencial para el desarrollo económico del yacimiento. Así, su obtención es crucial desde el punto de vista estratégico y económico en la explotación de los recursos (Baig and Urbancic, 2010; Maxwell, 2014) de dicho yacimiento. Una estimación de este tipo de información se puede hallar mediante la inversión de los mecanismos focales (Baig and Urbancic, 2010; Eisner et al., 2010b; Van Der Baan et al., 2013a; Maxwell, 2014). Debido a que los microsismos se suelen caracterizar por desplazamientos de cizalla similares a los generados por un terremoto natural, es posible aplicar los mismos principios de la sismología global (Aki and Richards, 2002; Shearer, 1999a) para estudiar y analizar el mecanismo focal y los parámetros asociados a las microfracturas. Las principales limitaciones de la microsísmica están ligadas a la baja relación señal-ruido presente en la mayoría de los registros, y a la escasa cobertura espacial (apertura angular) producto de la deficiente geometría de los pozos de inyección y de monitoreo (Vavryčuk, 2007; Warpinski and Du, 2010a).

El monitoreo de los procesos de fracturación hidráulica mediante un solo pozo monitor (por lo general vertical y recto), configuración habitual en la República Argentina y en la mayoría de los relevamientos microsísmicos del mundo, resulta en un severo problema de no-unicidad en la inversión del tensor momento. Estudios recientes han permitido establecer las bases para mitigar este problema cuando se utiliza un solo pozo monitor y se satisfacen ciertas condiciones (Nolen-Hoeksema and Ruff, 1999; Vavryčuk, 2007; Vera Rodríguez et al., 2011; Grechka, 2015a; Yu et al., 2015b). El monitoreo mediante múltiples pozos o arreglos superficiales de amplio acimut permitiría, en principio, aliviar el problema de no-unicidad. Sin embargo, esta última estrategia es más costosa y, en general, los datos sísmicos presentan menor relación señal-ruido en comparación a los registrados en monitoreos de pozo. Por otro lado, la búsqueda de mayor cobertura espacial no sería estrictamente necesaria cuando el medio presenta anisotropía (Grechka, 2015a), por lo que este tipo de arreglos aun es poco utilizado en la industria.

En este contexto se plantea entonces la necesidad de resolver los mencionados problemas asociados a la microsísmica con el fin de contribuir al desarrollo de este tipo de tecnologías en la República Argentina, como son la creación de algoritmos relacionados a la detección, filtrado de ruido, localización de eventos, inversión de tensor momento, interpretación (y otros), y mejorar la eficiencia relacionada a los complejos procesos de la fracturación hidráulica en general.

1.4. Objetivos y contribuciones de la Tesis

El objetivo general de esta Tesis consiste en el desarrollo de algoritmos especializados para el tratamiento de la información microsísmica con aplicación al estudio y monitoreo de procesos de inyección hidráulica en la exploración y explotación de reservorios de hidrocarburos no convencionales. Así, a lo largo de este documento se desarrollan diversas herramientas alternativas para el procesamiento de las señales microsísmicas y cuyo desempeño representa un aporte frente a la técnicas usualmente adoptadas por la industria. Es importante mencionar que, tanto las técnicas diseñadas como los resultados apuntan a contribuir al desarrollo de este tipo de tecnologías en la República Argentina. Por esta razón, los principales objetivos científicos de esta Tesis están en buena medida focalizados y motivados por el creciente desarrollo del yacimiento no convencional de Vaca Muerta, Neuquén. En pos de lograr los objetivos arriba mencionados:

- Se diseñó una estrategia para evaluar el impacto que tiene un algoritmo de filtrado de ruido sobre una señal microsísmica. Este algoritmo está basado en la comparación de histogramas de polarización construidos a partir de la señal ruidosa y la señal limpia o filtrada. Mientras que la mayoría de los estudios sobre el desempeño de los diferentes algoritmos de filtrado suele hacerse de forma cualitativa (ver Secciones 2.1 y 2.7), la principal contribución de este trabajo es brindar un análisis cuantitativo de dicho impacto. Contar con una cuantificación de dicho desempeño tiene una consecuencia directa sobre la etapa de decisión que refiere a la elección de un método de filtrado por sobre otro. Desde el punto de vista científico, los resultados de este trabajo dieron lugar a una publicación en la revista *Geophysics* y fueron presentados en el congreso internacional ICE2019, organizado por la AAPG y celebrado en Buenos Aires, Argentina.

- Se diseñó un algoritmo de localización de eventos microsísmicos basado en el método de optimización global denominado “Evolución diferencial”. Los resultados de su aplicación mostraron que es una estrategia viable para la resolución de este problema. El tiempo de cómputo para la implementación de este método resultó comparable al de otros algoritmos ya probados y publicados como son *particle swarm optimization* y *very fast simulated annealing*. Así, el algoritmo podría sumarse a la batería de métodos que se utilizan para la localización de eventos microsísmicos en “tiempo real”. Este trabajo fue presentado en la conferencia “2017 XVII Workshop on Information Processing and Control (RPIC)” auspiciada por la IEEE y organizada por el Instituto de Investigaciones científicas y Tecnológicas (ICYTE, CONICET-UNMDP). Este trabajo se publicó en el *proceedings* de la misma conferencia.

- Se diseñó un modelo de redes neuronales profundas para resolver el problema de

la inversión de tensor momento sísmico. Los resultados de su aplicación mostraron que el mismo es una alternativa fiable a los métodos tradicionales utilizados para esta tarea, como son los métodos determinísticos. En términos de contribución, este trabajo demuestra la viabilidad de utilizar un método del "aprendizaje automático" para resolver un problema de inversión de tensor momento en una geometría de dos pozos en medios iso/anisotrópicos. El mismo fue presentado en la "2021 XVII Workshop on Information Processing and Control (RPIC)", auspiciada por la IEEE y organizada por el Instituto de Automática de la Universidad Nacional de San Juan y CONICET (UNSJ-CONICET). El trabajo fue publicado en el *proceedings* de la misma conferencia.

1.5. Organización de la Tesis

El contenido de esta Tesis está organizado en cinco capítulos. El Capítulo 2 describe la estrategia de histogramas de polarización diseñada para la evaluación de las técnicas de filtrado de ruido. En el Capítulo 3 se muestra el diseño de un algoritmo de localización de eventos microsísmicos mediante Evolución diferencial.

En el Capítulo 4 se implementa un modelo de redes neuronales para la inversión de tensor momento. Por último, el Capítulo 5 enumera las conclusiones generales de esta Tesis. Adicionalmente, se incluyen dos Apéndices. En el Apéndice A se describen las ecuaciones necesarias para entender el problema directo relacionado al tensor momento, mientras que en el Apéndice B se detallan los conceptos fundamentales de las redes neuronales. A continuación, describimos el contenido de cada capítulo.

En el Capítulo 2 se muestra una estrategia para evaluar y comparar el desempeño de diferentes métodos de filtrado de ruido microsísmico. Comienza enunciando la importancia del filtrado de ruido (*denoising*) como parte del pre-procesamiento de datos microsísmicos. Luego, se muestra el desarrollo de una técnica diseñada para comparar el desempeño de diferentes algoritmos de filtrado de señales microsísmicas. La mencionada técnica realiza una comparación basada en las diferencias entre histogramas de polarización y nos permite dilucidar cuán cerca está la señal filtrada de la señal limpia. Para probar el método se utilizan tres estrategias de *denoising* diferentes, a saber: (1) filtrado mediante la transformada hiperbólica de Radon en un espacio de dominio restringido (Sabbione and Sacchi, 2016), (2) filtrado por reducción de rango mediante descomposición por valores singulares (Velis et al., 2015), y finalmente (3) un filtrado por descomposición en modos empíricos (Gómez and Velis, 2016). Adicionalmente, se compara utilizando un filtrado pasa-banda convencional y se realiza una validación mediante la medición del factor de calidad. Finalmente, cabe mencionar que los resultados expuestos están basados en la aplicación de la técnica sobre datos sintéticos y también de campo.

En el Capítulo 3 se expone el desarrollo y aplicación del método de optimización "Evolución diferencial", al problema de la localización de eventos microsísmicos. El capítulo comienza enunciando los fundamentos teóricos relacionados al método desarrollado. A continuación, se enuncia brevemente el problema de la localización de un evento microsísmico como un problema de optimización global. En este sentido, se trabaja para llegar a la expresión de la función de costo a minimizar como función de las diferencias de tiempo observado y calculados, lo que deja en evidencia su no-linealidad. Los resultados se obtienen tras aplicar esta estrategia sobre datos sintéticos limpios y ruidosos. Por completitud, también se realiza una comparación con el desempeño logrado por otros dos métodos de optimización diferentes: (1) *particle swarm optimization* y (2) *very fast simulated annealig*, cuyos resultados pueden encontrarse en Lagos et al. (2014).

En el Capítulo 4 se utilizan redes neuronales para resolver el problema de la inversión de tensor momento. Con el objetivo de analizar la capacidad de este método para resolver dicho problema, se trabaja con varios modelos de subsuelo y considerando diferentes geometrías de monitoreo realistas. El capítulo se divide en varias secciones. La primera parte está dedicada a introducir al problema específico, mientras que la segunda sección abarca la estructura de la red utilizada para resolver esta tarea y también detalla la geometría y datos utilizados para su testeo. A continuación, se describen los resultados de aplicar la red sobre estos modelos. Finalmente, las últimas secciones ofrecen una discusión, un anexo y conclusión en función de los resultados.

El Capítulo 5 destaca las principales conclusiones de esta Tesis. También describe las contribuciones científicas principales de este trabajo, que fueron presentadas en diversos eventos científicos y publicadas en revistas de alcance nacional e internacional.

El Apéndice A está dedicado a brindar los fundamentos teóricos necesarios para entender el problema directo relacionado al tensor momento. Para ello, se describen las ecuaciones que gobiernan el desplazamiento de una onda a través del medio. Entender las partes constitutivas de este desplazamiento nos permite tener una idea de la génesis del dato que luego será necesario para resolver el problema de "inversión de tensor momento".

Por último, el Apéndice B realiza una descripción teórico-matemática de las redes neuronales como herramientas de la "inteligencia artificial". El mismo comienza con una introducción general y continua con una sección dedicada a la definición de la "arquitectura" de las redes y sus partes fundamentales, como las "unidades" y "capas". Finalmente, una tercera y última sección detalla el concepto de "entrenamiento" de una red. La misma comienza con una descripción de las características que deben tener los datos que serán utilizados para un entrenamiento. Luego, se enumeran los principales tipos de entrenamiento existentes y los algoritmos necesarios para llevarlo a cabo. Por último, la parte final de esta sección abarca los fenómenos de sobre-ajuste (*overfitting*) y sub-ajuste (*underfitting*), cuyo entendimiento es crucial para evaluar el éxito de un entrenamiento.

Capítulo 2

Evaluación de filtrado de ruido mediante histogramas de polarización

Se presenta una comparación de métodos de filtrado de ruido para datos microsísmicos basada en su efecto sobre los atributos de polarización de las señales de tres componentes. Los métodos de filtrado de ruido utilizados en la comparación incluyen un filtrado pasabanda clásico y tres técnicas de filtrado de ruido no convencionales propuestas recientemente: filtrado de ruido por transformada hiperbólica de Radon en dominio restringido (Sabbione and Sacchi, 2016), un filtrado basado en la reducción de rango mediante descomposición de valores singulares (Velis et al., 2015), y finalmente, filtrado de ruido por descomposición en modos empíricos (Gómez and Velis, 2016). Para hacer la comparación, se aplican estos métodos sobre dato sintético contaminado con ruido extraído de registros de datos de campo (ruido real). Luego, se calculan tres atributos de polarización: rectilinealidad, acimut e inclinación, para finalmente ordenarse y organizarse en histogramas. La comparación se diseñó midiendo las distancias entre los histogramas de polarización entre los datos limpios y aquellos filtrados. Mediante los valores de estas distancias, se puede asumir que un método se desempeña mejor que otro cuando la distancia entre histogramas obtenida es menor que aquella calculada para el segundo método. Mientras que los métodos tradicionales para evaluar la calidad del filtrado de ruido se basan, típicamente, en el cálculo del error y factor de calidad, o en una inspección visual de los residuales calculados entre la señal original y la filtrada, esta estrategia permite cuantificar la mejora en los atributos de polarización obtenidos mediante los diferentes métodos de filtrado. Por completitud, también se calcula el factor de calidad de las señales, lo que agrega valor y robustez a la comparación. Los resultados indican que el método basado en la descomposición por valores singulares conserva los atributos de polarización originales en mejor medida que las otras técnicas probadas. Además, el mismo recupera la señal filtrada obteniendo mayores valores en el factor de calidad. Finalmente, probamos los métodos con datos de campo y

evaluamos su rendimiento cualitativamente sobre la base de la información obtenida de la pruebas numéricas con datos sintéticos.

2.1. Introducción

El filtrado de ruidos o *denoising* es una de las primeras y más importantes etapas de un procesamiento microsísmico. El objetivo de un proceso de *denoising* es remover la mayor cantidad de ruido posible de una señal sin dañar la misma. La estrategia de filtrado más simple es aplicar filtros pasabanda diseñados en el dominio de la frecuencia, como los conocidos filtro pasabanda clásicos (BPF, por sus siglas en inglés). Sin embargo, y debido a que el ruido y la señal suelen compartir contenido espectral, por lo general la aplicación de este tipo de filtros no es lo suficientemente efectiva (Vera Rodríguez et al., 2012; Mousavi et al., 2016). En este sentido, el diseño de algoritmos de filtrado más sofisticados y eficaces es una disciplina que continua en desarrollo.

Para tratar datos de baja relación S/R, Eisner et al. (2008) proponen el uso de un filtro emparejador o de matcheo. Para ello, se elige una señal picada sobre un evento con alta relación S/R como referencia para mejorar la detección de eventos con S/R más bajas y formas de onda similares (es decir, que se originaron con un mecanismo de fuente similar). La estrategia de emparejamiento está basada en correlaciones cruzadas. En el contexto de la sismología global, otros autores utilizan la transformada S (Stockwell et al., 1996), ya sea mediante el uso de umbrales para suprimir ruido no estacionario (Parolai, 2009), o para la construcción de filtros "dato-adaptativos" para atenuar el ruido no coherente utilizando la información proporcionada por la coherencia lateral en las fases de la matriz de datos (Schimmel and Gallart, 2007). En Vera Rodríguez et al. (2012) explotan la información proporcionada por los datos multicanal y propone un algoritmo denominado *group sparsity constrained time-frequency microseismic data*. Además, Sabbione et al. (2013b) presentan un método de filtrado para la remoción de ruido presente en datos microsísmicos 3C basado en la transformada hiperbólica de Radon con desplazamiento del vértice. Luego, desarrollan un algoritmo adaptativo de dos pasos para la detección y *denoising* usando un modelo restringido con una transformada de Radon parabólica en el espacio (Sabbione et al., 2015) que acelera los cálculos. La llamada transformada hiperbólica de Radon de dominio restringido (RHRT, por sus siglas en inglés), sin aproximación parabólica, se describe de forma concisa en Sabbione and Sacchi (2016). Del mismo modo, Velis et al. (2015) proponen un método de dos pasos que primero detecta los eventos utilizando una estrategia de reconocimiento de patrones y luego elimina el ruido aleatorio utilizando una aproximación por reducción de rango basada en la descomposición por valores singulares (SVD, por sus siglas en inglés). Por otro lado, en Han and van der Baan (2015) se combina un algoritmo de descomposición en modos empíricos por conjuntos (EEMD) (Wu and Huang, 2009) con

una estrategia de umbrales adaptativos para eliminar ruido aleatorio y coherente en datos microsísmicos y sísmicos con excelentes resultados. Más tarde, Zhang and van der Baan (2018) aplican con éxito la transformación de *shearlet* 3D para suprimir el ruido aleatorio de fondo para datos microsísmicos de múltiples componentes. Más recientemente, varios trabajos proponen utilizar técnicas de aprendizaje automático para eliminar el ruido de señales microsísmicas. Por ejemplo, Zhu et al. (2019) separan señales microsísmicas del ruido de fondo utilizando redes neuronales profundas, mientras que Shao et al. (2019) y Wang et al. (2020) usan técnicas de aprendizaje de diccionario para desarrollar métodos de *denoising* que también se pueden adecuar a aplicaciones microsísmicas. Igualmente, Zhang and van der Baan (2019) aplican el aprendizaje automático sin supervisión para realizar el *denoising* y la reconstrucción de datos microsísmicos incompletos y de mala calidad.

Como se puede ver, hay una amplia variedad de métodos presentados en la literatura que están dedicados al *denoising*. En general, todos superan las técnicas contra las que se los comparan, lo que lleva a relaciones S/R más altas y señales más limpias a partir de las cuales puede extraerse información más confiable (por ejemplo, atributos de polarización, tiempos de arribo de las señales, etc). No obstante, estas conclusiones a menudo no están soportadas de forma cuantitativa. En otras palabras, a pesar de que todos los enfoques y métodos de filtrado descriptos son herramientas efectivas que mejoran la calidad de la señal, no presentan resultados que permitan cuantificar su impacto en la estimación de parámetros clave de interés. Por ejemplo, en Vera Rodríguez et al. (2012) comparan su método contra un BPF analizando mapas de tiempo-frecuencia y mediante el uso de la inspección visual de la señales filtradas y sus correspondientes hodogramas. Han and van der Baan (2015) comparan su enfoque EEMD con BPF y un algoritmo de búsqueda de bases (Chen et al., 2001), y también concluyen analizando visualmente las señales filtradas y sus correspondientes espectros. Nuevamente, Sabbione et al. (2015) evalúan el desempeño de su estrategia basada en la transformada Radon mediante una inspección visual de las señales antes y después de la supresión de ruido. En Velis et al. (2015) comparan sus resultados obtenidos mediante el filtrado por reducción de rango por valores singulares contra aquellos obtenidos por el método de la transformada de Radon con desplazamiento de vértice propuesto por Sabbione et al. (2015). Para la comparación, se basan tanto en el análisis visual de los residuos de la señal, como en la correlación cruzada entre las señales limpia y filtrada.

En este capítulo, se presenta un análisis cuantitativo sobre el rendimiento de los algoritmos de filtrado microsísmico. El mismo evalúa el impacto que produce la aplicación de los filtros sobre la estimación de atributos clave de la señal. Es decir, en lugar de hacer foco en la preservación de la forma de onda, se centra en los parámetros que se estiman a partir de las señales filtradas. Con este fin, se miden las distancias entre los histogramas de los atributos de polarización estimados a partir de ventanas superpuestas sobre las señales

limpias y filtradas. Esta técnica fue introducida por Jones et al. (2016) para evaluar las similitudes en la polarización sísmica en el contexto de identificación de primeros arribos, el diagnóstico de efectos regionales de carácter local y detección de sensores desalineados y/o ruidosos. Para el análisis en esta Tesis, se seleccionan cuatro métodos (BPF, RHRT, SVD y EMD) y se comparan sus desempeños en la estimación del acimut, la inclinación y la rectilinealidad. Con este propósito, primero se generan datos microsísmicos sintéticos de polarización conocida y luego se contaminan con ruido real extraído de un registro de campo. Para esto, se considera una amplia gama de rangos para la relación S/R y se repite, con fines estadísticos, el experimento 100 veces para cada caso. Suponemos que un método supera a otro en la estimación de un atributo dado si la distancia del histograma correspondiente es menor que la del otro método. Además, y como soporte de los resultados, también se calcula el factor de calidad (QF, por sus siglas en inglés) de las señales filtradas.

Finalmente, se presenta un apartado con datos de campo para mostrar la capacidad de filtrado de los métodos en datos reales e ilustramos cómo realizar una comparación de sus desempeños basándonos en la distancia de histogramas. Los principales resultados demuestran que cuando la relación S/R es pequeña o moderada, se destaca el desempeño de los métodos de *denoising* más sofisticados (particularmente SVD y, en menor medida, RHRT), mientras que para escenarios de alta relación S/R conviene usar filtros más convencionales, o menos complejos, tales como BPF o EMD.

2.2. Teoría y métodos

Para evaluar el rendimiento de los algoritmos de filtrado se comparan los atributos de polarización estimados a partir de los datos limpios con aquellos estimados a partir de los datos filtrados. Para ello, seguimos el análisis cuantitativo sobre la similaridad de la polarización presentado por Jones et al. (2016), donde se propone medir la distancias entre histogramas construidos con atributos de polarización. Para esta tarea, debemos realizar una serie de pasos. Primero, realizamos *denoising* sobre un gran conjunto de datos sintéticos con los cuatro métodos a testear. En segundo lugar, calculamos los parámetros de polarización de los datos limpios y filtrados. Finalmente, se construyen los histogramas correspondientes para cada atributo y se miden las distancias entre los histogramas que corresponden a los datos limpios y los filtrados.

Atributos de polarización

Existen numerosos métodos presentados en la literatura para calcular atributos de polarización de una señal sísmica. Ejemplos de los atributos calculados más utilizados son el

acimut, la elipticidad, la inclinación de la dirección de máxima polarización, la planaridad y la rectilinealidad. Una explicación detallada de esta colección de atributos puede encontrarse en Vidale (1986) y Jurkevics (1988), quienes presentan diferentes métodos para su estimación. Por ejemplo, los atributos calculados en Vidale (1986) se basan en la matriz de covarianza estimada a partir de la señal analítica, lo que permite calcular atributos instantáneos, mientras que aquellos derivados en Jurkevics (1988) se basan en la matriz de covarianza estimada a partir de ventanas de tiempo de la señal sísmica de entrada. Este último es el enfoque que adoptamos en este estudio para estimar los atributos de polarización. En particular, nos centramos en el acimut de la onda P, inclinación de la dirección de máxima polarización y en la rectilinealidad, que se estima a partir de los autovalores y autovectores de la matriz de covarianza de los datos (Flinn, 1965; Montalbetti and Kanasewich, 1970; Vidale, 1986; Jurkevics, 1988).

Sea $\mathbf{s} = [\mathbf{s}_1, \mathbf{s}_2, \mathbf{s}_3]$ una señal sísmica 3C registrada mediante un único sensor de tres componentes. La matriz de covarianza $\mathbf{C}$, de dimensión 3×3, puede calcularse a partir de las muestras del dato extraídas de una ventana de longitud L_c (Jurkevics, 1988) y puede escribirse como:

$$\mathbf{C} = \begin{pmatrix} c_{xx} & c_{xy} & c_{xz} \\ c_{yx} & c_{yy} & c_{yz} \\ c_{zx} & c_{zy} & c_{zz} \end{pmatrix}, \tag{2.1}$$

donde se usa la convención de índices $x \equiv 1$, $y \equiv 2$, $z \equiv 3$. Los elementos de esta matriz pueden calcularse mediante la expresión:

$$c_{kl} = \frac{\mathbf{s}_k \, \mathbf{s}_l^T}{L_c} = \frac{1}{L_c} \sum_{i=1}^{L_c} s_{ki} \; s_{li}, \tag{2.2}$$

donde $1 \leq k, l \leq 3$ indica la componente del dato e i el índice de la muestra correspondiente.

Los autovalores y autovectores de la matriz $\mathbf{C}$ se obtienen resolviendo:

$$(\mathbf{C} - \lambda \mathbf{I})\mathbf{u} = \mathbf{0}. \tag{2.3}$$

Se busca ahora las tres soluciones no triviales de la ecuación 2.3, λ_1, λ_2 y λ_3, y sus autovectores correspondientes, $\mathbf{u}_1$, $\mathbf{u}_2$ y $\mathbf{u}_3$. Por conveniencia, los autovalores (y sus respectivos autovectores) se ordenan de modo que $\lambda_1 \geq \lambda_2 \geq \lambda_3$. Tomando en cuenta que $\mathbf{C}$ es una matriz real, simétrica y definida semi-positiva, λ_1, λ_2 y λ_3 son reales y no-negativos. Cabe destacar que $\mathbf{C}$ es la representación matricial de una elipsoide con ejes principales definidos por $\lambda_1 \, \mathbf{u}_1$, $\lambda_2 \, \mathbf{u}_2$ y $\lambda_3 \, \mathbf{u}_3$. Esta elipsoide es conocida como elipsoide de polarización y su determinante define el movimiento de la partícula en dicha ventana.

A partir de lo mencionado arriba, la rectilinealidad queda definida por:

$$\rho = 1 - \frac{\lambda_2 + \lambda_3}{2\lambda_1}, \tag{2.4}$$

lo que implica que $\rho \in [0, 1]$. Una señal con valores de ρ cercanos a 1 tiene un movimiento de partícula definido por una elipsoide de polarización mayormente elongada, lo cual es esperable en señales de tipo compresionales puras (donde la partícula se desplaza en el sentido de propagación de la onda).

Para una onda compresional P, el acimut se puede estimar como:

$$\theta = \tan^{-1}\left(\frac{u_{1x}}{u_{1y}}\right), \tag{2.5}$$

donde u_{1x} y u_{1y} son las componentes horizontales del autovector principal $\mathbf{u}_1$. Se deduce de la expresión 2.5 que $\theta \in (-90°, 90°]$.

La inclinación de la dirección de máxima polarización para una onda de tipo compresional pura se mide a partir de la componente vertical mediante:

$$\phi = \tan^{-1}\left(\frac{u_{1z}}{\sqrt{u_{1x}^2 + u_{1y}^2}}\right). \tag{2.6}$$

Como en el caso anterior, $\phi \in (-90°, 90°]$.

Para la construcción de $\mathbf{C}$, es recomendable seleccionar una ventana de tiempo con un número impar de muestras, permitiendo así asignar los atributos de polarización de dicha matriz a la muestra central de dicha ventana. Finalmente, se calculan los atributos $\rho(t)$, $\theta(t)$ y $\phi(t)$, en función del tiempo, simplemente desplazando la ventana de tiempo a lo largo del registro completo del dato.

Histogramas de polarización

Formalmente, se define a un histograma H como una colección de valores de un cierto atributo que son almacenados en N_b intervalos llamados "bines". Sea $h_{i,\alpha}$ el i-ésimo bin de un histograma H_α para un dado receptor y un atributo de polarización α, donde α puede referirse a cualquiera de los mencionados (ρ, θ, o ϕ). Debido a que usamos una ventana de tiempo corta que se desplaza a lo largo de dicho canal, siguiendo Jones et al. (2016), completamos cada bin $h_{i,\alpha}$ con la suma de las trazas de las matrices de covarianza que corresponden a un dato cuyo atributo cae dentro del i-ésimo bin. Así,

$$h_{i,\alpha} = \sum_{t:\alpha(t)\in i} Tr\{\mathbf{C}(t)\}, \tag{2.7}$$

donde $Tr\{\mathbf{C}(t)\}$ es la traza de la matriz de covarianza para el tiempo t y $\alpha(t)$ es el atributo de polarización correspondiente. En otras palabras, el histograma es pesado por la energía sísmica promedio, lo cual disminuye el impacto del ruido en el histograma (Jones et al., 2016).

Distancia entre histogramas

Se puede utilizar la distancia entre dos histogramas para cuantificar qué tanto se parecen, o equivalentemente, cuán diferentes son. Existen numerosos métodos en la literatura para medir la distancia entre histogramas (Pele and Werman, 2010; Marín-Reyes et al., 2016). Estos métodos se pueden dividir en dos categorías o grupos, que se distinguen principalmente por basarse en métricas de distancia distintas. La primera métrica es denominada *bin-to-bin*, y se calcula mediante la comparación de dos histogramas diferentes teniendo en cuenta solamente la diferencia entre los bines que se corresponden directamente. Así, en este tipo de distancias no se considera la influencia de bines vecinos. Por supuesto, esta medida requiere que los dominios de ambos histogramas coincidan, o lo que es lo mismo, que estén alineados (Pele and Werman, 2010). Esto a menudo no es el caso, debido a diversos factores como, por ejemplo, deformación en la forma del histograma y alto nivel de ruido (Ling and Okada, 2007). La segunda métrica se llama "distancia entre bines". A diferencia de la métrica anterior, sí tiene en cuenta los bines no correspondientes. Dentro del grupo de métodos que se agrupan bajo esta clasificación, Pele and Werman (2010) introdujo una métrica de distancia llamada χ-cuadrático (QC), que generaliza la llamada distancia "en forma cuadrática" (Hafner et al., 1995), y se calcula como:

$$\mathcal{D}_{QC}(H, G) = \sqrt{\max(\mathbf{B}\,\mathbf{A}\,\mathbf{B}^T, 0)}, \tag{2.8}$$

donde H y G son los histogramas que se desea comparar y $\mathbf{B}$ es una matriz que se calcula mediante la división elemento a elemento de las cantidades $P = (G - H)$ y $Q = [(H + G)\mathbf{A}]^q$, donde q es un factor de normalización en el rango $[0, 1)$. Por su parte, $\mathbf{A}$ es la matriz de "similitud de bines" (de aquí en adelante, matriz de similitud) y establece la relación entre bines vecinos.

En este trabajo utilizamos la ecuación 2.8 para medir la distancia entre histogramas. Para ello, seleccionamos la matriz de similitud $\mathbf{A} = \{a_{ij}\}$ propuesta por Jones et al. (2016), diseñada específicamente para comparar histogramas constituidos por atributos sísmicos:

$$a_{ij} = \begin{cases} \sqrt{\frac{3}{2\pi\tau}} \exp\left[-\frac{9(i-j)^2}{2\tau}\right] & l_c \leq i \leq r_c\,; \\ 0 & \text{c.c.}\,, \end{cases} \tag{2.9}$$

donde $\tau = N_b/10$, $l_c = \max[1, (i - \tau + 1)]$, $r_c = \min[N_b, (i + \tau - 1)]$, siendo N_b el numero de bines. Esta matriz de similitud representa una distribución Gaussiana con $\mu = i$ y $\sigma = \tau/3$. Cabe recordar que tanto el acimut como la inclinación son atributos de polarización de carácter periódico. Esta periodicidad debe ser tenida en cuenta a través de $\mathbf{A}$. Esto se logra relajando la primera restricción en la ecuación 2.9 a $\min(|i - j|, N_b - |i - j|) \leq 3\sigma$. Adicionalmente, se fija el factor de normalización $q = 0.5$, como sugieren en Jones et al. (2016).

2.3. Algoritmos de filtrado

En esta sección se describen brevemente los tres métodos que son utilizados para realizar la comparación (ver sección de ejemplos numéricos) analizando su impacto en la estimación de los atributos de polarización. Los métodos comparados son: (1) transformada hiperbólica de Radon en el espacio con modelo restringido (RHRT; Sabbione and Sacchi, 2016), (2) filtrado por reducción de rango basado en SVD (Velis et al., 2015), y (3) filtrado por descomposición en modos empíricos (EMD; Gómez and Velis, 2016). Los lectores pueden consultar los artículos que presentan estos algoritmos y las referencias en ellos para una completa y detallada descripción de los mismos.

Energía de la señal microsísmica

Sea un medio 2D con una velocidad efectiva v como aquella considerada en Yilmaz (2001) o Blias and Grechka (2013), donde se despliega un arreglo de receptores 3C en un pozo vertical. En este escenario, si consideramos un evento microsísmico en $\mathbf{x}_s = (x_s, y_s, z_s)$, la señal llegará a cada receptor ubicado en $\mathbf{x}_r = (x_r, y_r, z_r)$ con tiempos de arribo dados por:

$$t(\mathbf{x}) = t_0 + \frac{r}{v}\,, \tag{2.10}$$

donde t_0 es el tiempo en el cual ocurre la microfractura, y $r = |\mathbf{x}_r - \mathbf{x}_s|$ es la distancia fuente-receptor. La ecuación 2.10 nos muestra que los tiempos de arribo se alinean, en el dominio del dato, a lo largo de una hipérbola cuyo vértice está desplazado, siempre que los receptores hayan sido ubicados formando una línea recta (Velis et al., 2015).

En lo sucesivo, denotaremos al registro de datos 3D como $d_k(t, \mathbf{x}_r)$, con $k = x, y, z$. Los métodos de filtrado, RHRT y aquel basado en SVD, comienzan midiendo la coherencia de la señal o la energía media a lo largo de hipérbolas en el dominio del dato. Por este motivo, primero se define una nueva cantidad para promediar las tres componentes $k = x, y, z$ en el dominio del dato:

$$e(t, \mathbf{x}_r) = \frac{1}{3}\sum_k e_k(t, \mathbf{x}_r), \tag{2.11}$$

donde $e_k(t, \mathbf{x}_r)$ son atributos relacionados a la energía (p.ej. envolvente) obtenidos a partir de $d_k(t, \mathbf{x}_r)$.

Filtrado por RHRT

La ecuación 2.10 sugiere que cada componente del registro microsísmico $d_k(t, \mathbf{x}_r)$ puede sintetizarse mediante la suma de hipérbolas de vértice desplazado en el dominio de t-$\mathbf{x}_r$:

$$d_k(t, \mathbf{x}_r) = \sum_{\mathbf{x}_s, v} m_k \left(t_0 = t - \frac{r}{v}, \mathbf{x}_s, v \right), \tag{2.12}$$

donde $m_k(t_0, \mathbf{x}_s, v)$ son los coeficientes de la transformada de Radon para cada componente k del dato. La ecuación 2.12 es un operador "hacia adelante" (*forward*) que mapea el dominio de la transformada Radon hacia el dominio del dato. De la misma forma, el operador "adjunto", que mapea el dato hacia el dominio de la transformada de Radon está definido por:

$$m_k^{adj}(t_0, \mathbf{x}_s, v) = \sum_{\mathbf{x}_r} d_k \left(t = t_0 + \frac{r}{v}, \mathbf{x}_r \right). \tag{2.13}$$

Estas dos operaciones se pueden reescribir en forma matricial-vectorial como:

$$\mathbf{d}_k = \mathbf{L}\mathbf{m}_k \tag{2.14}$$

$$\mathbf{m}_k^{adj} = \mathbf{L}^T \mathbf{d}_k, \tag{2.15}$$

donde $\mathbf{L}$ representa al operador hacia adelante (ecuación 2.12) y $\mathbf{L}^T$ el operador adjunto (ecuación 2.13).

Los coeficientes de Radon $\mathbf{m}_k$ para cada componente del dato $\mathbf{d}_k$ se pueden estimar resolviendo un problema de inversión lineal (Thorson and Claerbout, 1985) que implica minimizar una función de costo definida como:

$$J = \|\mathbf{L}\mathbf{m}_k - \mathbf{d}_k\|_2^2 + \mu \|\mathbf{W}\mathbf{m}_k\|_2^2, \tag{2.16}$$

donde μ es un parámetro de *trade-off* y $\mathbf{W}$ es una matriz de peso cuyo propósito es restringir la solución.

El espacio del modelo $(t_0, \mathbf{x}_s, v)$ se discretiza definiendo intervalos de muestreo Δt_0, Δx_s, Δy_s, Δz_s y Δv. En RHRT, debido a que este espacio puede ser demasiado grande, el dominio de Radon es restringido antes de realizar la inversión. Con este fin, se calcula la envolvente media de las tres componentes utilizando ecuación 2.11. Las envolventes $e_k(t, \mathbf{x}_r)$ para cada componente k se calculan como un atributo instantáneo de los datos registrados $d_k(t, \mathbf{x}_r)$ y están definidos por el módulo de la señal analítica (Taner et al., 1979), que se obtiene mediante la transformada de Hilbert, tal como se hace en Michaud and Leaney (2008). Luego de calcular la envolvente media, se calculan los correspondientes coeficientes de Radon adjuntos como $\mathbf{m}_e^{adj} = \mathbf{L}^T \mathbf{e}$, definiendo así el siguiente conjunto en el espacio de Radon:

$$\mathbf{A} = \left\{ (t_0, \mathbf{x}_s, v) \ : \mid m_e^{adj}(t_0, \mathbf{x}_s, v) \mid > T_h \right\}, \tag{2.17}$$

donde T_h es una restricción definida por el usuario que determina el tamaño del dominio de Radon. Finalmente, la inversión se realiza sobre este mismo espacio restringido $\mathbf{A}$ para cada componente del dato, obteniendo:

$$\tilde{\mathbf{m}}_k^{\mathbf{A}} = \underset{\mathbf{m}_k^{\mathbf{A}}}{\operatorname{argmin}} \left[\|\mathbf{L}^{\mathbf{A}}\mathbf{m}_k^{\mathbf{A}} - \mathbf{d}_k\|_2^2 + \mu \|\mathbf{W}^{\mathbf{A}}\mathbf{m}_k^{\mathbf{A}}\|_2^2 \right]. \tag{2.18}$$

El poder de focalización mejora aun más si se define $\mathbf{W^A}=|(\mathbf{L^A})^T\mathbf{e}|^{-1}$. El problema de minimización se resuelve usando el método de gradientes conjugados (CG) (Hestenes and Stiefel, 1952). La restricción en el espacio del modelo reduce drásticamente el costo computacional y acelera la convergencia de la inversión. Asimismo, ayuda a mitigar los efectos del contenido de ruido en el dato. El producto final de esta técnica de inversión es una versión filtrada de las componentes del dato de entrada:

$$\tilde{\mathbf{d}}_k = \mathbf{L^A}\tilde{\mathbf{m}}_k^{\mathbf{A}}, \quad k = x, y, z. \tag{2.19}$$

Filtrado por reducción de rango basado en SVD

En este método, las fases de los primeros arribos se detectan buscando el conjunto de parámetros $(t_0, \mathbf{x}_s, v)$ que conducen a patrones hiperbólicos que maximicen la coherencia y/o energía. Para este fin, el método se basa en el atributo de energía dado por la ecuación 2.11, donde ahora $e_k(t, \mathbf{x}_r)$ se define como la envolvente de $d_k(t, \mathbf{x}_r)$. Por robustez, $e(t, \mathbf{x}_r)$ se promedia sobre una ventana de tiempo de longitud L_w predefinida y centrada en el tiempo dado por la ecuación 2.10. En la práctica, el algoritmo minimiza la función de costo

$$J(t_0, \mathbf{x}_s, v) = 1 - E(t_0, \mathbf{x}_s, v), \tag{2.20}$$

donde

$$E(t_0, \mathbf{x}_s, v) = \frac{1}{L_w}\sum_t \left[\frac{1}{N_r}\sum_{\mathbf{x}_r} e(t, \mathbf{x}_r)\right], \tag{2.21}$$

y N_r es el número de receptores. Para las expresiones 2.20 y 2.21, se asume que el dato fue normalizado previamente de manera que $0 \leq E, J \leq 1$. La minimización se lleva a cabo eficientemente mediante el algoritmo *very fast simulated annealing* (VFSA). Para que la búsqueda sea más eficiente, se restringe el rango de búsqueda tal cual se describe en Velis et al. (2015).

Una vez que las fases de los primeros arribos fueron detectadas, el dato "aplanado" u "horizontalizado" dentro de la ventana hiperbólica resultante para la componente k es visto como una matriz $\mathbf{S}_k$ de dimensión $L_w \times N_r$. Su descomposición en valores singulares (SVD) es:

$$\mathbf{S}_k = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T = \sum_{j=1}^{N_r} \sigma_j \mathbf{u}_j \mathbf{v}_j^T, \tag{2.22}$$

donde $\mathbf{U}$ es una matriz ortogonal de dimensión $L_w \times N_r$, cuyas columnas son los autovectores $\mathbf{u}_j$ de la matriz $\mathbf{S}_k\mathbf{S}_k^T$, $\mathbf{\Sigma}$ es una matriz diagonal de dimensión $N_r \times N_r$ con elementos positivos o nulos llamados "valores singulares", σ_j (ordenados por conveniencia de mayor a menor), y $\mathbf{V}$ es una matriz ortogonal de dimensión $N_r \times N_r$ cuyas columnas son los autovectores $\mathbf{v}_j$ de la matriz $\mathbf{S}_k^T\mathbf{S}_k$ (Press et al., 1992). Esta descomposición matricial es única y siempre existe, salvo por permutaciones de $\mathbf{U}$, $\mathbf{\Sigma}$ y $\mathbf{V}$.

La suma en la ecuación 2.22 permite ver el producto $\mathbf{u}_j\mathbf{v}_j^T$ como imágenes base de rango uno que a su vez forman una base ortogonal para la representación de imágenes ("autoimágenes"). Se puede demostrar que si uno considera la suma de n términos, con $n < N_r$, llamándose a esta suma $\tilde{\mathbf{S}}_k$, la norma Frobenius de la diferencia entre $\mathbf{S}_k$ y $\tilde{\mathbf{S}}_k$ se minimiza, y la norma de cada autoimagen es igual al valor singular correspondiente (Golub and Van Loan, 1989). Así, $\tilde{\mathbf{S}}_k$ es una matriz "aproximada" de rango n que permite capturar la mayor parte de la energía en unos pocos términos, especialmente cuando la señal comparte una misma ondícula (o similar) a lo largo de todos los canales. En efecto, para el caso de una señal limpia o libre de ruido, si todos los N_r arribos de la señal fuesen iguales, salvo por un factor de escala, luego $\sigma_j = 0$ para $j = 2, \cdots, N_r$, y $\mathbf{S}_k$ sería una matriz de rango uno. En la práctica, seleccionar $n = 1$ ó 2 permite una reducción considerable de los niveles de ruido presentes en la señal de entrada, preservando al mismo tiempo la forma de onda (Velis et al., 2015). El producto final de este método de filtrado basado en SVD es la versión filtrada de las componentes del dato de entrada:

$$\tilde{\mathbf{d}}_k = \mathcal{F}\{\tilde{\mathbf{S}}_k\}, \quad k = x, y, z, \tag{2.23}$$

donde $\mathcal{F}\{\cdot\}$ es el operador que deconstruye el aplanamiento.

Filtrado por EMD

La descomposición en modos empíricos (EMD, Huang et al. (1998)) descompone una señal dada $\mathbf{s}$ en un número finito de "funciones de modo intrínseco" (IMF, por sus siglas en inglés), en donde se encuentran contenidas las componentes oscilatorias de la señal. Cada IMF, a saber $\mathbf{f}_j$, $j = 1, \ldots, n_f$, se obtiene iterando en un proceso de dos pasos. Primero, se calcula la envolvente media de la señal como:

$$\overline{\mathbf{e}} = \frac{\mathbf{e}^+ + \mathbf{e}^-}{2}, \tag{2.24}$$

donde $\mathbf{e}^+$ y $\mathbf{e}^-$ representan las envolventes superior e inferior de la señal, respectivamente. En segundo lugar, se calcula un residuo $\mathbf{r} = \mathbf{s} - \overline{\mathbf{e}}$ que se considera como la nueva señal de entrada para la siguiente iteración. Este proceso de iteración, conocido como "tamizado", continúa hasta satisfacer algún criterio de terminación predeterminado, o bien cuando se alcanza un número máximo de iteraciones permitidas por el usuario (Rilling et al., 2003; Huang and Wu, 2008; Wu and Huang, 2009). En este punto, se considera que el residuo final es el primer IMF, $\mathbf{f}_1$. Para obtener los IMFs de orden superior, uno simplemente sustrae $\mathbf{f}_1$ de la señal original y repite el proceso de tamizado. Por lo tanto, todos los IMFs se calculan de forma recursiva y la descomposición termina cuando el último IMF se convierte en una función monótona de la cual no es posible extraer más IMFs (Huang et al., 1998).

Los IMFs satisfacen dos condiciones principales: (1) el número de extremos y cruces por cero deben ser iguales o diferir como máximo en uno, y (2) en cualquier punto de los datos,

el valor medio entre las envolventes superior e inferior es cero. En términos del contenido en frecuencia, $\mathbf{f}_1$ es la componente más oscilante de la señal. A medida que el orden de los modos aumenta, su contenido en frecuencia disminuye. Por lo tanto, la eliminación de los primeros IMFs de la señal de entrada actúa como un filtro pasabajos guiado por los datos y, en consecuencia, el método se puede utilizar como una herramienta de *denoising*.

El principio básico del EMD o cualquiera de sus variaciones (Torres et al., 2011; Colominas et al., 2014), es reconstruir una señal dada $\mathbf{s}$ como:

$$\mathbf{s} = \sum_{j=1}^{n_f} \mathbf{f}_j + \mathbf{r}_f, \tag{2.25}$$

donde $\mathbf{r}_f$ es el residuo final. La ecuación 2.25 demuestra se puede obtener una señal filtrada $\tilde{\mathbf{s}}$ a partir de la remoción del modo más oscilante:

$$\tilde{\mathbf{s}} = \mathbf{s} - \mathbf{f}_1. \tag{2.26}$$

Este filtrado se focaliza en el ruido de alta frecuencia y es, en general, suficiente para la mayoría de las aplicaciones sísmicas.

En este capítulo se aplica la variante 1D del método EMD presentado en Gómez et al. (2020), donde se presenta una ligera modificación al algoritmo propuesto en He et al. (2017). Este algoritmo evita el costo computacional inherente a la interpolación mediante *splines* cúbicas para calcular la envolvente media, y por ende, es más eficiente, en términos de rapidez, que los algoritmos tradicionales de EMD, logrando resultados similares. El producto final de este filtrado por EMD son las versiones filtradas de las componentes del dato de entrada,

$$\tilde{\mathbf{d}}_k = (\tilde{\mathbf{s}}_1, \cdots, \tilde{\mathbf{s}}_{N_r}), \quad k = x, y, z, \tag{2.27}$$

donde $\tilde{\mathbf{s}}_i$ es el i-ésimo canal de la componente k.

2.4. Ejemplos numéricos

Primero vamos a realizar la comparación de algoritmos de filtrado mediante histogramas de polarización utilizando datos sintéticos para evaluar si es capaz de identificar correctamente los atributos de polarización de una dada señal microsísmica. Para ello, realizamos un análisis estadístico utilizando datos con varios niveles de ruido y un gran número de realizaciones (experimentos) y comparamos el desempeño de los cuatro métodos de *denoising* antes mencionados: BPF, RHRT, SVD y EMD. Los resultados de BPF sirven como referencia.

Los datos sintéticos se generan utilizando la aproximación de campo lejano (Shearer, 1999b):

$$\mathbf{d}_k = \frac{1}{4\pi\rho v^3 r}\mathbf{R}_{klm}\dot{\mathbf{M}}_{lm}(t - \frac{r}{v}), \tag{2.28}$$

que define la componente k-ésima del desplazamiento $\mathbf{d}$ como una función del tiempo observado en un receptor ubicado en la posición $\mathbf{x}_r = (x_r, y_r, z_r)$, y que fue generado por una fuente a una distancia r del receptor. En la ecuación 2.28, ρ es la densidad del medio, v la velocidad efectiva de la onda P o S, y $\mathbf{R}_{klm}$ es el tensor del patrón de radiación correspondiente a la componente k-ésima del receptor y el elemento lm-ésimo del tensor momento sísmico $\mathbf{M}_{lm}$. Siguiendo a Aki and Richards (2002), si se asume que $\mathbf{M}_{lm}$ puede ser expresado como el producto entre un tensor momento invariante en el tiempo y una ondícula que varía en el tiempo $w(t)$, se puede escribir:

$$\mathbf{M}_{lm}(t) = \mathbf{M}_{lm} w(t). \tag{2.29}$$

Luego,

$$\dot{\mathbf{M}}_{lm}(t) = \mathbf{M}_{lm} \frac{dw(t)}{dt}, \tag{2.30}$$

y

$$\mathbf{d}_k = \frac{1}{4\pi\rho v^3 r} \mathbf{R}_{klm} \mathbf{M}_{lm} \frac{dw}{dt}\left(t - \frac{r}{v}\right). \tag{2.31}$$

Usamos la ecuación 2.31 para generar datos sintéticos que contengan ondas de tipo P y S. Para el análisis, sin embargo, solo se estiman los atributos de polarización de la onda P.

Datos sintéticos

Por simplicidad, para cada uno de los ejemplos se considera un medio homogéneo con densidad $\rho = 1$ g/cm^3, velocidad compresional $v_p = 3500$ m/s y de cizalla $v_s = 2400$ m/s. Colocamos la fuente en $(x_s, y_s, z_s) = (240, 320, 353.5)$ m, y utilizamos una ondícula de Ricker $w(t)$ con frecuencia $f_0 = 100$ Hz e intervalo de muestreo $\Delta t = 1$ ms. Los receptores se colocan en $(0, 0, z_j)$, con $z_j = (j-1)\Delta z$, $j = 1, \ldots, N_r$, $\Delta z = 30.5$ m y $N_r = 8$. Para el mecanismo de fractura, consideramos un tensor momento

$$\mathbf{M} = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}. \tag{2.32}$$

El dato microsísmico resultante se muestra en la Figura 2.1 (primera columna), donde se aprecian las tres componentes. Vale la pena mencionar que, para evitar divisiones por cero en el cálculo de los atributos de polarización, se agrega ruido gaussiano de banda limitada y muy baja amplitud, de modo que la relación S/R sea de 50 dB. Para los experimentos, consideraremos este dato como el dato "limpio".

A continuación, se generan datos ruidosos contaminando los datos "limpios" con ruido de campo extraído aleatoriamente de registros microsísmicos reales. El ruido real es escalado para obtener registros con relaciones S/R que van desde -10 a 30 dB con un incremento

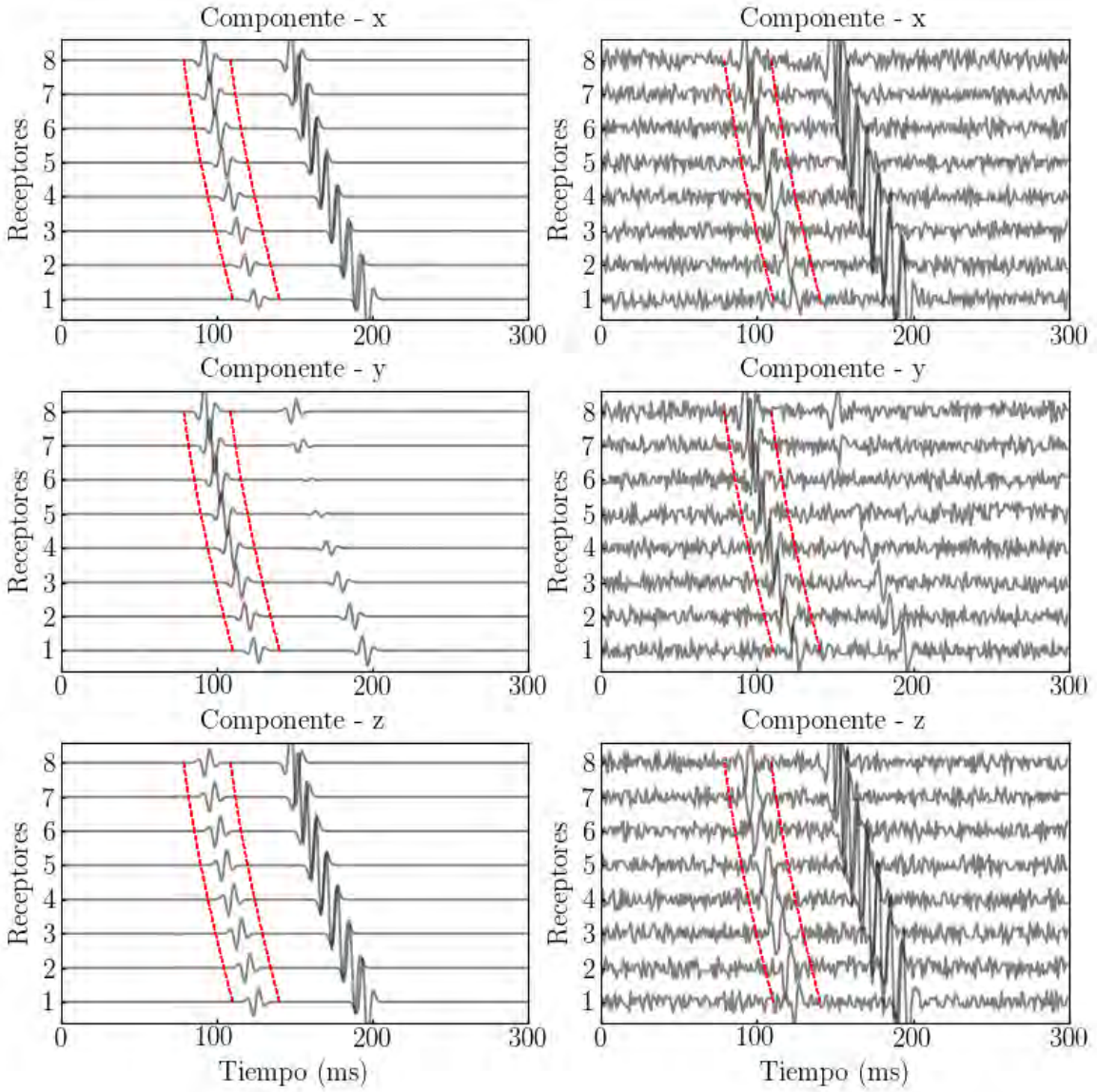


Figura 2.1: Dato sintético limpio (se muestran los arribos de las fases P y S) con S/R = 50 dB (primera columna). Dato sintético contaminado con ruido real de campo tal que S/R= -4 dB (segunda columna). Las líneas punteadas denotan la ventana hiperbólica de 30 ms a partir de la cual se aíslan los arribos de las señales microsísmicas. La señal se desplazó -37 ms por razones estéticas.

Método	Parámetro	Valor
BPF	f_1, f_2, f_3, f_4	0, 1, 300, 350 (Hz)
RHRT	$\Delta\tilde{x}_s$, Δz_s	20 m
	Δv	100 m/s
	μ	1.0
	T_h	3 %
SVD	L_w	31
	n	2
EMD	n_f	1
	# de tamizados	10

Tabla 2.1: Parámetros utilizados por cada método de filtrado. En RHRT, $T_h = 3\,\%$ hace referencia a que el criterio de corte se seleccionó de forma de utilizar únicamente el 3 % del total del dominio de Radon.

de 1 dB. Para fines estadísticos, se generan 100 realizaciones por cada nivel de ruido, obteniendo así un total de 4100 conjuntos de datos reunidos en 41 grupos con diferentes relaciones S/R. Como ejemplo, la segunda columna de la Figura 2.1 muestra un ejemplo de un conjunto de datos ruidosos con relación S/R= -4 dB.

Filtrado de ruido

Además de los tres métodos de *denoising* no convencionales descritos en la sección anterior (RHRT, SVD y EMD), utilizamos un filtro pasabanda convencional (BPF). En particular, se aplica un filtro tipo *Ormsby* en el dominio de la frecuencia (en realidad usamos la función *SeisBandPass* de SeismicJulia) (Stanton and Sacchi, 2016), con frecuencias de corte seleccionadas específicamente para no dañar la energía de la señal. En cuanto a los parámetros de RHRT, SVD, y EMD, se eligieron siguiendo las recomendaciones dadas por los autores correspondientes (Sabbione et al. (2015), Velis et al. (2015) y Gómez and Velis (2016), respectivamente), y se ajustaron realizando algunos ensayos de prueba y error. La Tabla 2.1 resume los principales parámetros seleccionados para cada método de filtrado. Se debe tener en cuenta que para el método RHRT, es necesario fusionar ambas coordenadas horizontales en una: $\tilde{x} = \sqrt{x^2 + y^2}$, por lo que solo es necesario definir $\Delta\tilde{x}_s$ en lugar de Δx_s y Δy_s.

En la Figura 2.2 se muestra el resultado de filtrar un dato sintético cualquiera dentro de nuestro set de datos, y que contiene dato ruidoso correspondiente a la señal P aislada y contaminada con ruido real con S/R = -4 dB. Observando los residuos de la Figura 2.2, es claro que todos los métodos aplicados están actuando para remover el ruido sin dañar la señal.

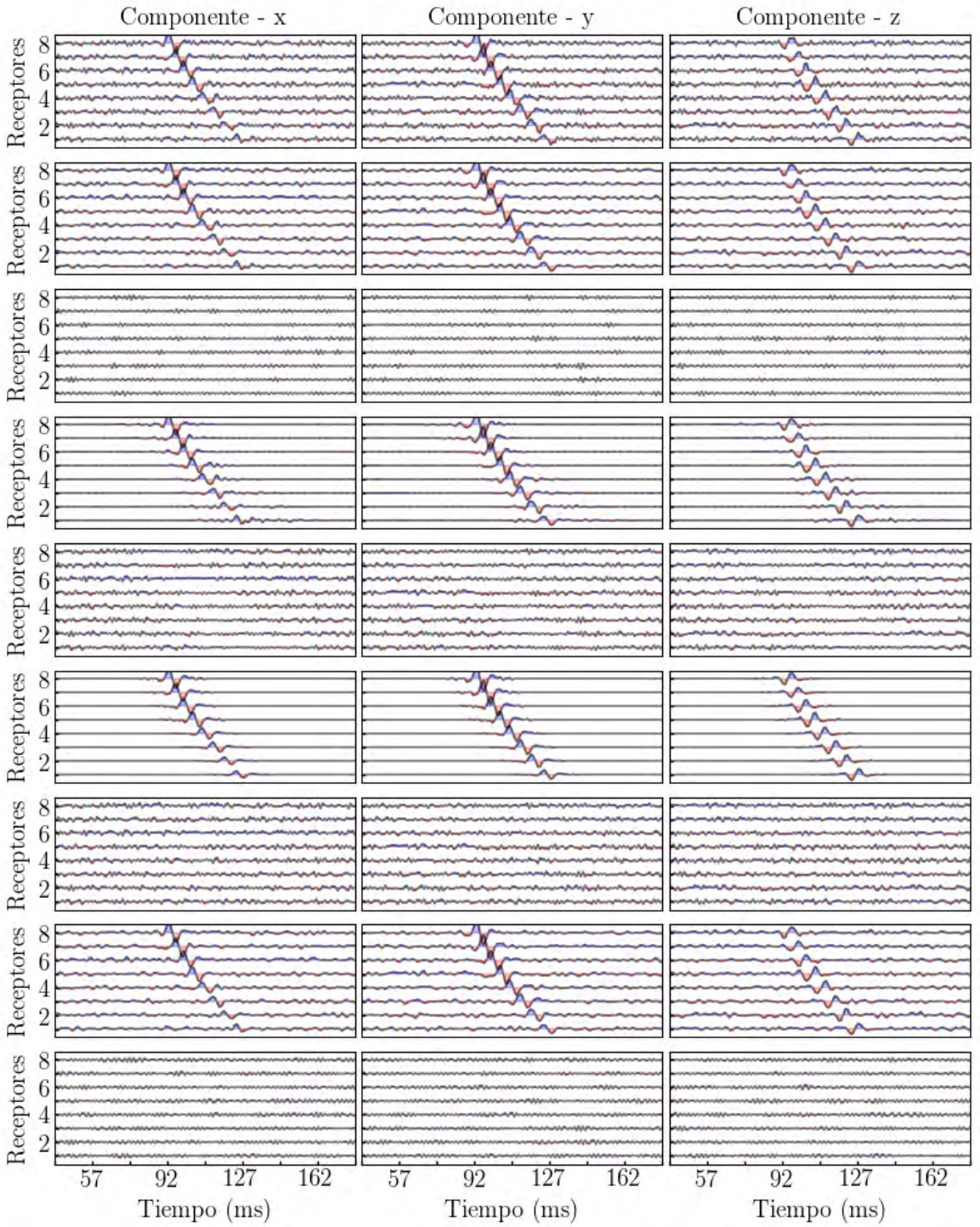


Figura 2.2: Ejemplo de filtrado del dato sintético. Fila 1: Dato con ruido real tal que S/R = −4 dB. Filas 2 a 9: Dato filtrado (y residuos correspondiente) utilizando los métodos de BPF, RHRT, SVD y EMD, respectivamente.

Histogramas de atributos de polarización

Una vez realizado el filtrado de los datos, se calculan los parámetros de polarización $\rho(t)$, $\theta(t)$, y $\phi(t)$ para cada conjunto de datos (limpio y filtrado) y se construyen los histogramas correspondientes. Sin pérdida de generalidad, realizamos el análisis para el canal 5. Para enfocar el análisis solo en la forma de onda de la señal que fue filtrada, primero se detecta y aísla el arribo de la onda P como se explicó al comienzo de la Sección 2.3 usando una ventana de tiempo de 30 ms ($L_w = 31$). La ventana óptima después de la minimización de la ecuación 2.20 se indica con las líneas punteadas en la Figura 2.1. La matriz de covarianza $\mathbf{C}(t)$ usada para calcular los parámetros de polarización y los histogramas correspondientes se calcula dentro de esta ventana, usando una ventana móvil de 4 ms ($L_c = 5$ en la ecuación 2.2). En cuanto a los histogramas, el número de bines se establece en $N_b = 51$.

Los atributos de polarización de los datos sintéticos limpios se derivan fácilmente de las ecuaciones 2.4 - 2.6. Debido a que solo se elimina el ruido de las ondas compresionales, se espera obtener un único autovalor distinto de cero. Para esta geometría esperamos entonces un valor de rectilinealidad $\rho \simeq 1$. En cuanto al acimut, teniendo en cuenta que $(x_r, y_r) = (0, 0)$ y $(x_s, y_s) = (240, 320)$ m, obtenemos $\theta = \tan^{-1}(320/240) \simeq 53°$. Finalmente, dado que la fuente está a 400 m de distancia del arreglo de receptores verticales, y la distancia vertical entre la fuente y el receptor 5 es 262 m, la inclinación se estima en $\phi = \tan^{-1}(262/400) \simeq 33°$. La Figura 2.3 (primera fila) muestra el promedio de los histogramas construidos a partir de 100 realizaciones del dato limpio para el receptor 5. Los resultados obtenidos concuerdan, como es esperable, con los resultados teóricos calculados anteriormente. La Figura 2.3 (segunda fila) muestra, además, los histogramas promedio calculados con el dato ruidoso para cada uno de los niveles de S/R considerados (promedio de 100 realizaciones para cada S/R). Como es esperable, los histogramas que corresponden a los datos de mayor relación S/R (aquellos en la región inferior de cada panel) exhiben un máximo pronunciado alrededor del valor teórico (en términos relativos). Por otro lado, los histogramas correspondientes a señales de baja relación S/R (las curva en la región superior de cada panel) son más planas y encuentran su máximo de forma menos pronunciada.

Los histogramas resultantes luego del filtrado se muestran en la Figura 2.3 (filas 3 a 6). En función de lo que muestra esta figura, se puede ver que la ventaja de filtrar el dato es evidente: para todos los casos, los histogramas del dato filtrado se concentran más sobre los valores reales que aquellos calculados con el dato ruidoso. En particular, notamos que el método basado en SVD es el que conduce a las estimaciones más precisas.

Evaluación de métodos de filtrado

Para evaluar el impacto de cada método de *denoising* sobre la estimación de los atributos de polarización, se realizan dos experimentos diferentes. Primero, se comparan los

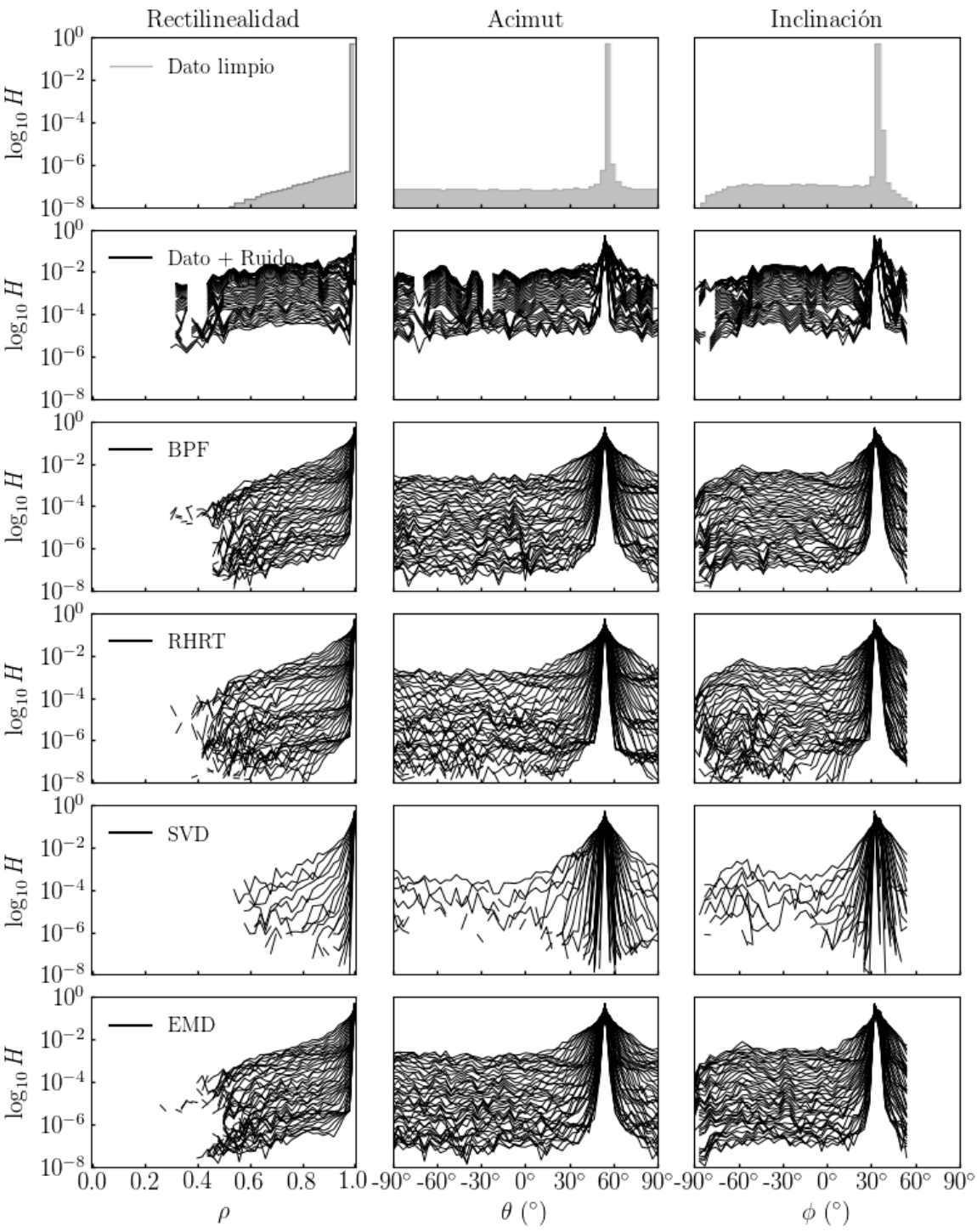


Figura 2.3: Histogramas promedio de la rectilinealidad (columna 1), acimut (columna 2) e inclinación (columna 3). Cada curva corresponde al promedio de histogramas luego de 100 realizaciones para cada relación S/R considerada. Filas 1 y 2: Dato limpio y contaminado con ruido, respectivamente. Filas 3 a 6: histogramas luego del filtrado usando BPF, RHRT, SVD y EMD, respectivamente.

histogramas correspondientes a las señales limpias y filtradas y, en segundo lugar, entre los histogramas de las señales con ruido y filtradas.

La Figura 2.4 muestra la distancia de histogramas para cada nivel de S/R y realización calculadas mediante la ecuación 2.8, con H y G representando los histogramas del dato limpio y filtrado, respectivamente.

Para el primer experimento, esperamos que distancias más pequeñas indiquen una mejor estimación del atributo de polarización. La última fila de la Figura 2.4 muestra las distancias medias correspondientes para cada caso. Como era de esperar, las distancias disminuyen al aumentar la relación S/R en todos los casos. La característica más notable que se puede observar en la Figura 2.4 es que las distancias correspondientes al método SVD son las más pequeñas para la mayoría de los niveles de ruido, lo que sugiere una mejor estimación de los parámetros de polarización y un mejor rendimiento en el *denoising*. Estas observaciones son más notorias para la rectilinealidad y, en menor medida, para el acimut. Sin embargo, con una relación S/R muy grande, el método basado en SVD muestra un rendimiento similar a los otros métodos. Por otro lado, RHRT, BPF y EMD, muestran un comportamiento similar, aunque el rendimiento de RHRT a niveles relativamente bajos de S/R son ligeramente mejores que aquel mostrado por BPF y EMD.

Para el segundo experimento, se calculan las distancias entre los histogramas de los datos filtrados y los correspondientes al conjunto de datos ruidosos. Parece razonable afirmar que el método con el mejor rendimiento presentaría la mayor distancia, siempre que ninguno de los métodos realice un sobre-filtrado del ruido afectando la señal. En otras palabras, cuanto mayor sea la mejora en la estimación del atributo de polarización, mayor será la distancia del histograma a la entrada de datos ruidosos. En la Figura 2.5 se muestran todas estas curvas de distancias, ahora calculadas a partir de la polarización del dato ruidoso (en lugar de los datos limpios). La última fila de la Figura 2.5, que muestra las distancias medias del histograma para cada método, resume los resultados de este segundo experimento. De acuerdo con el análisis de la Figura 2.4, los resultados que se muestran en la Figura 2.5 sugieren que el método SVD conduce a mejores estimaciones de atributos, especialmente para los casos de S/R más bajos.

Control de calidad

Para realizar un control de calidad y en orden de validar el análisis basado en las distancias entre histogramas, se calcula también el factor de calidad QF (*quality factor*) del dato filtrado para todos los niveles de ruido y para cada uno de los métodos (promedio luego de 100 realizaciones). El factor de calidad, en decibeles, puede ser calculado como (Chen and Sacchi, 2014):

$$\mathrm{QF}_{dB} = 20 \log_{10} \left(\frac{\|\mathbf{s}\|_F}{\|\mathbf{s} - \tilde{\mathbf{s}}\|_F} \right), \tag{2.33}$$

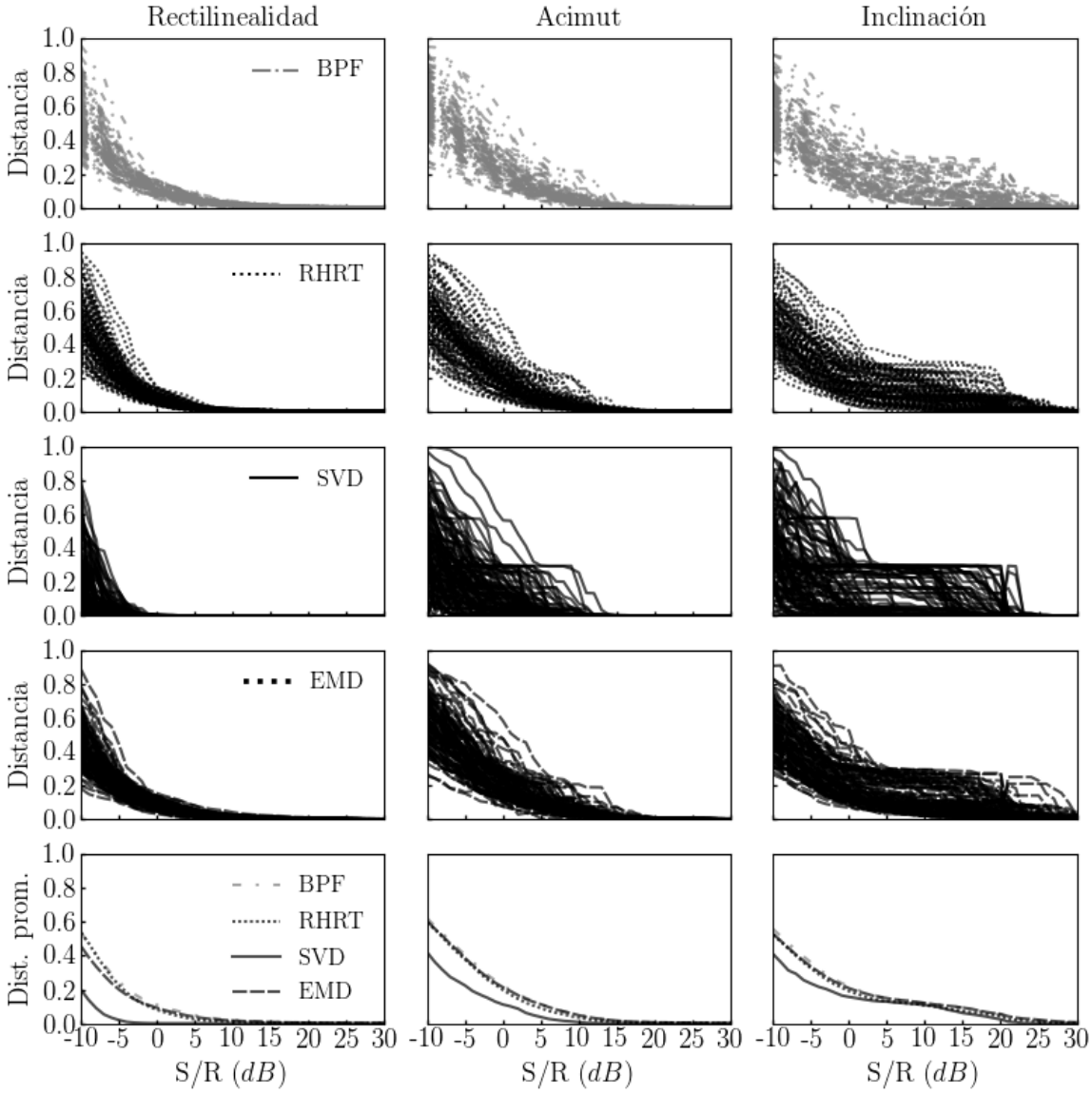


Figura 2.4: Distancia entre histogramas calculados con datos limpios y filtrados para la rectilinealidad (columna 1), acimut (columna 2) e inclinación (columna 3). Filas 1 a 4: distancia entre histogramas luego del filtrado utilizando BPF, RHRT, SVD y EMD, respectivamente (cada curva corresponde a una realización diferente). Fila 5: distancia de histogramas promedio para las 100 realizaciones.

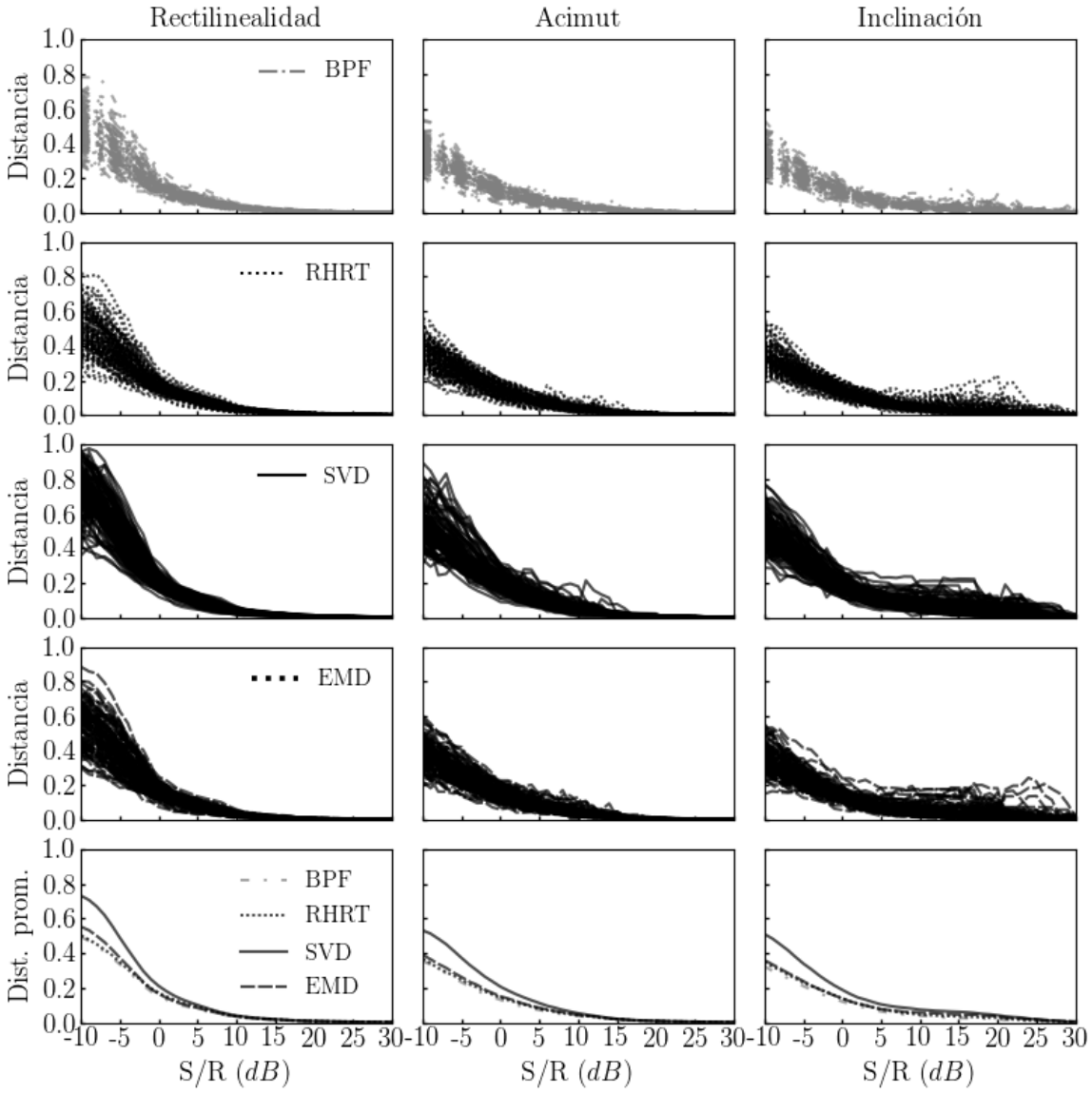


Figura 2.5: Distancia entre histogramas del dato ruidoso y filtrado para la rectilinealidad (columna 1), acimut (columna 2) e inclinación (columna 3). Filas 1 a 4: distancia entre histogramas luego del filtrado utilizando BPF, RHRT, SVD y EMD, respectivamente (cada curva corresponde a una realización diferente). Fila 5: distancia de histograma promedio luego de 100 realizaciones.

donde $\mathbf{s}$ y $\tilde{\mathbf{s}}$ son el dato limpio y filtrado, respectivamente, y $||\cdot||_F$ denota la norma Frobenius. Claramente, cuanto más $\tilde{\mathbf{s}}$ se asemeje a $\mathbf{s}$, mayor será el valor de QF.

La Figura 2.6 muestra el QF promedio después de 100 realizaciones para cada nivel de S/R de la señal 3C $\mathbf{s} = [\mathbf{s}_x, \mathbf{s}_y, \mathbf{s}_z]$, así como el QF para los componentes individuales $\mathbf{s}_k$, $k = x, y, z$. La línea roja en la Figura 2.6 corresponde a los valores QF de referencia calculados utilizando los datos ruidosos de entrada antes de eliminar el ruido. Esta línea recta representa una función de identidad que se encuentra desplazada en el eje vertical. Esto último es así porque si bien la relación S/R se midió para todo el conjunto de datos, el cálculo de QF se llevó a cabo dentro de la ventana de 30 ms que contiene el arribo del evento de la señal. Para todos los métodos, QF muestra una mejora, como es esperable, excepto para los valores más grandes de S/R. Debido a la complejidad de los métodos (a nivel de sus cálculos), los rendimientos de SVD, RHRT y EMD tienden a disminuir para grandes niveles de S/R (20 y superior). En estos escenarios, la calidad de la señal es muy alta (QF > 30), y eliminar el ruido puede ser contraproducente. Los resultados muestran que el método SVD produce los mayores valores de QF para casi todos los niveles de ruido considerados, seguidos de RHRT, EMD y BPF. Esto está totalmente de acuerdo con el análisis de distancias del histograma.

2.5. Ejemplo con dato de campo

Probamos los algoritmos con un registro de campo 3C, que fue previamente estudiado en Sabbione et al. (2015) y Velis et al. (2015). La Figura 2.7 (primera fila) muestra una parte de los datos de campo con un único arribo de una fase microsísmica e inmersa en ruido, mientras que los datos filtrados con BPF, RHRT, SVD y EMD se muestran en las filas 2 a 5 de la misma figura. Para el procesamiento de este dato, se utilizan los mismos parámetros que en el ejemplo sintético (Tabla 2.1). Al momento de realizar el filtrado, el canal 5 fue silenciado e ignorado por estar dañado. En general, los cuatro métodos mejoraron considerablemente la relación S/R, especialmente para la componente z. Por otra parte, puede observarse que las componentes x e y están severamente contaminadas por ruido y tanto el BPF como el EMD no realizan un filtrado satisfactorio. En contraste, la relación S/R más alta de la componente z favorece tanto a RHRT como a SVD, que se basan en un atributo de envolvente que combina las tres componentes del dato. Vale la pena señalar que el método basado en SVD realiza su aproximación por reducción de rango únicamente dentro de una ventana de forma hiperbólica de longitud L_w que contiene la señal. Por el contrario, los otros métodos son capaces de eliminar el ruido trabajando en el total del registro de tiempo de los datos. No obstante, si uno solo se enfoca en la señal microsísmica, el SVD parece recuperar las formas de onda de mejor manera, devolviendo señales más limpias. La Figura 2.8 muestra los histogramas de polarización correspondiente

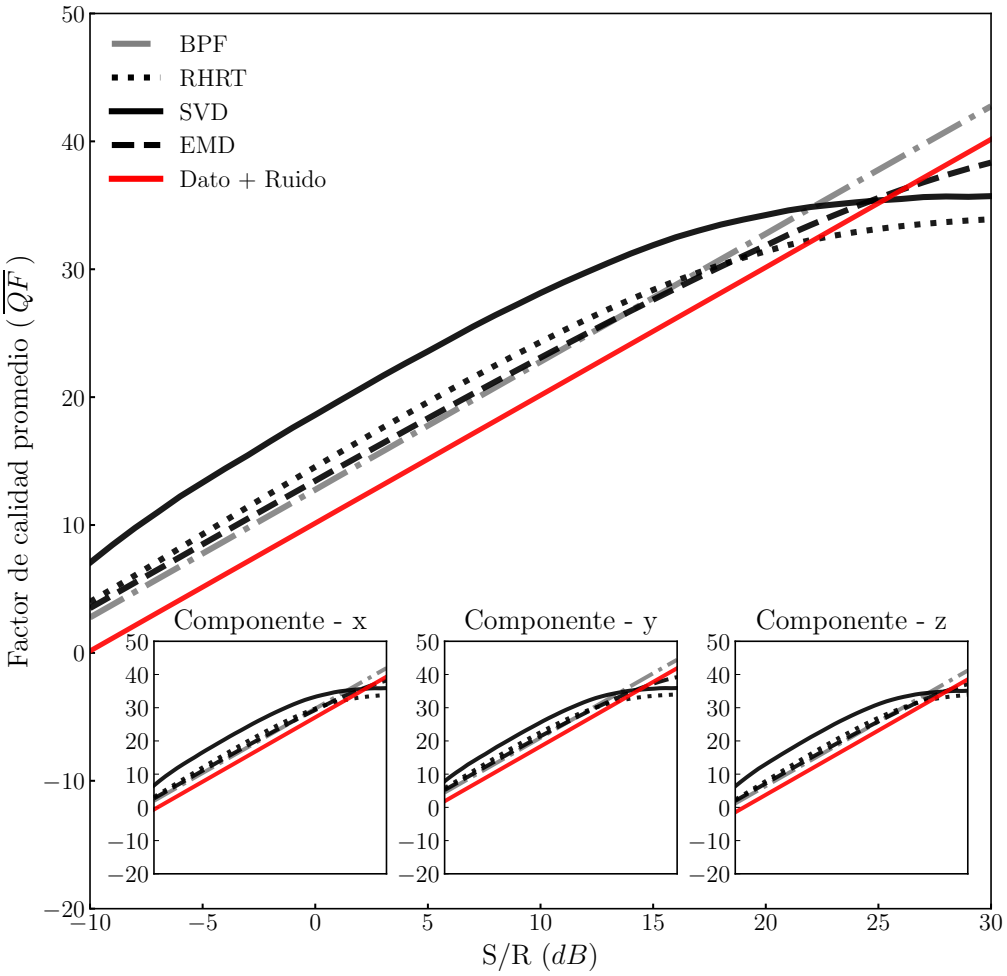


Figura 2.6: Factor de calidad promedio como función de la relación S/R (luego de 100 realizaciones) para cada método de filtrado. La línea roja denota el factor de calidad del dato ruidoso.

a dos canales diferentes para los datos crudos (primera fila) y los datos filtrados (de la segunda a la quinta fila). Debido a la mala calidad del dato en las componentes x e y, los métodos de filtrado no tienen un impacto muy visible en los histogramas, sin máximos claros, excepto en el caso de la SVD donde los histogramas se observan más estrechos. Dado que no hay datos limpios para disponer de una referencia, no es posible calcular distancias entre datos limpios y filtrados para evaluar la exactitud de las estimaciones.

Sin embargo, sí es posible calcular las distancias de polarización entre los datos sin procesar y los datos filtrados, como se hizo para el ejemplo sintéticos de la Figura 2.5. Estas distancias se indican dentro de cada panel en la Figura 2.8 y Tabla 2.2. Como se explica en el caso sintético, cuanto mayor sea la distancia, se interpreta que mejor fue el *denoising*, siempre que el filtrado no dañe la señal. Se observa que, en concordancia con los resultados obtenidos para los ejemplos sintéticos, el método basado en SVD supera a los demás, logrando las mayores distancias en la mayoría de los casos. Tanto RHRT como EMD muestran comportamientos similares, mientras que BPF, como se esperaba, exhibe la menor distancia para ambos canales.

Método	Canal	Rectilinealidad	Acimut	Inclinación
BPF	1	0.13	0.17	0.12
	8	0.24	0.28	0.20
RHRT	1	0.18	0.21	0.17
	8	0.31	0.32	0.28
SVD	1	0.38	0.61	0.31
	8	0.46	0.50	0.33
EMD	1	0.21	0.30	0.13
	8	0.36	0.29	0.26

Tabla 2.2: Distancia de histogramas entre dato crudo de campo y filtrado para los canales 1 y 8.

2.6. Discusión

Nuestro principal objetivo es presentar una estrategia que permita decidir qué método de filtrado de ruido recupera mejores resultados que un BPF tradicional. Sin embargo, hay algunos comentarios que vale la pena mencionar con respecto a los métodos que se analizan. El hecho de que BPF y EMD traten los datos como series de tiempo y no dependan de ninguna propiedad geofísica permite utilizarlos para procesar todo tipo de datos microsísmicos sin ninguna modificación. Esto puede ser visto como una ventaja sobre los otros métodos. Sin embargo, dado que no fueron diseñados particularmente para tratar con

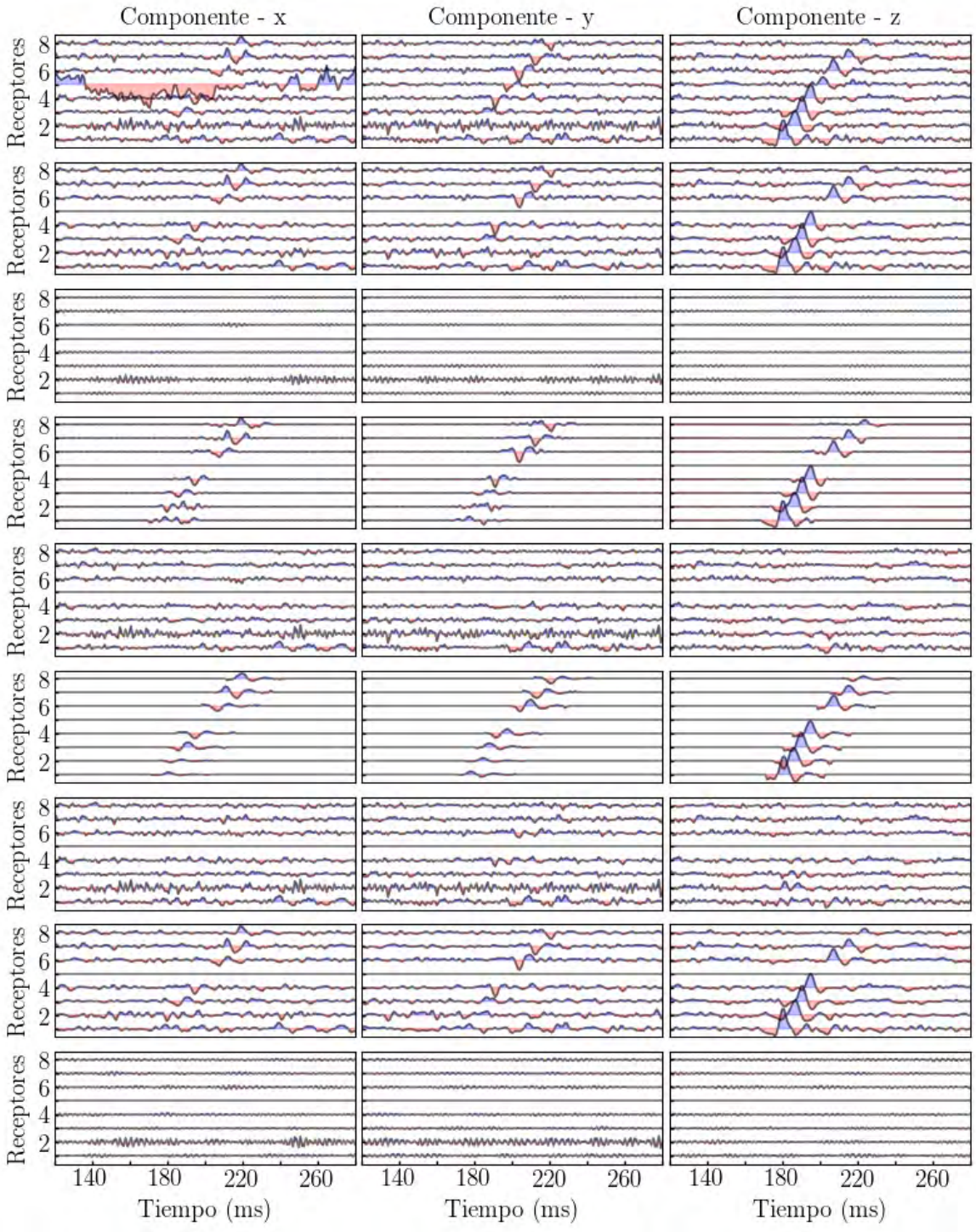


Figura 2.7: Ejemplo con datos de campo. Fila 1: dato con ruido. Filas 2 a 9: dato filtrado (y residuos correspondientes) utilizando BPF, RHRT, SVD y EMD, respectivamente. El canal 5 (corrupto) fue "muteado" antes de realizar el filtrado.

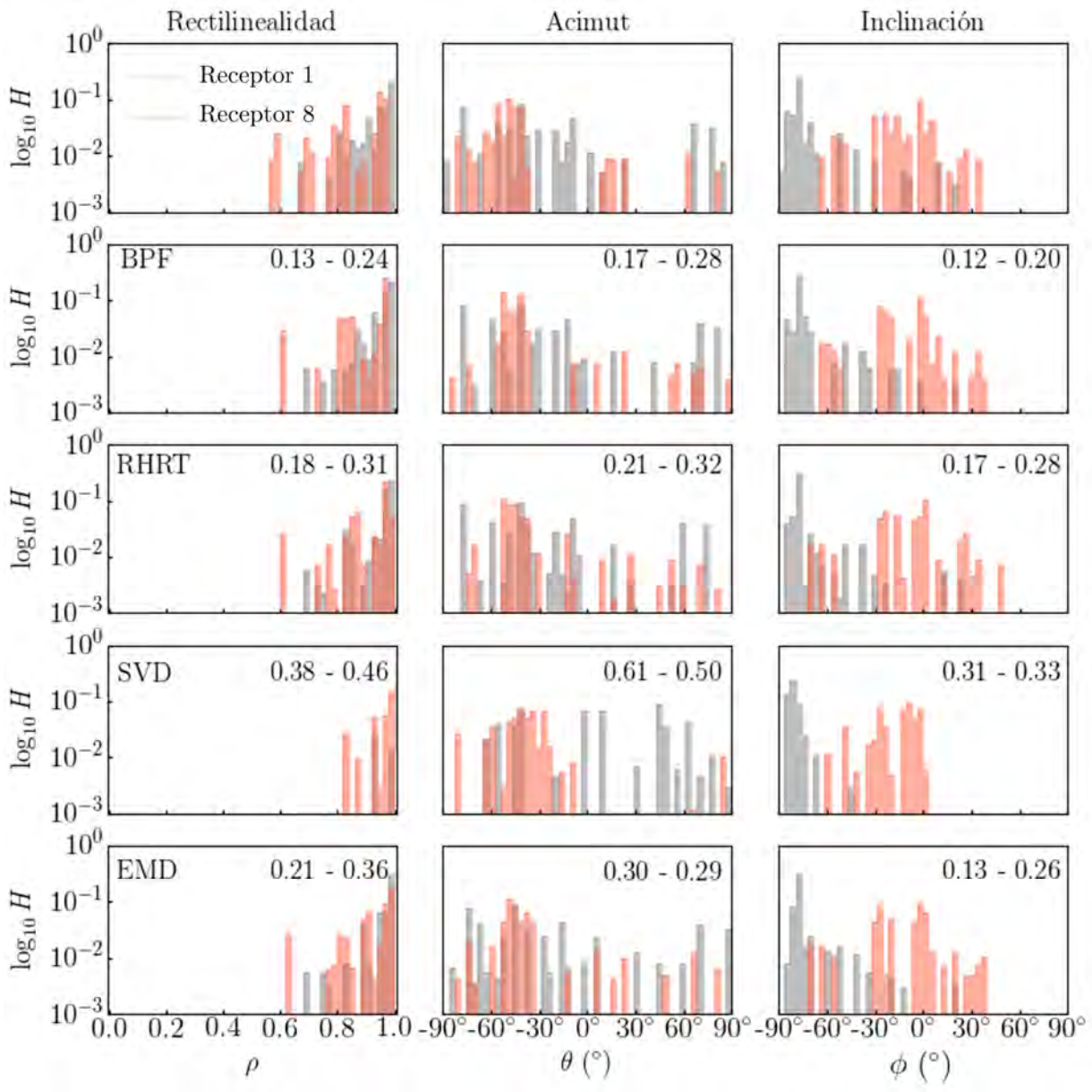


Figura 2.8: Ejemplo con datos de campo: histogramas de rectilinealidad (columna 1), acimut (columna 2) e inclinación (columna 3) para los canales 1 (gris) y 8 (naranja). Filas 1 a 5: resultados correspondientes al dato crudo y luego del filtrado utilizando BPF, RHRT, SVD y EMD, respectivamente. Los valores medidos para las distancias entre los histogramas del dato crudo y filtrado para cada canal se encuentran indicadas en la esquina superior derecha de cada panel.

datos microsísmicos, estos métodos no explotan la alineación hiperbólica esperada para la energía de la señal, entre otras características. Por el contrario, tanto RHRT como SVD se valen de la geometría de un modelo y así adaptan su desempeño según el dato. Esto, que puede verse como una ventaja, también supone un costo de procesamiento extra, ya que para poder adaptarse a cada geometría, el usuario debe configurar los parámetros manualmente y de forma correcta. Dicho esto, es posible aplicar estos dos métodos en otros entornos de adquisición, como arreglos de superficie o pozos inclinados, donde podrían ser necesarias algunas modificaciones a los algoritmos originales.

En cualquier comparación o evaluación para este tipo de métodos de filtrado de ruido, el ajuste y la configuración de los parámetros involucrados en su funcionamiento juegan un papel muy importante en el rendimiento alcanzado. Entre los algoritmos probados, BPF y EMD son los más simples, ya que solo se basan en unos pocos parámetros fáciles de seleccionar. Por otro lado, RHRT y SVD requieren algunos ajustes de los parámetros para lograr un rendimiento óptimo. No obstante, creemos que en la selección de estos parámetros no afectó los resultados ya que contamos con experiencia previa en el uso de ambos métodos. No es el objetivo de este trabajo discutir los detalles de cada método de filtrado a fondo, sino más bien, enfocarnos en el diseño de una estrategia que nos permita comparar sus resultados.

También es interesante observar que para relaciones S/R muy altas, los métodos más simples, como BPF y EMD, superan a los más sofisticados, RHRT y SVD. Este resultado es esperable, ya que para una relación S/R muy grande, el ruido es insignificante, y aquellos métodos que involucran cálculos más complejos pueden tender a afectar la señal más que el ruido en sí. En otras palabras, se puede argumentar que en estos escenarios, con una relación S/R grande, no existe una necesidad real de eliminación de ruido. Aun más, en esta situación, exponer el dato a un método de filtrado puede resultar contraproducente.

La evaluación que proponemos para la comparación de estas estrategias se basa en medir las distancias de los histogramas de ciertos atributos. Por lo tanto, los atributos utilizados para generar el histograma deben cumplir con algunos requisitos: representar alguna característica de los datos a analizar, ser sensibles al contenido de ruido y abarcar diferentes valores que permitan ordenarlos en un histograma. Para los datos microsísmicos, los atributos de polarización satisfacen claramente todas estas condiciones. Para adaptar esta estrategia a escenarios con diferentes datos y/o atributos, se pueden utilizar diferentes tamaños de bines para generar los histogramas. Es decir, que el ancho del bin puede variar de manera que sus diferencias sean valores distinguibles al medir la distancia.

Queremos enfatizar que el principal objetivo y motivación de método que se presenta en este capítulo no es decidir cuál es el mejor método de filtrado disponible, sino proponer una estrategia que permita evaluar cuantitativamente el desempeño de diferentes métodos a la hora de manejar datos reales. La mayoría de los métodos de filtrado se evalúan cualita-

tivamente mediante la inspección visual de los datos filtrados y/o analizando los residuos, prestando especial atención a no dañar la señal de interés. La razón principal de esto es fundamental en el procesamiento de señales: los datos limpios reales no son conocidos. Nuestra estrategia se basa en el supuesto clave de que los métodos de *denoising* se aplican correctamente de manera de no dañar las señales microsísmicas. Así, se espera que el histograma de los datos filtrados por el mejor método se desvíe al máximo del histograma de los datos ruidosos de entrada. Las pruebas numéricas validan esta afirmación. Por lo tanto, la medición de la distancia de los histogramas antes y después del *denoising* permite una evaluación cuantitativa de los métodos de filtrado.

2.7. Conclusiones

La forma estándar de medir el desempeño de cualquier proceso de filtrado de ruido se basa en la inspección visual de la versión filtrada de los datos y/o su residuo. Esta técnica es muy subjetiva y suele fallar en demostrar el verdadero valor de una determinada técnica de filtrado. En contraste, se propone un método basado en la medición de distancias de histogramas de polarización para evaluar y comparar el rendimiento de un conjunto de algoritmos de *denoising*. Este enfoque representa una técnica útil y poderosa para comparar cuantitativamente el valor real del impacto de dichos métodos sobre la señal de interés.

Los resultados usando datos sintéticos y de campo indicaron que cuando la relación S/R es pobre o moderada, el método SVD obtiene los mejores resultados, seguido por la transformada Radon en un dominio hiperbólico restringido. En otras palabras, para aquellos escenarios donde es crítico realizar un *denoising*, los métodos más sofisticados demostraron tener mejor desempeño. Por otro lado, para los casos donde la relación S/R es grande, los resultados indicaron que es preferible elegir métodos de filtrado más convencionales, ya sea un clásico filtro pasa banda, o en casos extremos, simplemente no aplicar un *denoising*.

Capítulo 3

Localización de eventos microsísmicos mediante evolución diferencial

La localización de las microfracturas generadas durante una estimulación hidráulica en yacimientos no convencionales puede plantearse como un problema de optimización no lineal. Para su resolución, en este capítulo se propone utilizar un algoritmo de evolución diferencial (DE, por sus siglas en inglés). El mismo es aplicado para minimizar una función de costo que se construye a partir de un modelo matemático que representa al subsuelo. Examinamos dos tipos de pruebas, una usando los tiempos de arribo teóricos y la otra basada en datos sintéticos que fueron picados utilizando un método de picado automático. En ambos casos, el desempeño del algoritmo DE se analiza y compara con los resultados obtenidos mediante *very fast simulated annealing* (VFSA) y *particle swarm optimization* (PSO). Los resultados demuestran que DE logra localizar los eventos microsísmicos con gran precisión a expensas de un mayor número de evaluaciones de la función de costo que VFSA, pero menor que PSO.

3.1. Introducción

El principal objetivo de un proceso de estimulación hidráulica es mejorar la productividad de los yacimientos hidrocarburíferos formados por rocas de baja permeabilidad, ya sean arcillas *tight-gas* o ciertas formaciones arenosas (Warpinski et al., 2001; Warpinski and Du, 2010b). Esto se consigue inyectando fluidos a altas presiones dentro de la roca, lo que crea microfracturas que permiten que los fluidos entrampados en los espacios porales de la roca fluyan hacia el pozo estimulado. La producción exitosa de hidrocarburos en este tipo de formaciones depende fuertemente de la producción individual de cada una de las etapas de fracturamiento (Cipolla et al., 2011) y, en general, de la generación de grandes áreas de fractura (Suárez-Rivera et al., 2006; Warpinski et al., 2001; Warpinski and Du, 2010b). Las

microfracturas creadas mientras se estimula el reservorio no convencional inducen pequeños eventos sísmicos que pueden ser detectados con instrumentos localizados estratégicamente en pozos de monitoreo cercanos o, incluso, sobre la superficie inmediatamente superior a la zona estimulada. Conocer la localización de las fracturas de una forma precisa es de gran importancia, no solo para evaluar la "completación" y el tratamiento de fractura (Cipolla et al., 2011), sino también para evitar potenciales daños ambientales, como contaminación de reservorios de agua dulce. Además, una mala estimación de las dimensiones de fractura puede conducir a incertezas en análisis posteriores del reservorio.

El mapeo de eventos microsísmicos puede ser visto como la aplicación directa de los principios generales de la sismología (Cipolla et al., 2012). La técnica apunta a localizar y caracterizar la geometría de fractura creada durante un proceso de estimulación hidráulica. La construcción de esta geometría se basa en el agrupamiento de eventos sísmicos que corresponden a una misma etapa de fractura. A pesar de que esta geometría de fractura siempre termina siendo una aproximación de la red de fracturas real (Weng et al., 2011), la precisión en la localización de los microsismos impacta directamente en la calidad del modelo geométrico. Dada la naturaleza de los eventos microsísmicos, que se caracterizan en general por ser de muy baja amplitud y tener baja relación S/R, su localización precisa puede ser difícil a nivel técnico. Además, las incertezas pueden ser significativas ya que el modelo de velocidades del subsuelo no se conoce con precisión (Maxwell, 2009b). Por estas razones, resulta fundamental desarrollar algoritmos eficientes y precisos que permitan la localización de eventos microsísmicos en ambientes de alto contenido de ruido.

En este capítulo se aplica DE (Storn and Price, 1996; Price et al., 2006) para estimar las coordenadas de los eventos microsísmicos. DE está basado en un procedimiento no determinístico, desarrollado específicamente para resolver problemas de optimización global donde se busca el extremo absoluto de una función de costo de tipo no lineal y/o multimodal. Por otra parte, DE es un algoritmo "evolutivo" que presenta muchas ventajas, tales como la simplicidad, independencia de derivadas, y la capacidad de trabajar con parámetros de tipo continuo. Los resultados prueban que el algoritmo basado en DE puede localizar los eventos microsísmicos de un tratamiento de fractura con gran precisión. Además, se analiza su desempeño comparando nuestros tests con los resultados presentados en Lagos et al. (2014), donde se resolvió el mismo problema utilizando los métodos de optimización global VFSA y PSO.

Este capítulo está organizado en cuatro secciones. Primero, se presentan los principios y elementos básicos que describen el método DE. En segundo lugar, se presenta el modelo de geometría utilizado para las pruebas y cálculos. La tercera sección muestra los resultados obtenidos al aplicar DE al modelo y realizamos una comparación con los resultados obtenidos con VFSA y PSO. Finalmente, en las conclusiones se resumen los principales aportes del enfoque propuesto.

3.2. Evolución diferencial (DE)

Preliminares

El DE es un algoritmo poblacional y como tal, utiliza una población de N_P vectores, $\mathbf{x}_{i,g}$, donde $i = 0, ..., N_P - 1$ representa el índice de los vectores. Con el objetivo de converger a la solución óptima, el método va evolucionando dicha población en saltos evolutivos discretos. Cada una de estas iteraciones forman un conjunto poblacional diferente al anterior. En la bibliografía, estos conjuntos poblacionales suelen denominarse "generaciones" y aquí se enumeran con el índice g. Este índice también puede encontrarse como índice "generacional".

Los elementos de los vectores que componen una población son los parámetros que caracterizan el problema a optimizar, cuyo número define la dimensión de los vectores de la población. En la literatura, comúnmente se refiere a estos vectores como "vectores de parámetros" o simplemente "puntos" dentro del espacio vectorial. Nos referiremos a estos vectores mediante cualquiera de estos dos nombres. A diferencia de otros métodos de optimización global como *simulated annealig* (SA), en cada generación DE crea nuevos vectores combinando linealmente vectores de generaciones anteriores. Esto es, los vectores de una nueva generación no son el resultado de extraer números de alguna función de distribución de densidad, como en el caso de SA, sino que se obtienen a partir de la evolución de generaciones anteriores.

En DE, los vectores $\mathbf{x}_{i,g}$ son perturbados por una diferencia escalada de vectores seleccionados aleatoriamente para producir los llamados "vectores de prueba", $\mathbf{u}_{i,g}$, cuyo objetivo es explorar el espacio de búsqueda. En la siguiente etapa, los vectores de prueba compiten contra los vectores del mismo índice, $\mathbf{x}_{i,g}$. Esta competencia entre parejas de vectores se repite hasta que todos los N_P vectores hayan competido contra un vector de prueba. Una vez terminado este proceso, los vectores supervivientes son aquellos que se ajustan a un criterio de selección y son aptos para convertirse en miembros de la próxima generación. Por lo general, el número N_P permanece fijo durante todo el proceso, aunque no hay restricciones teóricas para esto. La idea principal en DE viene dada por el esquema utilizado para generar vectores de parámetros de prueba; esto es, mediante la combinación de vectores ya existentes. Si el vector resultante produce un menor valor de función de costo que un miembro predeterminado de la población, este último es reemplazado por el nuevo vector.

El primer paso de cualquier algoritmo basado en DE consiste en generar una población inicial. Después de la inicialización de la primera población de vectores, las iteraciones siguientes constan de tres pasos diferentes: "mutación-recombinación", "entrecruzamiento" y "selección". La mutación se refiere al proceso destinado a ampliar el espacio de búsqueda, mientras que el objetivo de la recombinación es reutilizar individuos que fueron exitosos

previamente. Entrecruce se refiere a la recombinación discreta que produce vectores de prueba a partir de la combinación aleatoria de mutantes y vectores ordinarios. Finalmente, la selección imita la supervivencia del más apto apreciado en la mayoría de los fenómenos de la naturaleza.

Inicialización

Los elementos de la población inicial se seleccionan aleatoriamente teniendo en cuenta las condiciones de contorno definidas por el problema a optimizar. Una vez que se especifican los límites iniciales, un generador de números aleatorios asigna cada parámetro de cada vector a un valor dentro de los rangos permitidos. Por lo tanto, para la generación inicial (es decir, $g = 0$), se tiene que:

$$x_{j,i,g=0} = b_{j,Low} + \xi_j(b_{j,Up} - b_{j,Low}), \tag{3.1}$$

donde el índice $j \in [1, n]$ indica el parámetro individual dentro del vector i-ésimo, n denota la dimensión del problema, $\xi_j \in [0, 1)$ es un número aleatorio uniformemente distribuido, y $b_{j,Up}$ y $b_{j,Low}$ son los límites superior e inferior de cada parámetro, respectivamente. En términos microsísmicos, estos límites se pueden determinar en función de la ubicación de los disparos de perforación y la región donde se espera que ocurran eventos durante la fractura.

En la mayoría de los casos, DE se inicializa con una distribución uniforme, pero si la geometría del problema lo exige, también se puede utilizar otra distribución. La decisión sobre qué distribución utilizar está relacionada con la información disponible a priori sobre los valores esperados de los parámetros. En este sentido, si se dispone de una solución aproximada, una distribución normal representaría una alternativa adecuada para lograr una convergencia más rápida. Esto último debe tratarse con cuidado, ya que la elección de una distribución normal alrededor de una solución inicial aproximada también aumentaría la probabilidad de convergencia prematura (Price et al., 2006) o de estancamiento. En general, la elección estándar es una distribución uniforme, ya que refleja la falta de conocimiento previo sobre la solución.

Mutación y entrecruzamiento

Una vez creada la población inicial, el siguiente paso del algoritmo consiste en mutar y recombinar los vectores de población. Para cada vector de parámetros $\mathbf{x}_{i,g}$ se calcula una diferencia escalada de dos vectores tomados aleatoriamente de la misma generación y se la suma a $\mathbf{x}_{i,g}$. Tanto los dos vectores seleccionados al azar como $\mathbf{x}_{i,g}$ deben ser diferentes entre sí. Esta estrategia produce N_P nuevos vectores de prueba definidos como "mutantes"

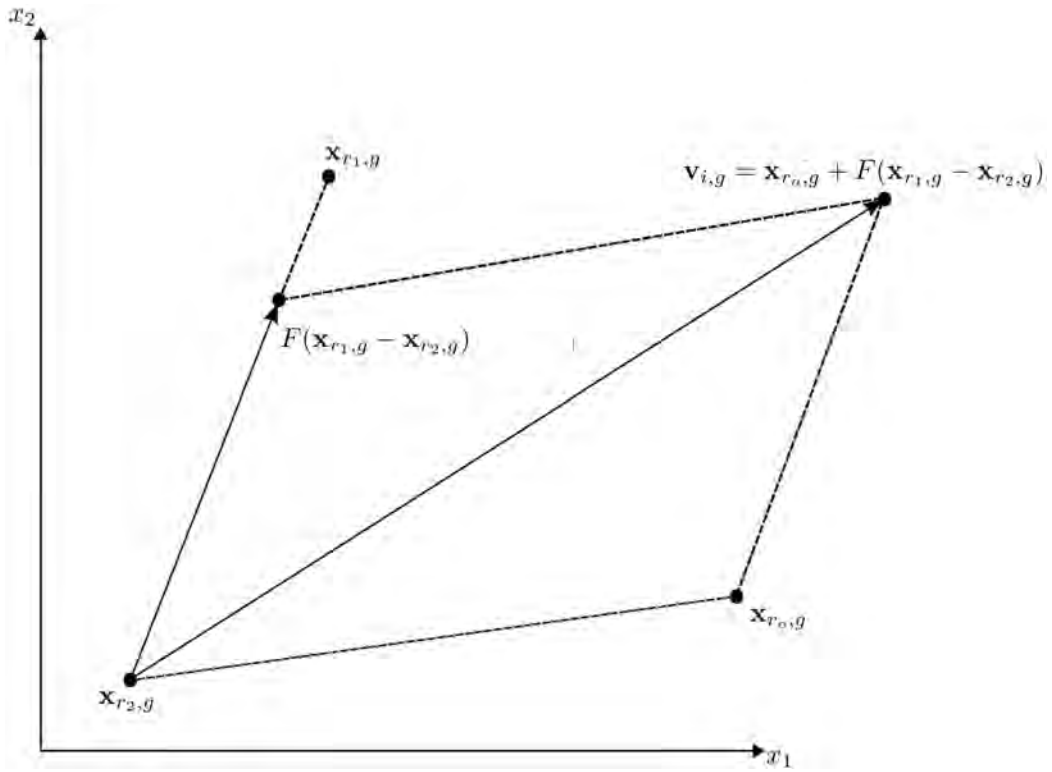


Figura 3.1: Un vector mutante, $\mathbf{v}_{i,g}$, se obtiene a partir de la suma de un vector base, $\mathbf{x}_{r_0,g}$, con una diferencia pesada, $F(\mathbf{x}_{r_1,g} - \mathbf{x}_{r_2,g})$. Representación gráfica para un espacio 2D

(ver Figura 3.1). Dados tres vectores diferentes $\mathbf{x}_{r_0,g}$, $\mathbf{x}_{r_1,g}$, y $\mathbf{x}_{r_2,g}$, el mutante $\mathbf{v}_{i,g}$ se crea usando:

$$\mathbf{v}_{i,g} = \mathbf{x}_{r_0,g} + F\,(\mathbf{x}_{r_1,g} - \mathbf{x}_{r_2,g}). \tag{3.2}$$

El factor de escala F es un número positivo que controla la velocidad con la cual la población cambia o evoluciona. No hay límite superior para restringir F. Sin embargo, los valores más efectivos rara vez son mayores que 1.0 (Price et al., 2006; Neri and Tirronen, 2010). El valor óptimo de F generalmente disminuye con la raíz cuadrada del tamaño de la población (Simon, 2013). Como se indica en Price et al. (2006), si se consideran todas las posibles diferencias escaladas entre los elementos de una población, la distribución resultante es tal que su media es cero. Por esta razón, F también es responsable de la naturaleza no duplicativa de los vectores de prueba, así como también de desplazar el foco de búsqueda de posibles extremos locales.

El vector $\mathbf{x}_{r_0,g}$ comúnmente se llama "índice de vector base". Es importante destacar que existen numerosas estrategias alternativas para obtener un vector mutante. La estrategia elegida para esta Tesis es una modificación a la denominada "*DE/curr. to best/1*" (Neri and Tirronen, 2010; Qin and Suganthan, 2005), donde *current to best* ("del actual al mejor") representa, como se describe más abajo, la forma en que se seleccionan los vectores. Matemáticamente, esta estrategia está representada por:

$$\mathbf{v}_{i,g} = \mathbf{x}_{i,g} + F_1\,(\mathbf{x}_{r_2,g} - \mathbf{x}_{r_3,g}) + F_2\,(\mathbf{x}_{best,g} - \mathbf{x}_{i,g}), \tag{3.3}$$

donde $\mathbf{x}_{best,g} \neq \mathbf{x}_{r_2,g}$ y $\mathbf{x}_{best,g} \neq \mathbf{x}_{r_3,g}$. El vector *current*, $\mathbf{x}_{i,g}$, está dado por el i-ésimo índice base (mientras que $\mathbf{x}_{best,g}$ es el mejor elemento de la población actual[*]). El "*/1*" en el nombre de la estrategia se refiere a la cantidad de diferencias entre vectores que son involucradas en la mutación. Aquí, la principal diferencia con la estrategia estándar "*DE/curr. to best/1*" es que se permite que las escalas de mutación dadas por F_1 y F_2 tomen valores diferentes. Los vectores restantes $\mathbf{x}_{r_2,g}$ y $\mathbf{x}_{r_3,g}$ se eligen al azar. En esta Tesis, los vectores mutantes son generados utilizando la ecuación 3.3.

Una vez finalizado el paso de la mutación y recombinación, se realiza una estrategia de mutación complementaria definida como *crossover* o "entrecruzamiento". Es el mecanismo mediante el cual se construyen vectores de prueba a partir de valores de parámetros que se han copiado de dos vectores diferentes. DE cruza cada uno de los vectores con un vector mutante para generar un nuevo vector de prueba $\mathbf{u}_{i,g}$ a partir de $\mathbf{v}_{i,g}$ y $\mathbf{x}_{i,g}$. Esto lo hace de la siguiente manera:

$$u_{j,i,g} = \begin{cases} v_{j,i,g} & \text{si } \nu_j \leq C_r \text{ o } j = \hat{j} \\ x_{j,i,g} & \text{c.c.,} \end{cases} \tag{3.4}$$

[*]El que corresponde al menor valor de la función de costo

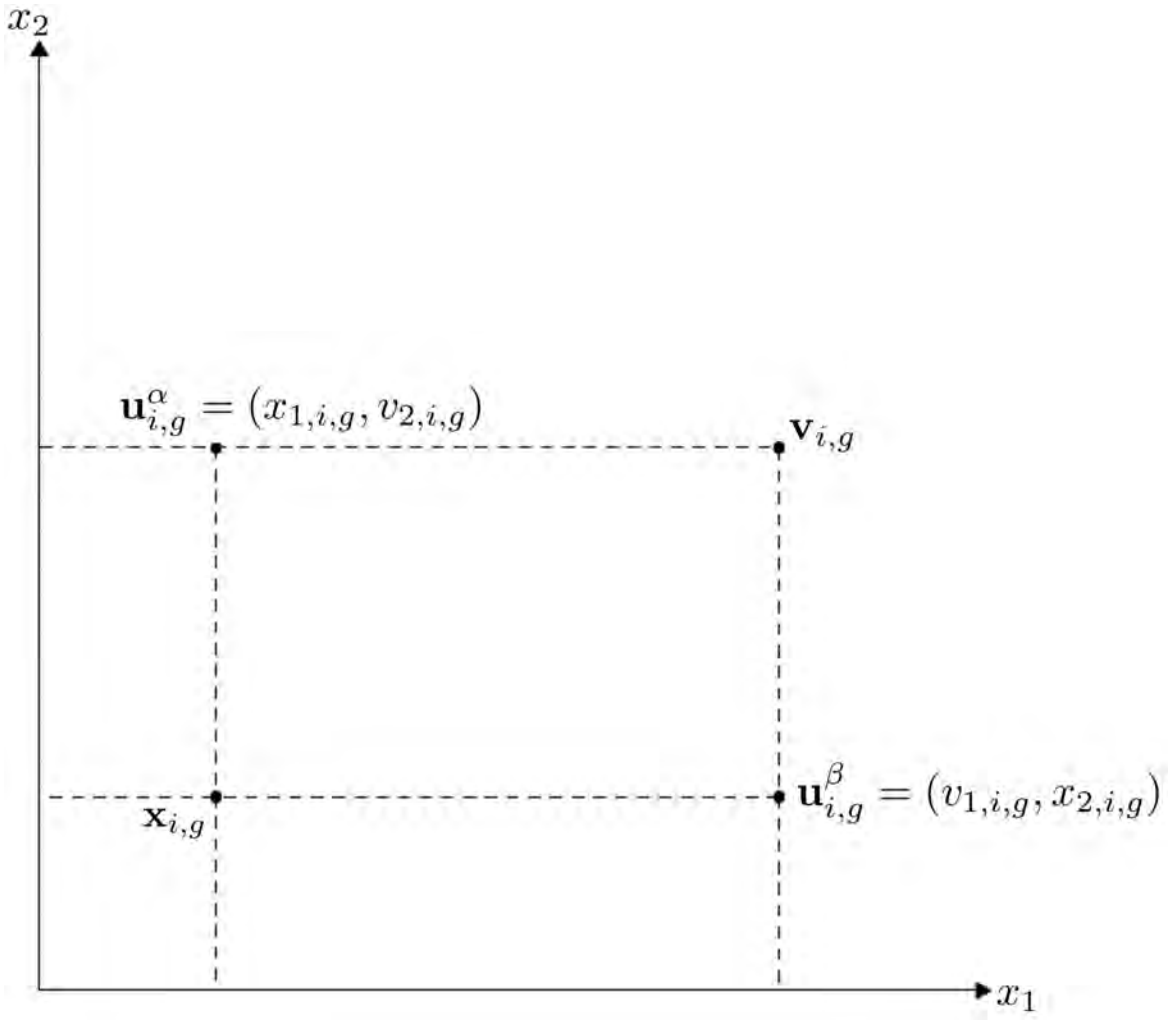


Figura 3.2: Posibles vectores de prueba, $\mathbf{u}^{\alpha}_{i,g}$ y $\mathbf{u}^{\beta}_{i,g}$, que pueden resultar de cruzar uniformemente a un vector mutante, $\mathbf{v}_{i,g}$, con $\mathbf{x}_{i,g}$, en un espacio 2D.

donde $j \in [1, n]$ representa los componentes de los vectores, ν_j es un número aleatorio tomado de una distribución uniforme en $[0, 1)$, $\hat{j}$ es un número entero al azar en $[1, n]$, y C_r es la "probabilidad de cruce". La probabilidad de cruce, $C_r \in [0, 1]$, controla qué valores de parámetros de un vector mutante $\mathbf{v}_{i,g}$ se copian o pasan a $\mathbf{u}_{i,g}$. Esto se hace comparando C_r con el número aleatorio ν_j. Si este último es menor o igual a la probabilidad de cruce, entonces el parámetro j-ésimo se hereda del vector mutante. Si esta condición falla, el parámetro se toma de $\mathbf{x}_{i,g}$. Aunque esta condición tiene como objetivo generar nuevos vectores de prueba mediante la combinación de parámetros de dos vectores, no garantiza que el vector de prueba $\mathbf{u}_{i,g}$ sea diferente a $\mathbf{x}_{i,g}$. Para evitar este problema, el parámetro de prueba con índice elegido al azar $\hat{j}$ siempre se toma del mutante. La Figura 3.2 muestra posibles resultados que pueden resultar de esta operación de cruce.

Selección

Finalmente, el método necesita hacer una selección de los elementos de la población actual. La decisión de qué vectores de parámetros pueden sobrevivir o no a la próxima generación se basa en la comparación directa de los respectivos valores de función de costo. Por lo tanto, si el vector de prueba $\mathbf{u}_{i,g}$, tiene un valor de función de costo igual o menor que el vector $\mathbf{x}_{i,g}$, también llamado "vector objetivo", el primero ocupará el lugar de este último en la próxima generación. Si, por el contrario, su comparación arroja el resultado opuesto, el vector que prevalecerá será el objetivo. La próxima generación es completada con los nuevos N_P elementos una vez realizado este experimento para todos los vectores de prueba.

3.3. Localización de microsismos

Para analizar el desempeño de DE en la localización de eventos microsísmicos se realiza una comparación contra otros algoritmos de optimización global disponibles. En este sentido, se simula el mismo escenario de monitoreo presentado en Lagos et al. (2014), donde se analizan dos métodos de optimización diferentes y se compararon en términos de su efectividad y eficiencia para la localización de eventos microsísmicos. Siguiendo esta idea, se configura una geometría que consiste en un único pozo vertical de monitoreo. El arreglo de receptores está compuesto por 8 geófonos 3C espaciados por 30 m. El receptor más superficial se encuentra en la posición $(x_{R_1}, y_{R_1}, z_{R_1}) = (200, 100, 440)$ m, mientras que la fuente (evento microsísmico) se ubica en las coordenadas $(x_s, y_s, z_s) = (600, 300, 500)$ m. Esta configuración de pozo monitor y fuente puede observarse gráficamente en la Figura 3.3. El subsuelo es modelado por un semiespacio homogéneo e isotrópico con velocidades de onda compresional y de corte $v_P = 3500$ m/s y $v_S = 2400$ m/s, respectivamente.

Este escenario de monitoreo y geometría implica una simetría en la que cualquier evento ubicado a una distancia radial $\sqrt{x^2 + y^2}$ del pozo (eje de simetría) exhibirá los mismos tiempos de arribo. Esto último conduce a un problema de no unicidad. Por simplicidad, asumiremos que se conoce el acimut del evento, lo que es equivalente a conocer una de las dos coordenadas planares (x o y). Bajo este supuesto, el problema se reduce a una geometría 2D en (x', z), donde $x' = \sqrt{x^2 + y^2}$. Finalmente, la solución de estimar la posición de la fuente a partir de los tiempos de llegada observados en los receptores se reduce a minimizar la función de costo (Lagos et al., 2014)

$$E(x, y, z) = \left(\frac{1}{N_R} \sum_{i=1}^{N_R} [\Delta t_i(x, y, z) - \Delta t_i^{obs}]^2 \right)^{\frac{1}{2}}, \tag{3.5}$$

donde N_R es el número de receptores y Δt_i y Δt_i^{obs} son, respectivamente, las diferencias de tiempo calculado y observado entre las llegadas de las ondas S y P. Para un medio

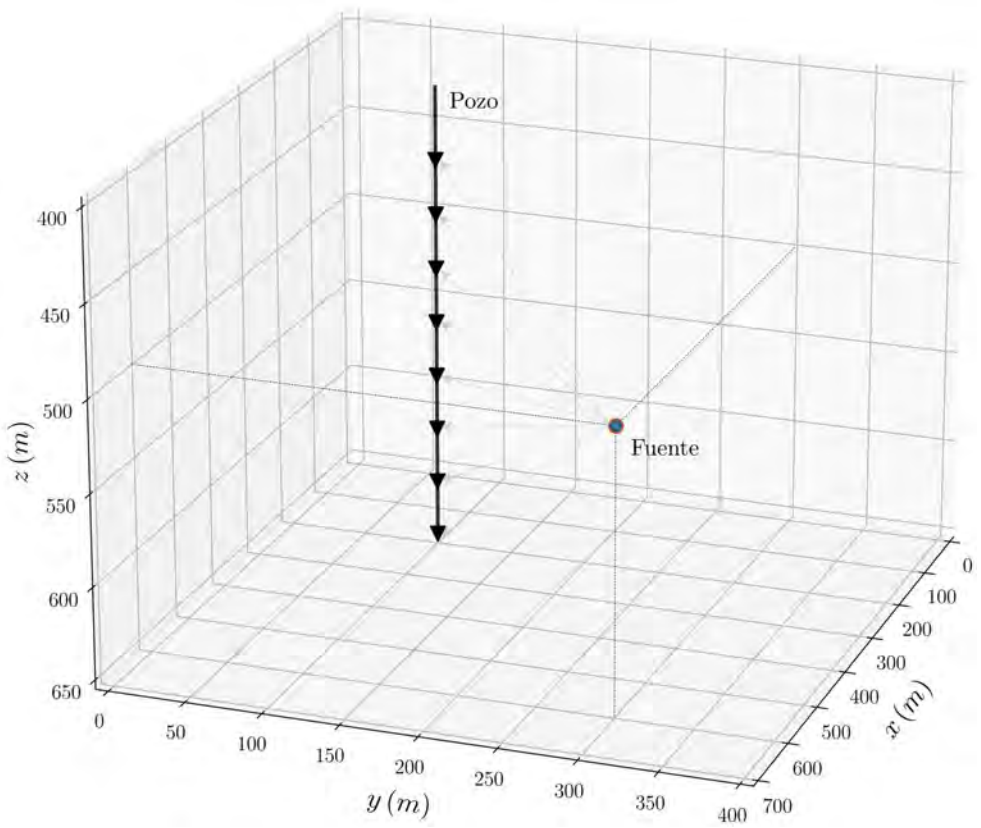


Figura 3.3: Geometría de monitoreo. Un pozo monitor vertical de 8 geófonos (triángulos negros). Se muestra la fuente en la posición $(600, 300, 500)$ m.

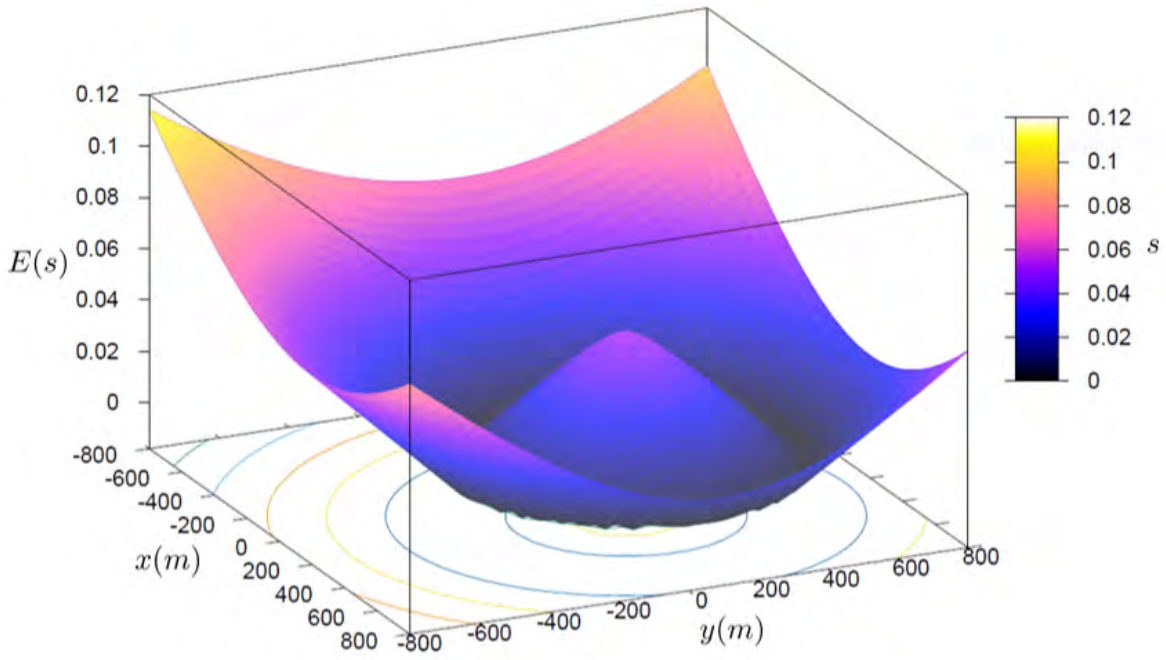


Figura 3.4: Función de costo para $z = 500$ m (la función de costo $E(s)$ tiene unidades de segundos). Se puede observar un mínimo en forma de circunferencia (o "anillo"). Esto implica la existencia de infinitas soluciones localizadas a una distancia radial constante al eje de simetría (el pozo de monitoreo).

homogéneo,

$$\Delta t_i(x, y, z) = \Delta t_i(x', z) = r'_i(x', z)\left(\frac{1}{V_S} - \frac{1}{V_P}\right), \tag{3.6}$$

donde $r'_i = \sqrt{(x'_i - x')^2 + (z_i - z)^2}$. La fuente se encuentra ahora en las coordenadas acimutales y profundidad $(x', z) = (670.82, 500.0)$ m. La Figura 3.4 muestra la función de costo $E(x, y, z = 500)$. Se observa que el mínimo no es un único punto en el plano (x, y), sino más bien una circunferencia centrada alrededor del pozo (eje de simetría). Se ve aquí la importancia de asumir que se conoce el acimut, ya que permite evitar este problema de no unicidad. Además, se debe tener en cuenta que $x' > 0$. No obstante, x puede tomar tanto valores positivos como negativos. Esto significa que dos eventos diferentes ubicados en (x, z) y $(-x, z)$ mostrarán la misma configuración para sus tiempos de arribo. Este es un caso particular de la ambigüedad acimutal presentada al considerar una búsqueda en un espacio 3D con esta geometría de adquisición. Por este motivo, limitamos la búsqueda a valores no negativos de x, $0 < x \leq 800$ m. El valor superior de 800 m en x representa una distancia razonable para el problema de localización de eventos microsísmicos. De manera similar, dado que no se espera que ocurran fuentes poco profundas, las profundidades son restringidas a valores superiores a 200 m.

3.4. Resultados

En esta sección se describe la aplicación de DE para la ubicación de eventos microsísmicos. Para ello, se utilizan tanto datos sintéticos ruidosos como sin ruido, y se comparan los resultados con aquellos expuestos en Lagos et al. (2014). En este último trabajo, se utilizan PSO y VFSA para resolver el mismo problema y comparar sus rendimientos. Además, realizan una búsqueda clásica por grillado (GS: *grid search*). Los autores encuentran que tanto PSO como VFSA superan a GS. Además, demostraron que para la misma tolerancia en la precisión, VFSA se desempeña considerablemente mejor que PSO en términos de eficiencia.

Siguiendo Lagos et al. (2014), se analiza el rendimiento de DE utilizando dos conjuntos de datos diferentes. Primero, se utilizan las diferencias reales de tiempo de arribo (sin ningún error asociado al picado) para cuantificar su comportamiento en un escenario óptimo. En segundo lugar, se aplica el método a un conjunto de datos simulados y contaminado con ruido gaussiano de banda limitada. Para tener un conocimiento estadístico sobre el rendimiento del método, el experimento de localización se repite 200 veces, lo que permite obtener un valor medio μ y una desviación estándar σ para cada uno de los conjuntos de datos.

En cuanto a los parámetros que definen a DE, usamos una población distribuida uniformemente para inicializar el espacio de búsqueda. Siguiendo a Storn (1996), el tamaño de la población N_P se establece como al menos 10 veces el tamaño de la dimensión del espacio de parámetros. Esto da 20 eventos, ya que la dimensión del espacio de parámetros (x', z) es 2. Tras realizar varias pruebas, observamos que aumentar el número de eventos iniciales N_P no solo no mejora significativamente la solución estimada, sino que también aumenta significativamente el número de evaluaciones de la función de costo necesarias en cada iteración. En lugar de utilizar factores de escala de mutación constantes, F_1 y F_2 son seleccionado de una distribución uniforme, cambiando así en cada iteración. La probabilidad de cruce se fija en 0.5 y el número máximo de generaciones se establece en 5000.

Los tiempos de arribo teóricos para el primer análisis se pueden calcular utilizando las siguientes ecuaciones:

$$t_P = t_0 + \Delta t_P = t_0 + \left(\frac{r}{v_P}\right) \tag{3.7}$$

$$t_S = t_0 + \Delta t_S = t_0 + \left(\frac{r}{v_S}\right), \tag{3.8}$$

con

$$r = \sqrt{(x_R - x_s)^2 + (y_R - y_s)^2 + (z_R - z_s)^2},$$

donde Δt_P, Δt_S son los tiempos de viaje desde la fuente a cada geófono para las fases P y S, respectivamente. Restando la expresiones (3.8) y (3.7) para todos los receptores se

Misfit	10^{-3} s			10^{-4} s		
Método	DE	VFSA	PSO	DE	VFSA	PSO
$\bar{x}'$	669.69	669.51	670.29	670.81	670.83	670.81
$\bar{z}$	495.38	498.55	497.56	500.35	499.87	499.93
$\sigma_{x'}$	4.77	4.82	4.19	0.45	0.48	0.47
σ_z	25.06	26.86	22.40	2.39	27.32	25.60
$\bar{N}_{fev}$	180	89	300	500	263	2300

Tabla 3.1: Comparación entre DE, VFSA y PSO. Datos teóricos. Valores medios luego de 200 experimentos. Todos los valores mostrados están en metros, exceptuando al número de evaluaciones de función de costo $\bar{N}_{fev}$. Las cantidades $\sigma_{x'}$ y σ_z indican las incertidumbres en las direcciones x' y z, respectivamente.

obtiene

$$\Delta t_{th} = t_S - t_P = \left(\frac{1}{v_S} - \frac{1}{v_P}\right) r, \tag{3.9}$$

donde Δt_{th} indica las diferencias de tiempo teóricas buscadas. Vale la pena notar que este método ignora el conocimiento del tiempo inicial t_0. De lo contrario, t_0 se convertiría en una incógnita extra a estimar. El problema de considerar el parámetro adicional t_0 está más allá del alcance de este trabajo.

Debido a que DE es un método de optimización iterativo, adicionalmente se debe definir un criterio de corte que determine hasta cuando debe realizarse el proceso de búsqueda. Para esto, elegimos definir dos criterios diferentes. Así, el DE terminará de ejecutarse cuando la precisión alcance una tolerancia o "umbral" pre-establecido, o bien se alcance un número máximo de iteraciones. En particular, en esta tesis realizamos experimentos con dos umbrales diferentes. Esto permite estudiar la respuesta del método cuando la exigencia sobre la precisión aumenta o disminuye. Teniendo en cuenta que la mayoría de los geófonos de la industria son capaces de registrar señales microsísmicas con frecuencias de muestreo entre 0.1 y 0.25 ms, se eligen valores de umbral que contemplen estas cantidades. Por lo tanto, establecemos las tolerancias en $1{\times}10^{-3}$ s y $1{\times}10^{-4}$ s, respectivamente. La Tabla 3.1 muestra los resultados de estos experimentos, así como también los presentados en Lagos et al. (2014). Se observa que, para el umbral de $E = 1 \times 10^{-3}$ s, los valores de $\sigma_{x'}$ y σ_z difieren entre ellos en un orden de magnitud. Además, los tres métodos resuelven la localización entregando valores similares en ambas cantidades.

Las diferencias más notables asociadas a las incertidumbres de cada método surgen cuando el criterio para el umbral toma el valor de $1{\times}10^{-4}$ s. Bajo esta condición más restrictiva, a pesar de que los tres métodos son capaces de ejecutar su tarea devolviendo incertidumbres similares en la dirección perpendicular al eje de la herramienta (σ_x), σ_z disminuye significativamente para DE (un orden de magnitud). Este es un resultado no-

table, ya que implica una mejora considerable en la localización respecto de los resultados arrojados por VFSA y PSO. Una incertidumbre menor para la coordenada z puede conducir a una mejor estimación de la altura de la fractura y, por lo tanto, puede mejorar las estimaciones del volumen del reservorio estimulado.

En principio, la diferencia observada entre los valores de $\sigma_{x'}$ y σ_z, (en este caso, $\sigma_{x'} < \sigma_z$ para los 3 métodos) no puede adjudicarse a la capacidad de los métodos para determinar mejor o peor una o otra coordenada, sino que está principalmente dominada por la configuración fuente-receptor. Así, puede existir otro evento que, teniendo otra posición, se localice con una relación entre $\sigma_{x'}$ y σ_z diferente. Por ejemplo, si consideramos el mismo medio, sistema de coordenadas y posición de la fuente, pero pivoteamos el arreglo de receptores 90 grados alrededor de dicha fuente, obtendríamos los valores opuestos para ambas cantidades. Esto es, $\sigma_{x'} > \sigma_z$. Habiendo dicho esto, es importante mencionar que la cantidad σ_z es más sensible al llamado "ángulo de apertura efectiva" que $\sigma_{x'}$. Este ángulo está determinado por todas las trayectorias de los rayos que, formando un "abanico", llegan al conjunto de receptores (Zimmer, 2010). El mismo se define como el ángulo de apertura entre los rayos fuente-receptor teniendo en cuenta el receptor más somero y el más profundo. Por lo expuesto en Zimmer (2010), cuanto mayor sea esta apertura, mayor será la precisión en la ubicación de la profundidad z del evento. Además, como la precisión en la profundidad mejora con ángulos aparentes más grandes, la incertidumbre en dicha coordenada también disminuye al disminuir la distancia al conjunto de receptores (Verkhovtseva and Shaffner, 2013). Lamentablemente, los resultados de estos trabajos se basan en la inspección geométrica de las elipses de error de localización de los eventos y no ofrecen un análisis matemático que justifique la mayor sensibilidad de σ_z respecto de este ángulo. Si bien este fenómeno no es responsable de la relación entre $\sigma_{x'}$ y σ_z, es cierto que el ángulo efectivo está determinado por la configuración fuente-receptor* y, para el caso típico de un único pozo vertical, es usual observar incertidumbres tales que $\sigma_z > \sigma_{x'}$ (Zimmer, 2010; Verkhovtseva and Shaffner, 2013).

En términos de eficiencia computacional, es estándar para la evaluación de los métodos evolutivos referir el número de evaluaciones de la función de costo N_{fev}. Para DE, $N_{fev} = (iter + 1) \times N_P$, donde $iter$ refiere al número de generaciones. Por lo tanto, es evidente que la diferencia en eficiencia con el PSO descripto en Lagos et al. (2014) está principalmente relacionada al número de iteraciones necesarias para la convergencia. Como puede observarse en la Tabla 3.1, el método convergió utilizando, en promedio, $N_{fev} = 180$ y 500, respectivamente, mientras que VFSA y PSO requirieron un promedio de $N_{fev} = 89, 263$ y $300, 2300$, respectivamente. Está claro que para ambas tolerancias, DE puede llegar a la solución deseada requiriendo menos operaciones que PSO, pero más que

*También influye la naturaleza y geometría del medio, así como las fases de onda utilizadas para realizar la localización. Ver Zimmer (2010).

VFSA.

Como se mencionó anteriormente, también se prueba el algoritmo bajo un entorno realista donde los datos difieren de los valores teóricos, simulando así, tener observaciones ruidosas. Ya que nuestra intención es la de realizar una comparación justa, el conjunto de datos utilizado en este trabajo es exactamente el mismo que el utilizado en Lagos et al. (2014). En el trabajo citado se generan tiempos de arribo considerando una microfractura asociada a un mecanismo de cizalla en el plano (x, z), con un deslizamiento en la dirección x. La señal propagada es una ondícula de Ricker con una frecuencia pico de 100 Hz. Posteriormente, los datos fueron contaminados con ruido gaussiano de banda limitada y relación S/R de 3. Los eventos fueron detectados y los tiempos picados automáticamente utilizando el método de Allen modificado (MAM) (Sabbione and Velis, 2013). Finalmente, se calcularon las diferencias de tiempo correspondientes mediante la ecuación (3.9). Cuando se registran datos reales, es posible que una señal que viaja a través de un medio (con cierta complejidad) no alcance al conjunto completo de geófonos. Además, aunque los mismos logren llegar a todos los receptores, es posible que en ciertas trazas, algunas de las fases sean indistinguibles. Como es de esperarse, esta dificultad aumenta considerablemente en entornos de baja relación S/R. En una situación extrema, cuando la relación S/R es crítica, es esperable que la mayor parte del carácter de la señal esté enmascarada por el ruido y la detección se vuelve problemática (y/o imposible). Durante la simulación del dato, cuando ocurren cualquiera de las situaciones descriptas anteriormente, el MAM presenta dificultades para seleccionar los tiempos de arribo correctamente, y en algunas trazas, no detecta la fase deseada, sea P, S, o ambas. En este escenario, no se podrá utilizar la ecuación (3.9) para calcular Δt, por lo que disminuirá la cantidad de dato disponible.

Dos de las principales fuentes de incertidumbre en la ubicación de eventos microsísmicos se asocian a errores en los tiempos seleccionados o a la falta de datos debido a la imposibilidad de detectar la fase deseada. En estos casos, es necesario considerar solo aquellas trazas donde ambas fases son seleccionadas. La Tabla 3.2 muestra los datos teóricos y sintéticos de entrada que consideramos para el método DE. Debe tenerse en cuenta que el conjunto de datos sintéticos disponible comprende solo 8 de las 24 diferencias de tiempo posibles.

A continuación, se realizan 200 experimentos y se establece el criterio de umbral en $E \sim 1.3 \times 10^{-3}$ s, donde se encuentra el mínimo de la función de costo para este conjunto de datos en particular. Los resultados muestran que se logra converger en un promedio de 820 evaluaciones de la función de costo. El número máximo y mínimo de iteraciones son 75 y 21, respectivamente (ver Figura 3.5). En todos los casos, el evento microsísmico se localiza con éxito dentro de la tolerancia deseada. Considerando todos los experimentos, el valor medio de la solución resulta $(\bar{x}', \bar{z}) \simeq (668.6, 514.6)$ m, lo que significa que el error en la distancia entre las ubicaciones de la fuente real y la fuente estimada es de aproximadamente 14.78 m. Las desviaciones estándar calculadas para todos los experimentos toman valores

Geófono	Δt_{th}(s)	Δt_{sint}(s)		
		$\underline{x}$	$\underline{y}$	$\underline{z}$
1	0.0590	0.059	-	-
2	0.0586	0.059	-	-
3	0.0585	0.058	0.058	-
4	0.0586	0.059	-	-
5	0.0590	0.059	-	-
6	0.0606	-	-	-
7	0.0617	0.057	-	-
8	0.0585	0.063	-	-

Tabla 3.2: Diferencias de tiempo teóricas y picadas para cada componente de los geófonos usadas para las pruebas de DE. Un guión indica que el algoritmo de detección automática no pudo picar alguna de las fases para la componente correspondiente.

Método	DE	VFSA	PSO
$\sigma_{x'}$	0.03	2.77	2.92
σ_z	0.18	18.28	18.91
$\bar{N}_{fev}$	820	673	39500
Δr	14.78	14.80	14.80

Tabla 3.3: Incertidumbres, número de evaluaciones de función de costo y error obtenido por DE, VFSA y PSO, utilizando el dato sintético. Todos los valores, excepto $\bar{N}_{fev}$, están en metros.

de $\sigma_{x'} = 0.03$ m y $\sigma_z = 0.18$ m. La Tabla 3.3 resume los resultados del experimento así como también los encontrados en Lagos et al. (2014).

Encontramos que DE localiza con éxito el evento sintético para todos los experimentos. Destacamos que DE localiza el evento con menos evaluaciones que PSO, pero necesita más evaluaciones que VFSA. Es importante enfatizar que la incertidumbre asociada a ambas componentes es significativamente menor para DE que para PSO y VFSA cuando se usa este conjunto de datos. Respecto a la componente z, se observa que PSO y VFSA obtienen mejores resultados que aquellos correspondientes a los datos teóricos, aunque las dispersiones son mayores que las obtenidos por DE en dos órdenes de magnitud.

Los errores mostrados en la Tabla 3.3 para la localización de los eventos se calculan mediante:

$$\Delta r = \left[(\bar{x}' - x'_s)^2 + (\bar{z} - z_s)^2\right]^{\frac{1}{2}}. \tag{3.10}$$

Estos valores indican que los tres métodos localizan los eventos con una precisión similar. Además, las magnitudes de las mismas se mantiene dentro de los requerimientos esperados

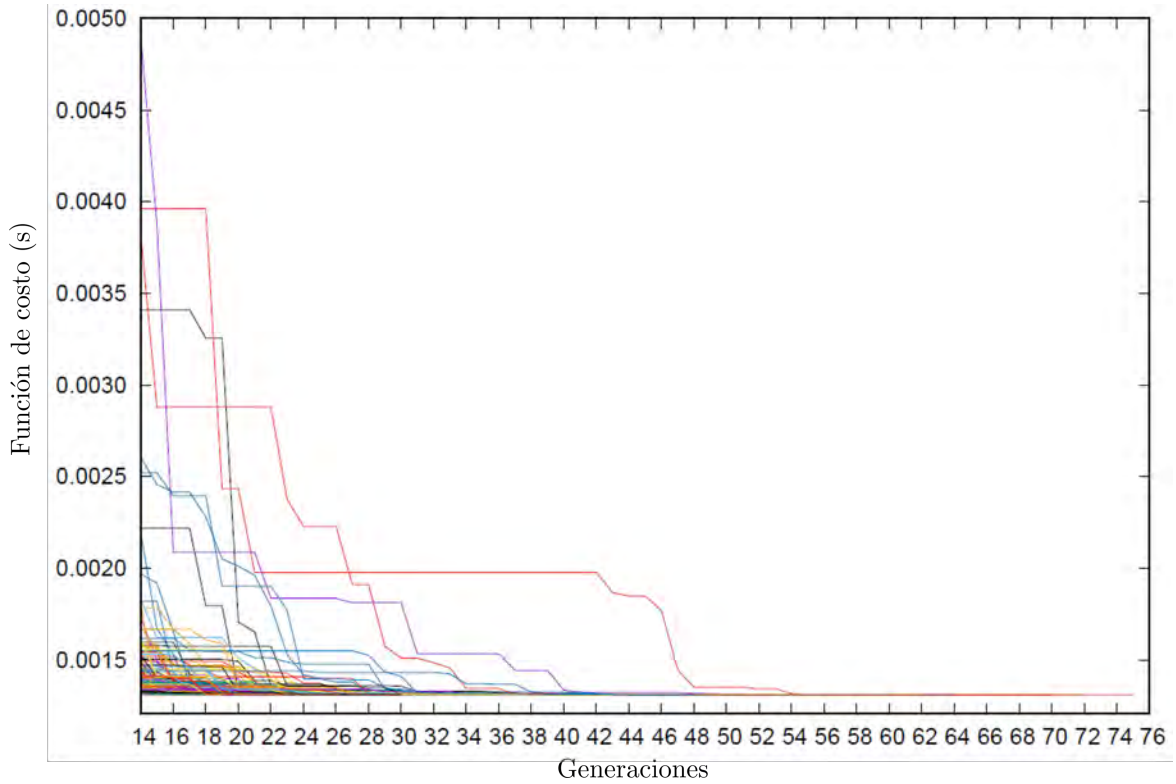


Figura 3.5: Curvas de convergencia de DE para 200 experimentos con un valor de tolerancia de 1.3×10^{-3} s.

en un monitoreo microsísmico estándar

3.5. Conclusiones

Aplicamos la técnica de evolución diferencial (DE) para resolver el problema geofísico de la ubicación de eventos microsísmicos. Para evaluar el rendimiento del método, asumimos un modelo compuesto por un medio homogéneo e isótropo con un solo pozo monitor vertical. Este esquema define la función de costo a ser minimizada. Las diferencias de tiempos de arribo teóricos y sintéticos son picadas mediante el método de Allen modificado (MAM, Sabbione and Velis (2010)) y constituyen los datos para la inversión estocástica. El proceso de localización se repite 200 veces a fin de realizar un análisis estadístico. Además, los resultados se comparan con los encontrados en Lagos et al. (2014) y se realiza un análisis detallado utilizando dos niveles de tolerancia.

Se muestra que el método DE se desempeña con éxito en todos los experimentos con una precisión aceptable. El número de evaluaciones de funciones de costo necesarias para alcanzar la solución es menor que los requeridos por PSO, aunque más alto que las requeridas por el algoritmo VFSA. Para todos los métodos y tolerancias, la dispersión en la

dirección vertical es más alta que en la dirección horizontal. Cuando se utilizan las diferencias teóricas de arribos, la dispersión en la coordenada horizontal es comparable a los resultados mostrados en Lagos et al. (2014) para ambos valores de tolerancia, mientras que la dispersión en la dirección vertical disminuye considerablemente para la tolerancia más pequeña. Con respecto a las pruebas realizadas con los tiempos de arribo seleccionados automáticamente sobre datos sintéticos, la dispersión en la posición de los eventos es considerablemente menor que la obtenida con VFSA o PSO, tanto para la coordenada x' como para z.

Capítulo 4

Inversión de tensor momento mediante redes neuronales

En este capítulo se presenta el problema de la inversión de tensor momento para datos microsísmicos y proponemos el uso de una red neuronal profunda (DNN, por sus siglas en inglés) para su resolución. Con el objetivo de testear la red, generamos datos sintéticos en diferentes modelos 3D: un medio isotrópico homogéneo, un medio anisotrópico homogéneo y, por último, un medio compuesto por dos capas planas y horizontales* con anisotropía VTI. Este tipo de anisotropía refiere a medios transversalmente isótropos respecto del eje vertical, donde las velocidades de propagación varían únicamente en la dirección de dicho eje. Para la geometría de monitoreo, elegimos un escenario realístico formado por dos pozos monitores. También aplicamos la misma red neuronal en una geometría de tipo *single-well* desviado. Como resultado mostramos que bajo estas geometrías, una red neuronal, como la descripta más adelante, puede recuperar exitosamente los 6 elementos independientes del tensor momento. Además, realizamos un análisis estadístico de los coeficientes de correlación y los errores relativos calculados a partir de los resultados obtenidos y los valores deseados, y demostramos que el tensor momento puede estimarse de forma precisa con una DNN como la propuesta, convirtiendo estas técnicas en una alternativa confiable a otros tipos de métodos más tradicionales de MTI.

4.1. Introducción

La inversión de tensor momento (MTI, por sus siglas en inglés) se puede llevar a cabo mediante varias técnicas cuyo propósito es recuperar el mecanismo focal de una dada fuente sísmica a partir de los registros sísmicos obtenidos en receptores. En microsísmica, la

*Por simplicidad, se eligió un modelo de dos capas, pero la aplicación de una red neuronal como la propuesta en este capítulo puede extenderse a un medio de N capas.

obtención de este mecanismo es altamente deseable ya que arroja luz sobre la orientación de las fracturas generadas por los diferentes eventos microsísmicos, así como también sobre las direcciones de deslizamiento de esta misma fractura (Baig and Urbancic, 2010). Conocer este dato permite un mejor entendimiento sobre el comportamiento geomecánico de un reservorio (Grechka and Heigl, 2017). La aplicación exitosa y precisa de cualquier técnica de MTI depende de muchos factores, entre los cuales se destacan dos en particular. El primer factor está relacionado a las geometrías de adquisición, usualmente caracterizadas por ángulos de apertura "bajos" (Nolen-Hoeksema and Ruff, 2001; Yu et al., 2016; Vera Rodríguez et al., 2011; Grechka, 2015a; Eaton and Forouhideh, 2011). Este es el caso de un monitoreo utilizando un solo pozo, siendo más crítico cuando este último además tiene una trayectoria vertical[*]. Este tipo de geometrías derivan en un problema de MTI donde el sistema de ecuaciones a resolver presenta un alto grado de indeterminación, evidenciado por un mal condicionamiento de la matriz que representa dicho sistema. La razón por la cual se destaca del resto de los problemas relacionados al MTI es debido a que la mayoría de los monitoreos microsísmicos a nivel mundial son realizados utilizando geometrías de este tipo (por cuestiones prácticas y económicas). En este sentido, esto último también es la causa por la que el interés por diseñar técnicas capaces de realizar MTI con estas geometrías sea mayor al mostrado para casos de MTI con dos o más pozos.

El otro factor a considerar tiene que ver con que el conjunto de técnicas de MTI son altamente sensibles a ambientes con baja relación señal ruido. Esto suele ser determinante a la hora de realizar un estudio de factibilidad de MTI, ya que ambientes de baja relación S/R son comunes a la gran mayoría de los relevamientos microsísmicos, donde la señal puede llegar a mostrar amplitudes comparables (o incluso menores) al ruido[**]. Este tipo de escenario, no solo es crítico en geometrías *single-well* (un único pozo de monitoreo), sino que también puede serlo en geometrías *mutli-well* (Pei and Warpinski, 2015).

Existen diferentes cuestiones a tener en cuenta a la hora de decidir realizar un MTI en una geometría en favor de otra. Si bien es cierto que los costos de monitoreo relacionados a las geometrías con múltiples pozos de monitoreo es considerablemente mayor al utilizado en monitoreos *single-well*, y que el número de trabajos realizados bajo esta última configuración es superior a los *dual-well*, las limitaciones teóricas impuestas sobre MTI para geometrías de un solo pozo monitor la convierten en una opción poco viable en términos prácticos (ver bibliografía sugerida más adelante). Además, las condiciones geométricas que permitirían la estimación del tensor en forma completa en escenarios de tipo *single-well* no suelen ser las más usuales. Por otro lado, desde el punto de vista del modelo de velocidades, resulta difícil realizar calibraciones apropiadas de estos modelos en geologías

[*]En realidad, esto depende no solo de la trayectoria del pozo sino también de la anisotropía del medio.

[**]El rango típico de valores para un evento microsísmico oscila entre: $-4 \leq M_W \leq 2$ (Grechka and Heigl, 2017), donde M_W es la magnitud definida en Kanamori (1977).

que presentan anisotropías complejas. Si el lector está interesado en conocer más detalles sobre esta discusión, puede referirse a trabajos como Vavryčuk (2007); Du et al. (2011); Du and Warpinski (2011); Vera Rodríguez et al. (2011); Eaton and Forouhideh (2011); Yu et al. (2015a, 2016); Grechka (2015a), o Grechka (2015b), entre otros, donde se trabaja la problemática relacionada a la aplicación de MTI en geometrías *single-well*.

Al margen de los desafíos técnicos arriba mencionados, el problema de MTI suele plantearse, en general, como un problema de tipo "inverso", donde primero se define una función de costo y luego se encuentra una solución resolviendo un sistema de ecuaciones. Si bien este enfoque suele ser el más utilizado (Vavryčuk and Kühn, 2012; Grechka, 2015a; Pei and Warpinski, 2015) también existen otras estrategias menos tradicionales, tales como Vavryčuk (2001), Hardebeck and Shearer (2003) o Zhu et al. (2015).

Redes neuronales en microsísmica

Recientemente, con el surgimiento del aprendizaje automático (*machine learning*: ML) y la inteligencia artificial (AI, por sus siglas en inglés), han habido muchos avances en resolver los problemas inversos típicos de la microsísmica, como son la detección de eventos microsísmicos (Akram et al., 2017; Ross et al., 2018; Binder and Tura, 2020; Chen, 2020; Qu et al., 2020), su localización (Huang et al., 2018) y el filtrado de ruidos o *denoising* (Zhao and Gross, 2017; Zhu et al., 2019; Zhou and Wu, 2020). Sin embargo, solo han habido un puñado de trabajos dedicados a resolver el problema de MTI mediante la utilización de ML/AI. Entre la bibliografía, se destaca el trabajo de Ovcharenko et al. (2018), donde se utilizan redes neuronales artificiales (*artificial*-NN: ANN) para realizar MTI sobre un dato microsísmico sintético de 3-componentes utilizando una geometría de un pozo vertical y un modelo homogéneo de una sola capa. Los autores de este trabajo analizan el impacto de utilizar diferentes diseños de redes y algoritmos de optimización sobre la exactitud de sus predicciones. Más recientemente, Wamriew et al. (2020) proponen utilizar una red neuronal de tipo convolucional (CNN) para resolver el problema de MTI en una geometría de pozo vertical en un medio de tipo anisotrópico VTI. Mediante la aplicación de un primer set de capas de tipo convolucionales, los autores evitan realizar una gran cantidad de pre-procesamiento sobre el dato de entrada. Esto se debe a que explotan una de las principales ventajas que ofrecen las redes de tipo CNN, que es la "extracción de características" o *feature extraction*.

En forma sencilla, para el caso de una red neuronal simple (no CNN) y en pos de lograr buenos resultados, el usuario debe primero invertir una considerable cantidad de tiempo y esfuerzo en estudiar o diseñar qué tipo de entrada es la mejor para resolver un dado problema. Para estos casos, las preguntas que el usuario suele hacerse son: ¿cuál es el mejor conjunto de parámetros a partir de los cuales debo construir mis entradas $\mathbf{x}_i$?, y/o ¿cuál es el conjunto de parámetros que contiene mayor información útil?. En cambio,

para el caso de las redes de tipo CNN, será trabajo de las primeras capas convolucionales el distinguir qué tipo de información puede extraerse del dato crudo para minimizar una dada función de costo. Se dice que el entrenamiento de este tipo de capas genera un "mapa de activación" a partir del dato crudo. En este trabajo, los autores entrenan una CNN para resolver los parámetros de rumbo, inclinación, buzamiento y magnitud momento para un dado evento microsísmico. También, Binder (2018) entrena una NN para resolver MTI en una microsísmica de superficie. En este trabajo, los autores invierten las 6 componentes del tensor momento utilizando datos sintéticos para una geometría de adquisición de tipo estrella.

Este capítulo estará mayormente dedicado al diseño y aplicación de una DNN para resolver el problema de MTI en un escenario de tipo *dual-well*. Sin embargo, y por completitud, se dedica una sección a los resultados obtenidos de aplicar este tipo de redes a una geometría de tipo *single-well* desviado.

Para testear la red se simula un monitoreo realista con dos pozos monitores verticales (o uno desviado) sumergidos en dos medios anisotrópicos VTI diferentes: semi-espacio infinito y de capas planas y horizontales. Para el entrenamiento de esta DNN y en todos los casos, generamos un gran número de eventos microsísmicos y proveemos un análisis estadístico de las soluciones predichas por la red. Los resultados muestran que la DNN propuesta puede resolver el problema de MTI con un alto grado de precisión (caso *dual-well*), a excepción de aquellos eventos localizados en el plano que contiene a los dos pozos monitores. Para el caso *single-well*, si bien los tensores momento predichos por la red son similares a los tensores reales, los errores son mayores a los calculados para un modelo *dual-well*.

4.2. Teoría y método

Inversión de tensor momento

La inversión de tensor momento (o MTI) puede plantearse, típicamente, como un problema de inversión lineal. En este caso, la ecuación matricial sobre la que debe trabajarse se escribe como:

$$\mathbf{d} = \hat{\mathbf{G}}\,\mathbf{m}, \tag{4.1}$$

donde $\mathbf{m} = [m_{11}, m_{22}, m_{33}, m_{23}, m_{13}, m_{12}]^T$ es un vector columna con las 6 componentes independientes del tensor momento $\mathbf{M}$ y la matriz $\hat{\mathbf{G}}$ se construye a partir de las componentes de "campo lejano" de las funciones de Green correspondientes al medio sobre el cual se está trabajando. Por último, $\mathbf{d}$ es la señal que arriba a cada uno de los receptores (ver: Vavryčuk and Kühn (2012)). Si el lector lo desea, puede encontrar un desarrollo más detallado de esta ecuación y sus componentes en el Apéndice A, o a lo largo de la bibliografía sugerida, como por ejemplo Leaney (2014).

En general, para resolver el sistema 4.1 se suele definir la función de costo:

$$J = ||\mathbf{d} - \hat{\mathbf{G}}\,\mathbf{m}||_2^2. \tag{4.2}$$

Así, se obtiene un problema de optimización cuya solución puede encontrarse de forma cerrada utilizando mínimos cuadrados tradicionales, por lo que

$$\mathbf{m} = (\hat{\mathbf{G}}^T\,\hat{\mathbf{G}})^{-1}\hat{\mathbf{G}}^T\mathbf{d}, \tag{4.3}$$

o mediante *damped-least squares* (Pei and Warpinski, 2015). El desafío principal a la hora de obtener esta solución se encuentra en el cálculo y la naturaleza de la matriz $\hat{\mathbf{G}}$. Por un lado, la construcción de esta matriz implica el cálculo de las funciones de Green, que dependen de las propiedades del medio en el cual se propaga la señal del evento sísmico. Utilizar esta estrategia para invertir el tensor momento de una dada cantidad de eventos implicaría calcular la matriz $\hat{\mathbf{G}}$ para cada uno de ellos. Además, debido a que esta matriz depende exclusivamente de las propiedades del medio que atraviesa el rayo sísmico fuente-receptor, es claro que la dificultad para encontrar los elementos que componen esta matriz aumenta con la complejidad del medio. En este sentido, la utilización de una "red neuronal" permitiría independizarse del cálculo de la matriz $\hat{\mathbf{G}}$. Esto es, una vez entrenada dicha red con un conjunto dado de eventos y sus tensores, la misma podría entregar el tensor momento correspondiente a un evento microsísmico únicamente a partir de sus amplitudes registradas por los geófonos.

Por otro lado, la calidad de la solución encontrada para el sistema de ecuaciones 4.1, mediante alguno de los métodos tradicionales, está relacionada al número de condición de la matriz $\hat{\mathbf{G}}^T\hat{\mathbf{G}}$. Debido a factores geométricos relacionados al monitoreo (Vavryčuk, 2007; Du et al., 2011; Du and Warpinski, 2011; Vera Rodríguez et al., 2011; Eaton and Forouhideh, 2011) este número con frecuencia es alto, afectando severamente la calidad de la inversión. Nuevamente, las "redes neuronales" vuelven a presentarse como un método que permitiría evitar el trabajo necesario para encontrar el parámetro de amortiguamiento óptimo a fin de resolver el sistema mal condicionado.

Este capítulo se propone investigar el potencial de las "redes neuronales" como método alternativo para atacar el problema de MTI, así como también sus fortalezas y desventajas frente a las dificultades anteriormente mencionadas.

Geometría de monitoreo y datos sintéticos

Para el monitoreo de la fractura hidráulica se considera una geometría con dos pozos verticales (ver Figura 4.1). Cada uno de los pozos contiene 5 geófonos de 3 componentes espaciados 30.5 m. Los receptores más someros están posicionados a $z_R = 200$ m de profundidad en ambas perforaciones. Además, la boca de ambos pozos están localizadas

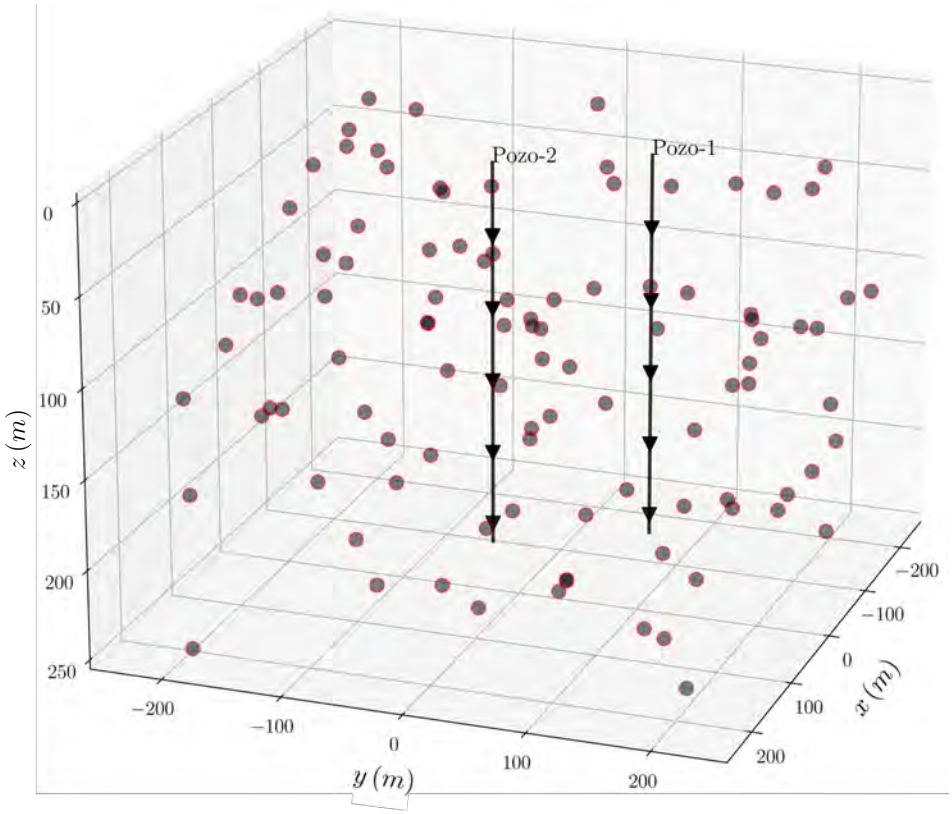


Figura 4.1: Geometría de monitoreo. Dos pozos verticales con 5 geófonos cada uno (triángulos negros). Se muestran, a modo únicamente ilustrativo, un total de 100 eventos distribuidos de forma aleatoria (círculos grises).

en las coordenadas superficiales $(x, y) = (0, 110)$ m y $(60, 0)$ m, respectivamente, lo que corresponde a una separación entre bocas de $\Delta \backsimeq 125.30$ m.

Para la generación del dato sintético simulamos una nube de eventos en la vecindad de los pozos y cuyas posiciones son generadas con una distribución espacial uniforme. Los mismos están contenidos dentro de un cubo de $500 \times 500 \times 250$ m^3. Además, los tensores momento para cada uno de estos eventos se arman generando 6 valores diferentes con una distribución aleatoria uniforme en $[-1, 1]$. Con el propósito de estudiar el desempeño de la red bajo diferentes condiciones, se realizan experimentos considerando 3 modelos de velocidades diferentes. El primero es un modelo isotrópico homogéneo y semi-infinito, mientras que el segundo es un medio anisotrópico VTI de anisotropía débil. Por último, se modelan eventos en un medio anisotrópico de dos capas planas y horizontales. Los valores de las densidades, velocidades y parámetros de anisotropía utilizados para cada uno de estos modelos se resumen en la Tabla 4.1.

Para cualquiera de los modelos, los datos son generados utilizando el campo de ondas lejano (Grechka and Heigl, 2017) correspondiente a la ecuación A.27. Cabe recordar que

Modelo	Tipo	Capas	z (km)	V_P (km/s)	V_S (km/s)	α	δ	γ
1	isótropo	1	-	3.5	2.5	-	-	-
2	VTI	1	-	3.5	2.4	0.10	0.08	0.09
3	VTI	2	0.125	3.5	2.4	0.10	0.045	0.031
			0.500	4.5	3.4	0.11	-0.032	0.029

Tabla 4.1: Detalle de los 3 modelos considerados en los experimentos principales. En todos los casos se consideran medios homogéneos con densidades $\rho = 2.7$ kg/m^3.

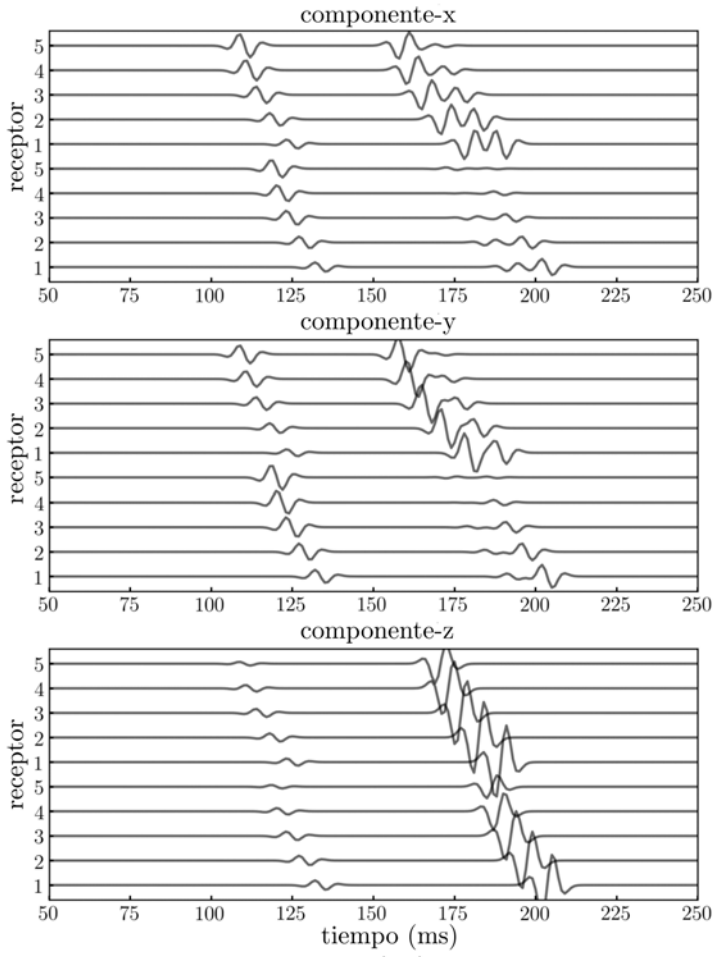


Figura 4.2: Señal microsísmica para un evento de posición y tensor momento aleatorios. Se muestran 3 paneles diferentes, correspondientes a las 3 componentes del registro. Además, cada panel muestra las 5 trazas de cada uno de los pozos. Tanto la componente x (primer fila) como la componente y (segunda fila) muestran el fenómeno de partición para la onda S, causado por considerar un medio con anisotropía débil VTI.

la misma se reduce a la ecuación A.21 cuando no se considera anisotropía.

La Figura 4.2 muestra una señal microsísmica típica. La misma fue generada por un evento de posición y tensor momento aleatorios y fue registrada con los 5 receptores 3C de cada uno de los 2 pozos verticales de nuestro arreglo. Se puede identificar el arribo de la onda P en las 3 componentes y para cada receptor, seguido por un arribo que corresponde a la superposición de las dos ondas de cizalla. En este caso, dicha superposición de fases puede deberse, principalmente, a una distancia "fuente-receptor" pequeña, una anisotropía muy débil, o una combinación de ambas.

Redes neuronales de tipo profunda (DNN)

En el caso más general, una red neuronal (NN) es un conjunto de capas apiladas cuyos elementos están conectados por alguna operación matemática específica. Los elementos de estas capas reciben, comúnmente, el nombre de "neuronas", aunque es más apropiado referirnos a ellos como "unidades"*. Históricamente, las primeras redes neuronales consistían en una única capa de estas unidades, pero hoy en día es común trabajar con NNs de varias decenas o centenas de capas, dependiendo de las exigencias del problema a resolver. Así, a aquellas NNs que contienen un gran número de capas apiladas se les anexa el término "profundas", y por ello, se las conoce como "redes neuronales profundas" o *deep neural network* (DNN, por sus siglas en inglés) (Géron, 2019). En este tipo de arquitecturas, la primer capa suele denominarse "capa de entrada" o *input layer* (IL, por sus siglas en inglés) y es la encargada de recibir el dato. Como esta capa no efectúa ningún cambio efectivo sobre el dato de entrada (su comportamiento es análogo a una función identidad), es común que la bibliografía intercambie esta capa con el mismo dato de entrada.

Sin pérdida de generalidad, consideremos $\mathbf{x} = \{x_i\}$, $i = 1, \cdots, r$, como un elemento del conjunto de entradas. Así definido, cada uno de estos elementos $\mathbf{x}$ se denomina como "instancia" del dato de entrada y puede pensarse como un vector de dimensión r. Por otro lado, los elementos individuales que componen a este vector son llamados "características" o *features*.

Cuando un elemento $\mathbf{x}$ es ingresado a una NN, cada unidad de la capa de entradas distribuye sus elementos sobre todas las unidades de la siguiente capa, a la que denotaremos con el índice $L = 1$ (notar que la capa de entradas sería entonces $L = 0$). A continuación, cada unidad j de esta capa interactúa con estas cantidades para producir una salida escalar o_j^1. La salida es pasada y procesada siguiendo esta lógica a través de todas las capas de la red hasta llegar a la última, denominada "capa de salida" o *output layer* (OL, por sus siglas en inglés), donde sus unidades producen un resultado final o_j^{OL}. Estas salidas serán,

*El término "unidades" deriva de las siglas LTU, cuyo significado en inglés refiere a *Linear Threshold Unit* (unidad lógica de umbral).

finalmente, los elementos de la predicción final del modelo, $\hat{\mathbf{y}}$. Obviamente, este proceso es repetido para cada una de las instancias del dato de entrada.

En general, cuando se alimenta una unidad j con una entrada $\mathbf{x}$, la misma es sometida a un proceso que involucra dos pasos. El primero es la combinación lineal de todos los *features* x_i con un set de pesos w_{ij} y un parámetro de *bias* b_j, correspondiente a esa unidad particular. El resultado de esta operación será una salida z_j. El segundo paso consiste en aplicar una función no-lineal sobre esta cantidad para producir una salida o_j. Esta función se denomina comúnmente "función de activación" y es la que permite a la NN representar las relaciones no-lineales que pueden existir entre el conjunto de datos de entrada y su salida deseada. Sin estas funciones, la red solo sería una combinación lineal de operaciones, y por lo tanto, sería únicamente capaz de producir mapeos lineales entre las entradas y sus salidas.

Finalmente, el conjunto de parámetros (pesos, *bias*, etc) para toda la red compone un "modelo" matemático y sus valores deben ser ajustados de manera óptima para producir una salida deseada final $\hat{\mathbf{y}}$. La manera de realizar este ajuste es mediante lo que se conoce como "entrenamiento". La idea principal detrás del entrenamiento está condensada en el algoritmo de *back-propagation* (Rumelhart et al., 1985; Hecht-Nielsen, 1992), que se vale de los gradientes de la función de costo respecto de todos los parámetros de la red para luego actualizarlos iterativamente a fin de reducir el error entre la predicción del modelo y la salida deseada. Para una descripción detallada de las NNs, el lector puede referirse al Apéndice B.

Configuración del dato de entrenamiento

Según lo visto en el Apéndice B, los dos tipos de aprendizaje automático más comunes son los de tipo "supervisado" y "no supervisado". En este capítulo, se describe el entrenamiento de una red para resolver el problema de MTI mediante redes neuronales como uno de tipo supervisado. Dentro de esta clasificación, diremos incluso que nuestro problema será del tipo regresivo y no-lineal. Esto es, a partir de un conjunto de observables extraído de la señal producida por un evento microsísmico, nos interesa encontrar un modelo (la red) que permita conocer el mecanismo focal que produjo dicha señal. Así, cada instancia del dato de entrenamiento estará formada por un par de datos $(\mathbf{x}_i, \mathbf{y}_i)$, donde cada $\mathbf{x}_i$ será un dato de entrada e $\mathbf{y}_i$ será la respectiva salida deseada (ver Apéndice B, ecuación B.10). Es evidente que si nos interesa conocer el tensor momento $\mathbf{m}_i$ que describe el mecanismo focal de un evento i, el dato de salida deseado $\mathbf{y}_i$ será:

$$\mathbf{y}_i = \mathbf{m}_i. \tag{4.4}$$

Cabe mencionar que no se impone ninguna restricción *a priori* sobre el mecanismo focal de los datos de salida para ninguna de las instancias del dato de entrenamiento (y testeo). Así,

los 6 valores independientes de $\mathbf{y}_i$ son extraídos de una distribución aleatoria uniforme.

Por otra parte, la construcción del dato de entrada $\mathbf{x}_i$ (el observable) no es tan evidente. Claramente se debe ensamblar este dato de forma tal que transporte información relevante y suficiente para resolver una tarea de MTI.

En primer lugar, se sabe que la solución de MTI para un cierto evento sísmico está íntegramente relacionada con las amplitudes relativas de la señal observada y con la posición relativa entre la fuente y los geófonos desplegados en el arreglo de receptores. En segundo lugar, y en menor medida, depende también de la naturaleza del medio por el cual se propaga la señal. Por esta razón, nuestro dato de entrada estará constituido por las 3 coordenadas cartesianas espaciales del evento y las amplitudes máximas (con su respectivo signo) medidas en las 3 componentes para cada una de las fases de onda involucradas y para cada uno de los geófonos. Por ejemplo, si se realiza un experimento con 2 pozos monitores, cada uno con 5 geófonos 3C y considerando un medio homogéneo, se tendrá un total de 30 componentes (3C×10 geófonos). Así, el total de amplitudes registradas será de 60, correspondientes a 30 amplitudes de fase P y otras 30 de la fase S. El dato de entrada para un evento i tendría la forma:

$$\mathbf{x}_i = [x, y, z, P_1, P_2, \ldots, P_{30}, S_1, S_2, \ldots, S_{30}]_i, \tag{4.5}$$

donde x, y, z son las coordenadas conocidas del evento y P_j y S_j con $j = 1, \ldots, 30$ son las amplitudes medidas de las fases P y S, respectivamente. En cambio, si se mantiene la misma geometría, pero se considera un medio VTI (de anisotropía débil), se obtendrán un total de 30 amplitudes para cada una de las tres fases P, Sh y Sv.

Desde el punto de vista práctico, las amplitudes se extraen como el máximo valor absoluto dentro de una ventana temporal de ancho fijo que se posiciona comenzando en los tiempos de arribo estimado para cada una de las fases. En este punto, se considera que dichos tiempos son un dato conocido que puede obtenerse previamente mediante algún método de localización o detección. Para la ventana, debe considerarse un ancho que permita contener la fase de manera completa, pero a su vez, lo suficientemente pequeño como para que las ventanas no puedan contener, total o parcialmente, dos fases. Claramente, esto se vuelve difícil de cumplir para modelos de velocidad con $V_{sh} \sim V_{sv}$, o para aquellos eventos muy cercanos a los receptores.

En cualquier caso, el conjunto de amplitudes para cada evento es normalizado tal que su máximo (o mínimo) sea 1 (o -1). El propósito de esta normalización está relacionado principalmente con el entrenamiento de las redes. Por diversas razones, las redes neuronales entrenadas con entradas cuyos elementos tengan valores en escalas muy diferentes suelen tener malos desempeños (Géron, 2019). Por ello, y siempre que sea posible, es aconsejable normalizar los datos de entrada y salida. En nuestro caso, se elige no normalizar los valores que corresponden a las coordenada del hipocentro. Así evitamos que dos eventos que estén

en coordenadas diferentes como $(200.0, 100.0, 100.0)$ m y $(20.0, 10.0, 10.0)$ m sean mapeados a la misma coordenada $(1.0, 0.5, 0.5)$ m. Es importante notar que la normalización de las amplitudes observadas para cada evento remueve toda información relacionada a la magnitud de dicho evento. Al remover esta información, se permite al modelo concentrarse en predecir el mecanismo focal de los eventos y no en la predicción de patrones asociados a la distancia "fuente-receptor" o la magnitud momento de los mismos. Al hacer esto, se asume que para un dado evento, su mecanismo focal puede inferirse unívocamente a partir de los patrones presentes en las amplitudes relativas del conjunto de fases que son observadas por el arreglo de geófonos.

Finalmente, cabe destacar que nuestro dato $\mathbf{x}_i$ no transporta información del modelo. Esto es así debido a que la red será entrenada con una nube de eventos pertenecientes a un mismo medio. En estas circunstancias, incorporar información del medio (las velocidades de fases, anisotropía, etc), común a todos las instancias del dato, no aportarían nada relevante al entrenamiento.

Arquitectura del modelo propuesto

La Figura 4.3 es una representación de la arquitectura propuesta para nuestra DNN. La misma está compuesta por 6 capas completamente conectadas (FC) y alimentadas hacia adelante. La primera capa, denotada en la figura como L_1, tiene un total de 53 neuronas cuya cantidad de pesos está definida por el número de elementos del dato de entrada. Si consideramos un medio anisotrópico, cada una de ellas tendrá un total de 93 pesos, compuestos por 90 amplitudes y 3 coordenadas (más un *bias*). Es importante mencionar que el número de neuronas utilizado en esta primer capa no está relacionado con la dimensión del dato ni con las características del medio. Así, su elección es arbitraria y se eligió luego de numerosos ensayos de tipo "prueba y error", quedándonos con el número que arrojó mejores resultados*. Como se verá más adelante, el mismo criterio fue utilizado para la arquitectura general de toda la red.

Así definida, esta capa producirá 53 salidas (una por cada neurona de la capa de entrada). El diseño de la red continua de forma tal que la dimensión de las salidas de cada una de las capas se va reduciendo paulatinamente hasta llegar a la capa final, que cuenta con 6 neuronas, produciendo así las 6 salidas que se necesitan para construir el tensor momento. Nuevamente, las dimensiones de las capas deben ser arregladas de forma tal que el número de pesos de cada neurona coincida con el número de salidas producidas por la capa anterior. Esta forma "piramidal" es un diseño típico usado para tareas de tipo regresivas, como es este caso (Géron, 2019). Para el caso isotrópico, la primer capa tendrá

*Para la elección de las dimensiones finales de la red se tuvo en cuenta que a mayor cantidad de neuronas y pesos, mayor es el riesgo de *overfitting* (ver Apéndice B), por lo que se trató siempre de minimizar el tamaño sin perjudicar su desempeño.

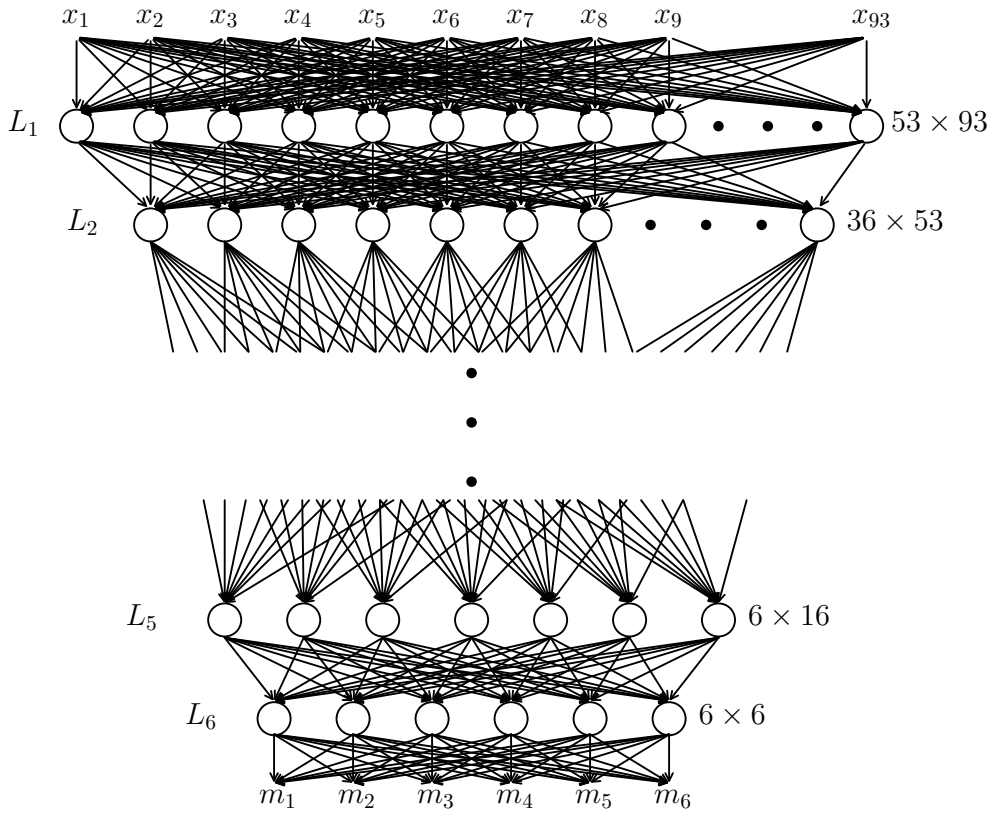


Figura 4.3: Representación de la arquitectura de la NN. La red tiene un total de 6 capas cuyas dimensiones asemejan a una forma de tipo piramidal. En esta figura, se muestra una entrada correspondiente a un medio anisotrópico, por lo que el dato tiene 93 elementos. La primera capa cuenta con 53 neuronas de 93 pesos cada una, dando un total de $53 \times 93 + 53 = 4982$ parámetros entrenables. La última capa, de dimensión 6×6, produce un vector final de 6 elementos. Este vector de salida será considerado como la predicción del tensor momento para cada entrada.

la misma cantidad de elementos, pero cada uno de ellos estará formado por 63 pesos, compuestos por 60 amplitudes y 3 coordenadas (más el término de *bias*).

Finalmente, las funciones de activación utilizadas para cada una de las capas es la tangente hiperbólica: $tanh(x)$. La elección de esta función se fundamenta en que la misma tiene una imagen comprendida en el intervalo $(-1, 1)$, lo cual asegura que la salida (y en general, la salida de cada una de las capas) esté contenida también en el mismo intervalo. Dicho esto, es evidente que la red estará entrenada (y testeada) con un dato de salida correspondiente a un mecanismo focal normalizado. Esto implica, indirectamente, que nuestra red será capaz de reconstruir el mecanismo, pero será ciega a la magnitud momento del evento.

Entrenamiento y testeo de la red

Con el propósito de entrenar y testear la red, generamos un dato compuesto de 9×10^5 eventos con las características ya mencionadas. De este grupo principal, se separa un 90 % del dato para la etapa de entrenamiento y se retiene el restante 10 % para la etapa del testeo. Esto corresponde a 810000 y 90000 eventos, respectivamente. Típicamente, cuando se diseña una arquitectura de NN, es una práctica común dividir el dato de entrenamiento en dos partes. La primera de ellas, conteniendo la mayor cantidad de instancias, será el dato de entrenamiento propiamente dicho. Este dato se usará para optimizar la red mediante *back-propagation*. El dato restante, típicamente más pequeño, es utilizado para evaluar el desempeño del modelo entrenado. Si el mismo no se comporta como es deseado, se procede a ajustar los parámetros de entrenamiento (hiperparámetros) en función de lo observado, tales como la tasa de aprendizaje o el tamaño de la red, entre otros. Este último conjunto de datos se conoce como "set de validación" (o "set de desarrollo") (Géron, 2019). Debido a que nuestro objetivo principal es investigar la capacidad de una NN para resolver una tarea de MTI, y no la de encontrar la combinación de hiperparámetros que mejor logre esta tarea, decidimos no apartar una porción del dato de entrenamiento para componer un set de validación. Dicho esto, el trabajo de ajustar estos hiperparámetros fue realizado con detenimiento pero no se describe en la Tesis.

A continuación, se realiza un "batcheo" de los datos de entrenamiento y testeo agrupándolos en grupos de 1000 instancias. Para la función de costo, se utiliza el error cuadrático medio relativo (porcentual), definido como:

$$\bar{E}_r(\mathbf{y}, \hat{\mathbf{y}}) = \frac{1}{M} \sum_{k=1}^{M} \frac{|\mathbf{y}_k - \hat{\mathbf{y}}_k|}{|\mathbf{y}_k|} \times 100\,\%, \tag{4.6}$$

donde M es el número de instancias en el dato de entrenamiento. Aquí, $\bar{E}_r(\mathbf{y}, \hat{\mathbf{y}})$ es evaluada en cada una de las iteraciones del entrenamiento y su función es la de medir la discrepancia entre el dato predicho por la red, $\hat{\mathbf{y}}_k$, y el dato deseado, $\mathbf{y}_k$. Para el entrenamiento se utilizan

900 épocas*. El algoritmo de optimización elegido es ADAM (Kingma and Ba, 2014) (con sus parámetros estándar) y una tasa de aprendizaje de $\eta = 0.001$. Además, se aplica un decaimiento sobre η con un factor de 1.1 cada 15 iteraciones donde no se registraron mejoras en la función de costo.

Adicionalmente, a medida que avanzan las épocas nos interesa saber cómo se desempeña el modelo sobre el dato de testeo. Para ello, en cada época evaluamos la función $\bar{E}_r(\mathbf{y}, \hat{\mathbf{y}})$ y una función de certeza (*accuracy*) sobre el dato de testeo. Para este trabajo en particular, dicha función fue definida como el coeficiente de correlación medio, cuya expresión tiene la forma:

$$\bar{\rho}(\mathbf{y}, \hat{\mathbf{y}}) = \frac{1}{M} \sum_{k=1}^{M} \frac{\mathbf{y}_k \cdot \hat{\mathbf{y}}_k}{|\mathbf{y}_k||\hat{\mathbf{y}}_k|}, \tag{4.7}$$

donde "$\cdot$" es el producto vectorial escalar. El diseño de esta función pone en evidencia que para evaluar el desempeño de una red en el dato de testeo no es necesario utilizar la misma métrica que se utilizó para entrenar la red. Es decir, la métrica de testeo no tiene porqué ser la misma que la métrica de entrenamiento. Esto último es tratado a lo largo y ancho de la bibliografía específica del aprendizaje automático, y por lo tanto, evitaremos entrar en detalles sobre este asunto. Por simplicidad, nos limitaremos a decir que mientras que la función de certeza es aquella que nos interesa para evaluar el desempeño de la red sobre el dato de campo (de testeo), la función de costo es una mera herramienta de entrenamiento. Una diferencia significativa entre ambas funciones es, por ejemplo, que mientras que la función de error debe ser diferenciable para poder aplicar *back-propagation*, la función de certeza no debe serlo, por lo que se tiene mayor libertad en su elección.
En particular, la función $\bar{\rho}(\mathbf{y}, \hat{\mathbf{y}}) \in [-1, 1]$ es ideal para evaluar que tan buena es una predicción $\hat{\mathbf{y}}$ respecto de $\mathbf{y}$, ya que nos permite tener una medida del paralelismo entre estos dos vectores. A su vez, esta medida de paralelismo es independiente de la magnitud (módulo) de los vectores involucrados, por lo que sirve exactamente a nuestro propósito.

Finalmente, como criterios de corte para el entrenamiento, se considera que el mismo termina cuando se llega al máximo número de épocas o cuando la certeza, medida sobre el dato de testeo, llega al valor 0.99.

4.3. Resultados

En esta sección se muestran los resultados arrojados por la NN entrenada para predecir tensores momento (TMs) de eventos generados utilizando los 3 modelos de la Tabla 4.1 y

*Recordar que una época consiste en correr el algoritmo de *back-propagation* para todos los *batches* del dato de entrenamiento, mientras que una iteración corresponde a efectuar la operación de descenso de gradiente para un único *batch*.

un caso *single-well*. Por claridad, los resultados individuales correspondientes a cada uno de los modelos son separados en sub-secciones diferentes.

Modelo 1: dos pozos inmersos en un medio isotrópico semi-infinito

Los resultados de la predicción considerando las 9×10^4 instancias apartadas para el testeo se resumen en la Figura 4.4 y la Tabla 4.2. Como puede observarse, la red neuronal es capaz de predecir casi la totalidad de los eventos del set de datos de testeo con una gran certeza y errores relativos pequeños. Para tener una idea del desempeño a nivel global, se calculan los promedios de $\bar{\rho}$ y $\bar{E}_r$ dados por la ecuaciones 4.6 y 4.7, utilizando todos los eventos del set de testeo. Estos promedios arrojan valores de $\bar{\rho} \sim 0.985$ y $\bar{E}_r \sim 17.52\,\%$. En detalle, se observa que $\sim 62\,\%$ de todos los mecanismos de fractura son predichos con un coeficiente de correlación $\bar{\rho} \geq 0.99$ y un error relativo $\bar{E}_r \sim 13\,\%$ (ver Tabla 4.2). Es decir, más de dos tercios de la población de datos de testeo cumple con el criterio de corte del entrenamiento. Continuando con este análisis, un $\sim 21\,\%$ y $\sim 6.8\,\%$ del dato restante fue invertido entregando valores de correlaciones en los intervalos $0.98 \leq \bar{\rho} < 0.99$ ($\bar{E}_r \sim 18.70\,\%$) y $0.97 \leq \bar{\rho} < 0.98$ ($\bar{E}_r \sim 24.55\,\%$), respectivamente. Puede decirse que más del $90\,\%$ de los eventos fue predicho con una correlación mayor a 0.97.

Mientras que la inspección de los valores numéricos del coeficiente de correlación y el error relativo sirve como un indicador de la capacidad de la NN para predecir los tensores momento, no brinda mucha información acerca de la calidad real de los mecanismos recuperados. Para ello, es necesario visualizar los mecanismos. Una de las formas más comunes de graficar un TM es mediante las "pelotas de playa", más comúnmente conocidas como *beach-balls* (por su denominación en inglés). En general (porque puede suceder que sea una proyección de la semiesfera superior), una pelota de playa es la proyección de la semiesfera inferior y da lugar a lo que se conoce como "mecanismo focal". Esto es, una esfera imaginaria con centro en la fuente sísmica que proyecta todos los rayos sísmicos desde los posibles observadores hasta su origen. Así, la esfera puede dividirse por medio de 2 planos nodales (donde uno de ellos representa el plano de falla) generando 4 cuadrantes. Los cuadrantes sombreados en este trabajo, con color rojo o verde, representan zonas compresivas, mientras que los cuadrantes más claros o en blanco representan zonas distensivas (Shearer, 2009). En la Figura 4.5 se muestran las *beach-balls* correspondientes a los mecanismos de algunos eventos del dato de testeo y sus respectivas predicciones. Los mismos fueron seleccionados al azar de los subgrupos con $0.99 \leq \bar{\rho} \leq 1.0$ y $0.94 \leq \bar{\rho} \leq 0.95$. El primer sub-grupo fue elegido para mostrar las predicciones de mejor calidad, mientras que el segundo sub-grupo fue elegido debido a que la cantidad de mecanismos predichos con valores de $\bar{\rho} \leq 0.94$ es despreciable (menor al $3\,\%$ del total de instancias de testeo).

De la figura se desprende que las *beach-balls* predichas (rojas) son prácticamente indiscernibles de aquellas obtenidas a partir del dato real (verdes), especialmente para el caso

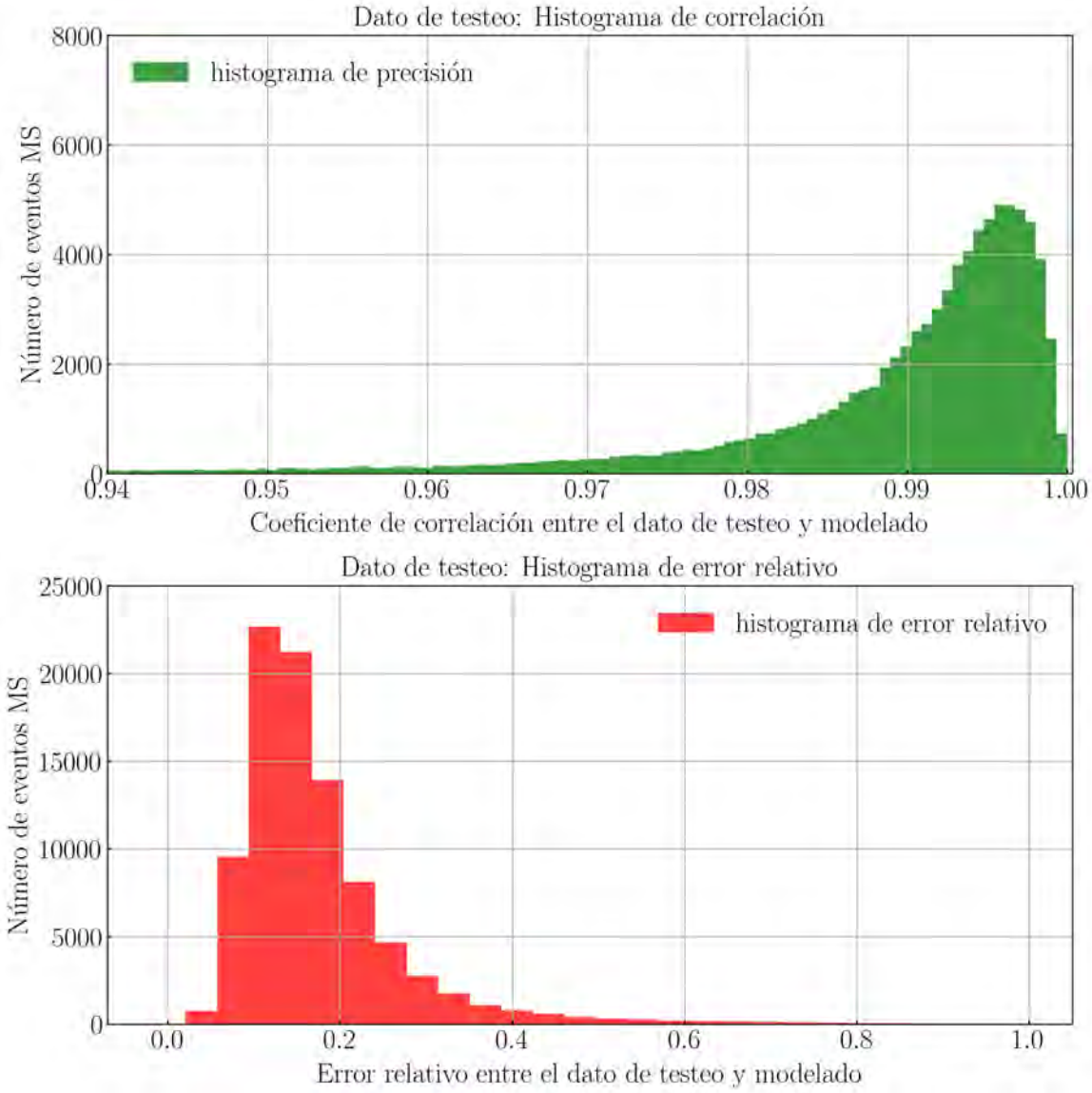


Figura 4.4: Modelo 1: Histogramas de los resultados predichos. Arriba: histograma de coeficientes de correlación. Abajo: histograma de errores relativos. En ambos casos el histograma es calculado utilizando los resultados predichos por el modelo al ser evaluado sobre el dato de testeo. Los máximos de los histogramas indican valores ~ 0.997 y $\sim 10\,\%$.

Certeza			Error relativo	
$\bar{\rho}$	Eventos %	$\bar{E}_r$ %	$\bar{E}_r$ %	Eventos %
$[0.99, 1.00]$	61.9411	12.81	$[100, \infty)$	0.088
$[0.98, 0.99)$	21.470	18.70	$[90, 100)$	0.074
$[0.97, 0.98)$	6.8444	24.55	$[80, 90)$	0.152
$[0.96, 0.97)$	2.8089	29.10	$[70, 80)$	0.279
$[0.95, 0.96)$	3.1656	33.10	$[60, 70)$	0.468
$[0.94, 0.95)$	1.7678	36.78	$[50, 60)$	0.790
$[0.84, 0.94)$	1.073	47.41	$[40, 50)$	1.652
$[0.74, 0.84)$	2.9433	69.35	$[30, 40)$	4.532
$[0.64, 0.74)$	0.5589	81.85	$[20, 30)$	17.46
$[0.54, 0.64)$	0.1633	93.92	$[10, 20)$	59.48
$[0.44, 0.54)$	0.0478	104.61	$[0, 10)$	15.01
$[0.34, 0.44)$	0.0200	290.07		
$[0.24, 0.34)$	0.0044	-		
$[0.14, 0.24)$	0.0000	-		
$[0.04, 0.14)$	0.0000	-		
$[-1.0, 0.04)$	0.0000	-		

Tabla 4.2: Modelo 1: Análisis de los histogramas de correlación y error relativo. El dato es separado en sub-grupos con intervalos irregulares que contienen información relevante a partir de ambos histogramas.

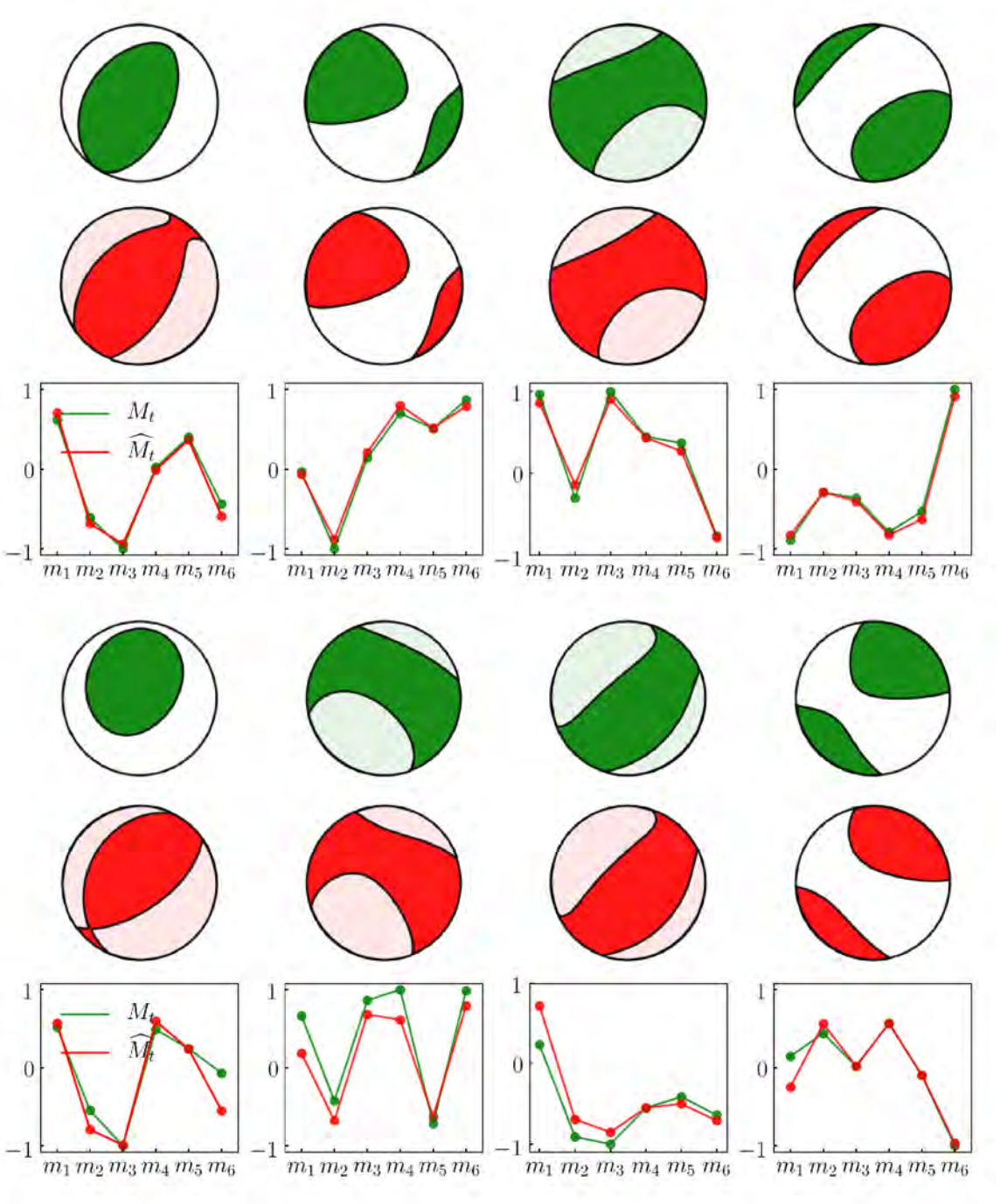


Figura 4.5: Modelo 1: Diagramas *beach-ball* calculados a partir de los TMs (predichos y su correspondiente tensor deseado). Filas 1 y 2:: *beach-balls* calculadas a partir de un mecanismo de testeo (verde, dato sintético) y su correspondiente mecanismo predicho (rojo) para 4 (uno por columna) datos pertenecientes al grupo con $0.99 \leq \bar{\rho} \leq 1.0$. Fila 3: valores numéricos para los 6 elementos independientes de cada evento. Filas 4 a 6: idem para 4 eventos del grupo con $0.94 \leq \bar{\rho} < 0.95$.

$\bar{\rho} \geq 0.99$, que agrupa más del 60 % de todos los eventos de testeo. Finalmente, la tercera y sexta filas de la Figura 4.5 muestran los 6 elementos independientes del tensor momento de testeo y su correspondiente tensor predicho para los eventos seleccionados. Como se puede observar, los valores numéricos son muy similares, incluso para el sub-grupo de menor $\bar{\rho}$.

A continuación, nos interesa conocer la distribución espacial de los errores. En este sentido, se observa una distribución espacial no-uniforme de los errores relativos, donde se evidencia que estos valores dependen de la posición relativa entre los eventos y los arreglos de receptores. Este fenómeno es ilustrado en la Figura 4.6, donde se grafica la proyección de todos los eventos de testeo sobre el plano horizontal y se colorean según su error relativo. La figura cuenta con 4 paneles que muestran diferentes rangos de error relativo.

Del análisis de esta figura se desprende que los errores relativos son mayores para las predicciones de aquellos eventos que se encuentran más próximos al plano que contiene ambos pozos verticales. Por ser una vista en planta, la figura solo nos permite observar la proyección de este plano, que se encuentra representada por una recta (roja) uniendo ambos pozos. Este comportamiento es esperado y constituye uno de los principales resultados de este trabajo. Si consideramos un medio isótropo y homogéneo, la cantidad de información provista por las amplitudes de las fases P y S para aquellos eventos posicionados en la vecindad de este plano resulta insuficiente (señales colineales) para garantizar una inversión precisa y unívoca del mecanismo (Grechka, 2015a). Ahora, si bien la teoría solo asegura que en este plano no pueden recuperarse las 6 componentes en forma independiente, nada dice sobre cuál de ellas es la que depende de las demás. Cabe recordar que la construcción aleatoria de nuestros tensores no supone una relación previa sobre ninguno de sus elementos. Así, es justo atribuir este agrupamiento planar de mecanismos de "baja calidad" a la ambigüedad que se desprende de la teoría (manifestada en señales colineales), pero nada podemos decir sobre cuál elemento particular del tensor es el que tiene mayor error. Dicho de otra forma, el análisis individual de cada uno de estos mecanismos no tiene porqué indicar una tendencia sobre alguna de sus componentes en particular. Cabe aclarar que este análisis, aunque interesante, está fuera del alcance de esta Tesis.

Volviendo a la Figura 4.6, el primer panel (panel superior izquierdo) solo muestra los mecanismos con $\bar{E}_r \in [0.0, 20.0)$ %. Como puede verse, la escala de colores es uniforme y los eventos muestran una distribución espacial pareja en todo el plano, a excepción del plano que une ambos pozos. El hecho de que los sub-grupos con mejores y peores errores relativos (arriba izquierda y abajo derecha) tengan una distribución espacial que es mutuamente excluyente muestra que existe una clara dependencia espacial en la capacidad de predicción de la red. Conforme se avanza con el análisis por los paneles 2, 3 y 4 (arriba derecha, abajo izquierda y abajo derecha, respectivamente) se observa que van desapareciendo las predicciones de mayor calidad y comienza a distinguirse una región del espacio que aglutina las predicciones con mayor error cercanas al plano de los pozos. En función de este resultado

se puede decir que, bajo esta geometría y considerando un medio isotrópico, la NN no es capaz de predecir los mecanismos verdaderos de aquellos eventos localizados en la vecindad de este plano. Esto también nos permite pensar que una red entrenada para realizar MTI en una geometría *single-well* vertical también presentaría esta dificultad, pero acentuado en todas las direcciones de acimut. Esto es, la escala de colores sería pareja para toda la nube de eventos pero en escalas similares a las mostradas por los paneles 3 y 4.

Además del plano mencionado, en el panel 2 de la Figura 4.6 puede observarse una zona circular centrada en cada uno de los pozos que contiene una alta densidad de eventos con errores $20 \leq \bar{E}_r < 40\,\%$. Esta zona es formada por la proyección de aquellos eventos situados en esferas con sus centros en cada uno de los las receptores de ambos pozos. Este fenómeno ocurre porque las ventanas temporales utilizadas para extraer las amplitudes del dato se superponen para los eventos situados en dicha región. Como resultado, las amplitudes extraídas por las ventanas pueden ser erróneas en algunos casos por la imposibilidad de discernir entre las fases P y S. Como consecuencia, en algunos casos el algoritmo asigna la misma amplitud a ambas fases, lo que naturalmente no es correcto. La frontera de este círculo (o una esfera en 3D[*]) indicaría la región a partir de la cual las ventanas permiten extraer las amplitudes correctamente. Esta distancia crítica, d_c, define el radio de este círculo y queda determinada por el ancho de la ventana y el modelo de velocidades. En nuestro caso, el método considera una ventana de 0.015 s, por lo que la superposición de ventanas se daría cuando la diferencia entre los tiempos de arribo de las fases P y S fuese igual o menor a este número. Planteando esta condición, obtenemos que

$$t_S - t_P = \frac{r}{V_S} - \frac{r}{V_P} = r\left(\frac{1}{V_S} - \frac{1}{V_P}\right) = 0.015\,\mathrm{s}, \tag{4.8}$$

lo que permite despejar la distancia crítica, $r = d_c$, como

$$d_c = \frac{V_P\,V_S}{V_P - V_S} \times 0.015\,\mathrm{s} = 131.25\,\mathrm{m}. \tag{4.9}$$

Se puede ver que la NN no solo tendrá una región de baja confidencia en el plano definido por ambos pozos, sino que también tendrá dificultades para invertir eventos cercanos a dichos pozos, en especial cuando la distancia a los mismos tenga una proyección igual o menor a d_c.

En la Figura 4.7 (columna izquierda) se muestra una señal de muy mala calidad. La misma pertenece al grupo de eventos cuyo mecanismo es recuperado con un error relativo superior a 175 %. La mala calidad de esta señal se debe, probablemente, a una combinación de la posición misma del evento y su mecanismo focal. En cuanto a las amplitudes recuperadas para este evento, se observa que debido a su cercanía con varios de los receptores del arreglo, las ventanas de tiempo a partir de las cuales se extraen las amplitudes

[*]Si consideramos un medio no-isotrópico (ya sea anisotropía o un medio *layered*, o un pozo desviado, por ejemplo) estas regiones ya no serían esféricas y su proyección en planta no produciría un círculo perfecto

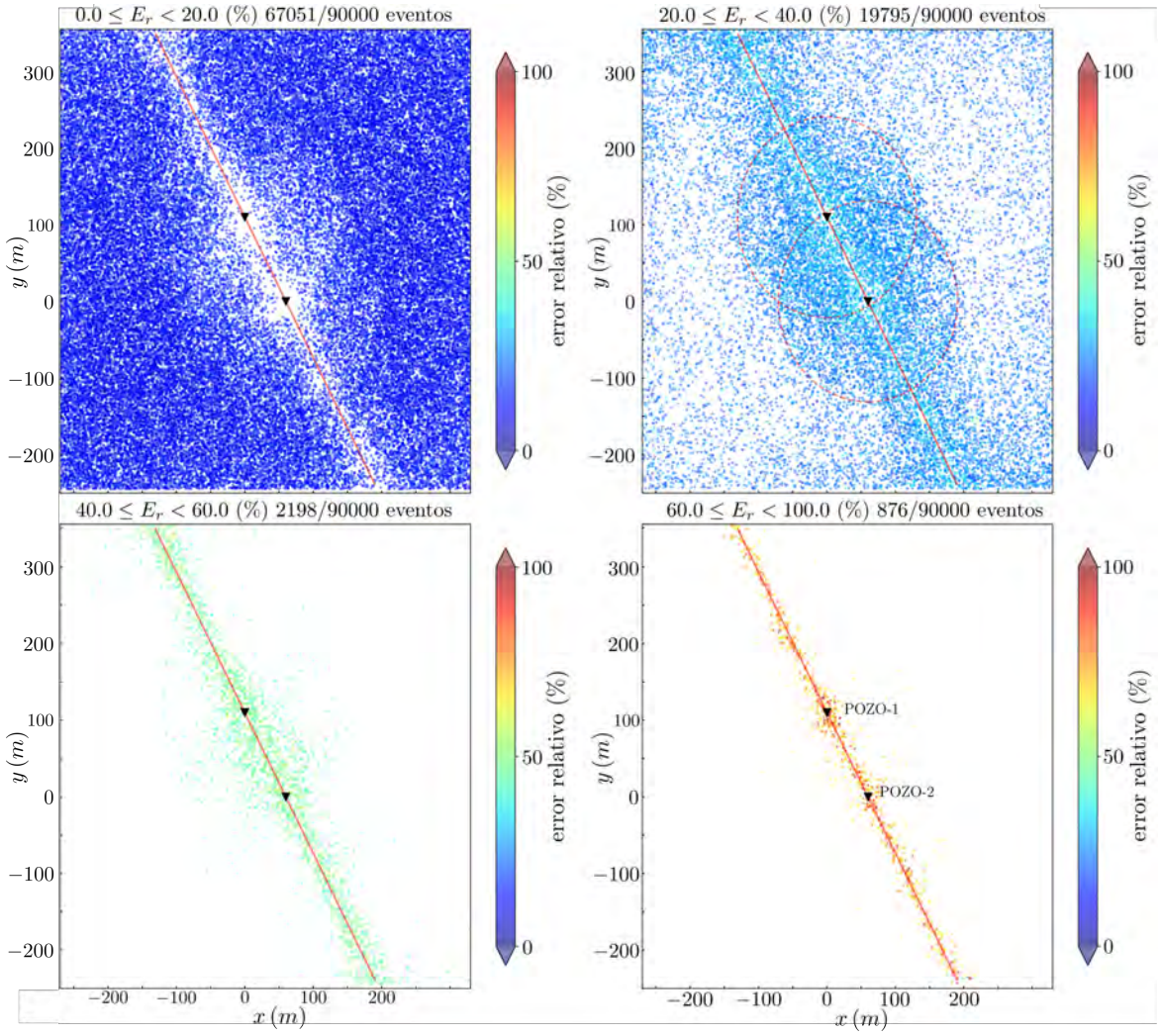


Figura 4.6: Modelo 1: Distribución espacial de los errores relativos para los tensores momento predichos para diferentes rangos de $\bar{E}_r$. Los mayores errores se observan en posiciones cercanas al plano vertical que contiene a los dos pozos monitores.

se encuentran total o parcialmente superpuestas. Esto implica que el vector de amplitudes utilizado como dato de entrada sea erróneo, y por lo tanto, el mecanismo invertido será de mala calidad. Como se menciona anteriormente, el problema de la superposición de las ventanas es inherente al método utilizado para la extracción de las amplitudes y persiste hasta una distancia crítica que depende del modelo de velocidades y el ancho de la ventana utilizada, como expresa la ecuación 4.9.

Modelo 2: dos pozos inmersos en un medio VTI semi-infinito

En este caso, los resultados de la predicción se resumen en la Figura 4.8 y la Tabla 4.3. Como vemos, la red neuronal entrenada también es capaz de predecir casi la totalidad de los eventos del set de datos con gran certeza y errores relativos pequeños. Los promedios globales de correlación y de error relativo arrojan valores de $\bar{\rho} \sim 0.988$ y $\bar{E}_r \sim 15.6\,\%$. Bajo este modelo, se ve que $71\,\%$ de todos los mecanismos de fractura son predichos con un coeficiente de correlación $\bar{\rho} \geq 0.99$ y un error relativo $\bar{E}_r < 12.4\,\%$, mientras que aproximadamente $19\,\%$ y $4.6\,\%$ del dato restante entregó valores de correlaciones en los intervalos $0.98 \leq \bar{\rho} < 0.99$ ($\bar{E}_r \sim 18.2\,\%$) y $0.97 \leq \bar{\rho} < 0.98$ ($\bar{E}_r \sim 24.1\,\%$), respectivamente. Puede decirse que casi la totalidad de los eventos ($94.6\,\%$) fue predicho con una certeza mayor a 0.97.

En la Figura 4.9 se muestran las *beach-ball* correspondientes a los resultados arrojados por la red neuronal bajo el modelo 2. Nuevamente, se observa que las *beach-balls* predichas (rojas) son prácticamente indiscernibles de aquellas correspondientes al dato verdadero (verdes), especialmente para el caso $\bar{\rho} \geq 0.99$, que agrupa más del $71\,\%$ de todos los eventos de testeo. Al igual que para el modelo 1, los valores numéricos son también muy similares.

Del análisis de la Figura 4.10 se desprenden conclusiones similares a las encontradas para el modelo 1. Comparando los paneles inferiores derechos de esta figura y su equivalente para el modelo 1, vemos que la cantidad de eventos con errores $60 \leq \bar{E}_r \leq 100\,\%$ es mayor para el modelo 1. Esto indicaría que, si bien la anisotropía VTI no otorga información crucial para resolver la ambigüedad para aquellos eventos posicionados en el plano que contiene a los dos pozos (Vavryčuk, 2007; Grechka, 2015a), la inversión mejora levemente para aquellos eventos en la vecindad de dicho plano. Por otro lado, los valores $\bar{\rho}$ y $\bar{E}_r$ son superiores a los calculados para el caso isótropo del modelo 1. Esto se debe principalmente a que el volumen de eventos correspondientes al sub-grupo con $0.0 \leq \bar{E}_r \leq 20.0\,\%$ es mayor al correspondiente para la red del modelo 1. Esto puede verse en los valores de la Tablas 4.2 y 4.3, así como también en las Figuras 4.6 y 4.10, donde los paneles superiores izquierdos indican 67051 y 74592 eventos para el modelo 1 y 2, respectivamente.

Por otro lado, en el segundo panel de la Figura 4.10 ya no puede observarse una proyección "circular" tan evidente como la mostrada para el modelo homogéneo. Esto puede

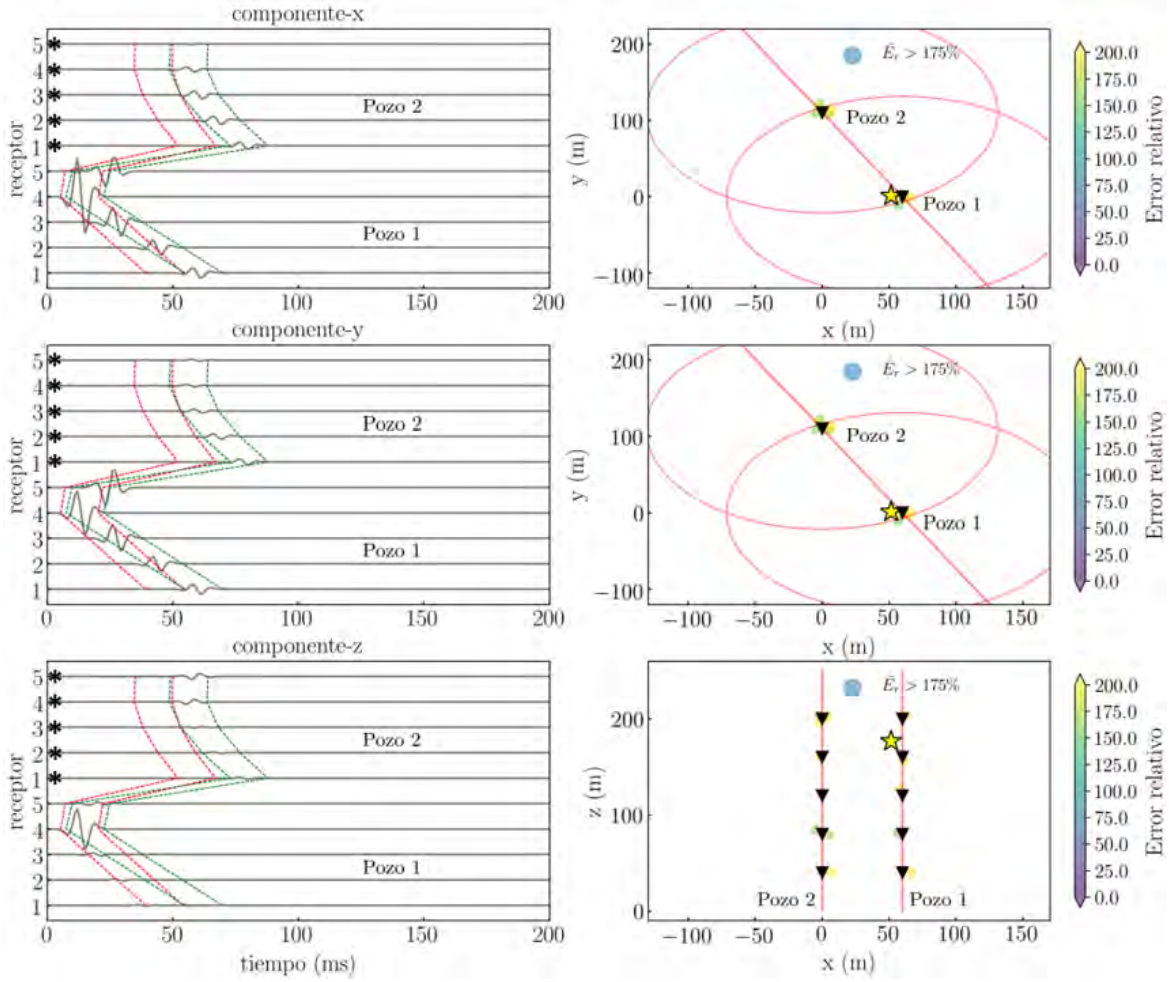


Figura 4.7: Modelo 1: Columna izquierda: Señal correspondiente a un evento con $\bar{E}_r > 175\,\%$. Se indican las ventanas utilizadas para extraer las amplitudes de las fases P (roja) y S (verde). Columna derecha: Disposición espacial de todos los eventos con $\bar{E}_r > 175\,\%$. El evento particular cuya señal se muestra en la columna de la izquierda está indicado con una estrella. Las trazas marcadas con un asterisco corresponden a la señal recibida por el Pozo 2.

Certeza			Error relativo	
$\bar{\rho}$	Eventos %	$\bar{E}_r$ %	$\bar{E}_r$ %	Eventos %
$[0.99, 1.00]$	71.0378	12.39	$[100, \infty)$	0.058
$[0.98, 0.99)$	18.9756	18.24	$[90, 100)$	0.076
$[0.97, 0.98)$	4.6489	24.09	$[80, 90)$	0.083
$[0.96, 0.97)$	1.6944	28.75	$[70, 80)$	0.180
$[0.95, 0.96)$	0.8667	32.89	$[60, 70)$	0.315
$[0.94, 0.95)$	0.5433	36.62	$[50, 60)$	0.472
$[0.84, 0.94)$	1.6789	47.75	$[40, 50)$	0.887
$[0.74, 0.84)$	0.3678	70.45	$[30, 40)$	2.301
$[0.64, 0.74)$	0.1222	289.87	$[20, 30)$	12.74
$[0.54, 0.64)$	0.0489	96.19	$[10, 20)$	63.76
$[0.44, 0.54)$	0.0100	104.09	$[0, 10)$	19.11
$[0.34, 0.44)$	0.0033	113.55		
$[0.24, 0.34)$	0.0011	128.24		
$[0.14, 0.24)$	0.0000	-		
$[0.04, 0.14)$	0.0011	219.98		
$[-1.0, 0.04)$	0.0000	-		

Tabla 4.3: Modelo 2: Análisis de los histogramas de correlación y error relativo. El dato es separado en sub-grupos con intervalos irregulares que contienen información relevante a partir de ambos histogramas.

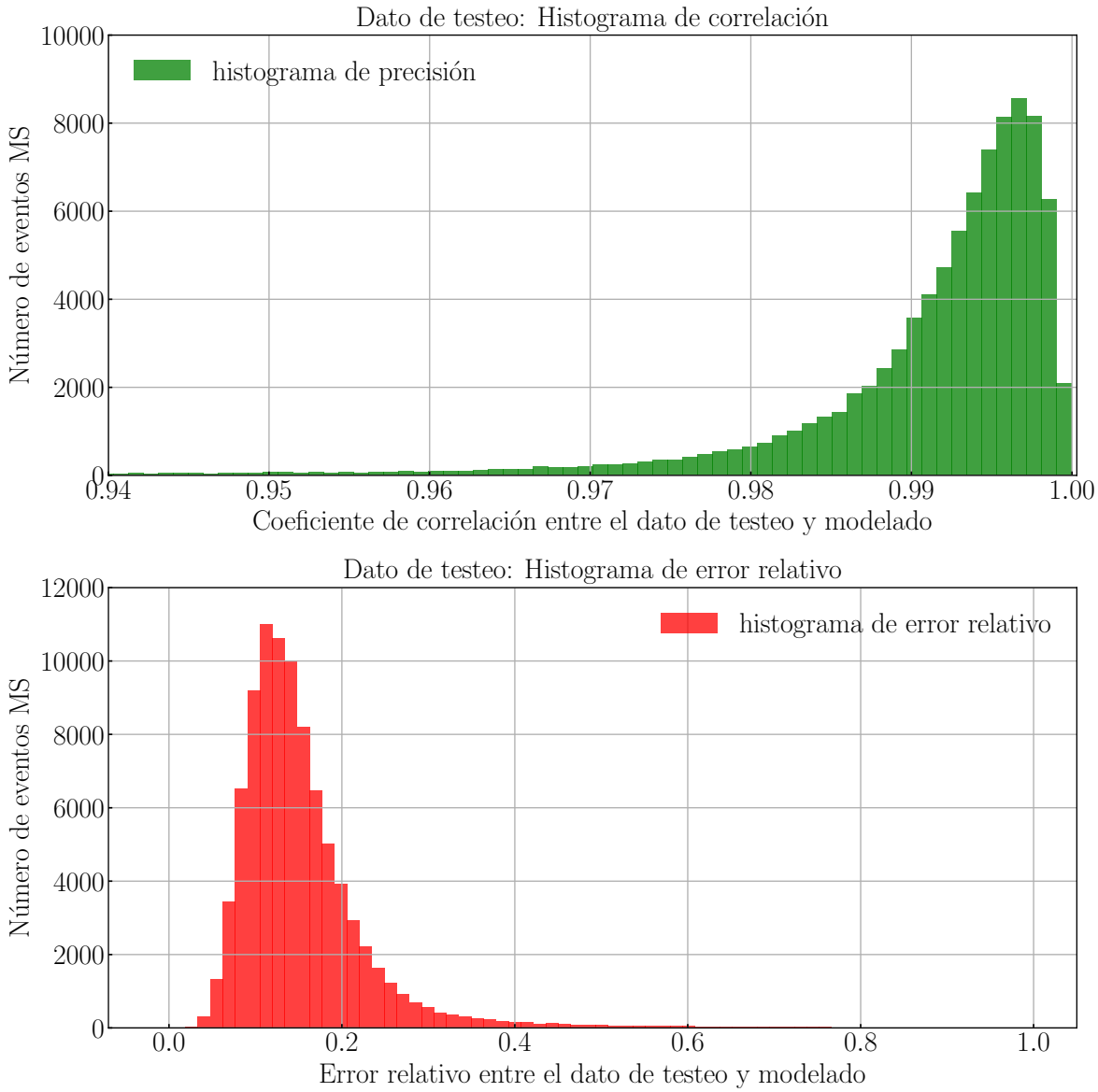


Figura 4.8: Modelo 2: Histogramas de los resultados predichos. Arriba: histograma de coeficientes de correlación. Abajo: histograma de errores relativos. En ambos casos el histograma es calculado utilizando los resultados predichos por el modelo al ser evaluado sobre el dato de testeo. Los máximos de los histogramas indican valores ~ 0.997 y $\sim 10\,\%$.

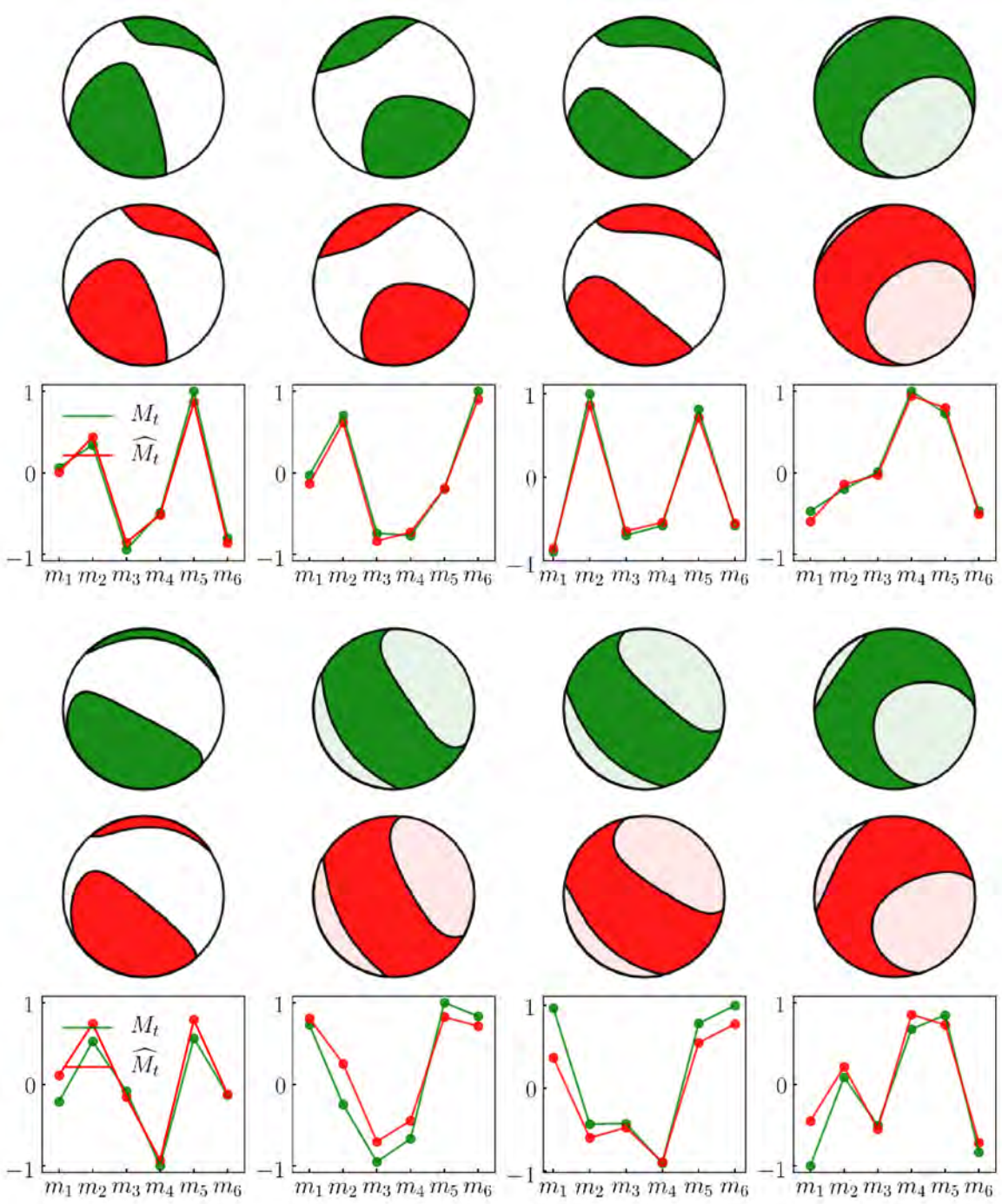


Figura 4.9: Modelo 2: Diagramas *beach-ball* calculados a partir de los TMs (predichos y su correspondiente tensor deseado). Filas 1 y 2: *beach-balls* calculadas a partir de un mecanismo de testeo (verde) y su correspondiente mecanismo predicho (rojo) para 4 (uno por columna) datos pertenecientes al grupo con $0.99 \leq \bar{\rho} \leq 1.0$. Fila 3: valores numéricos para los 6 elementos independientes de cada evento. Filas 4 a 6: idem para 4 eventos del grupo con $0.94 \leq \bar{\rho} < 0.95$.

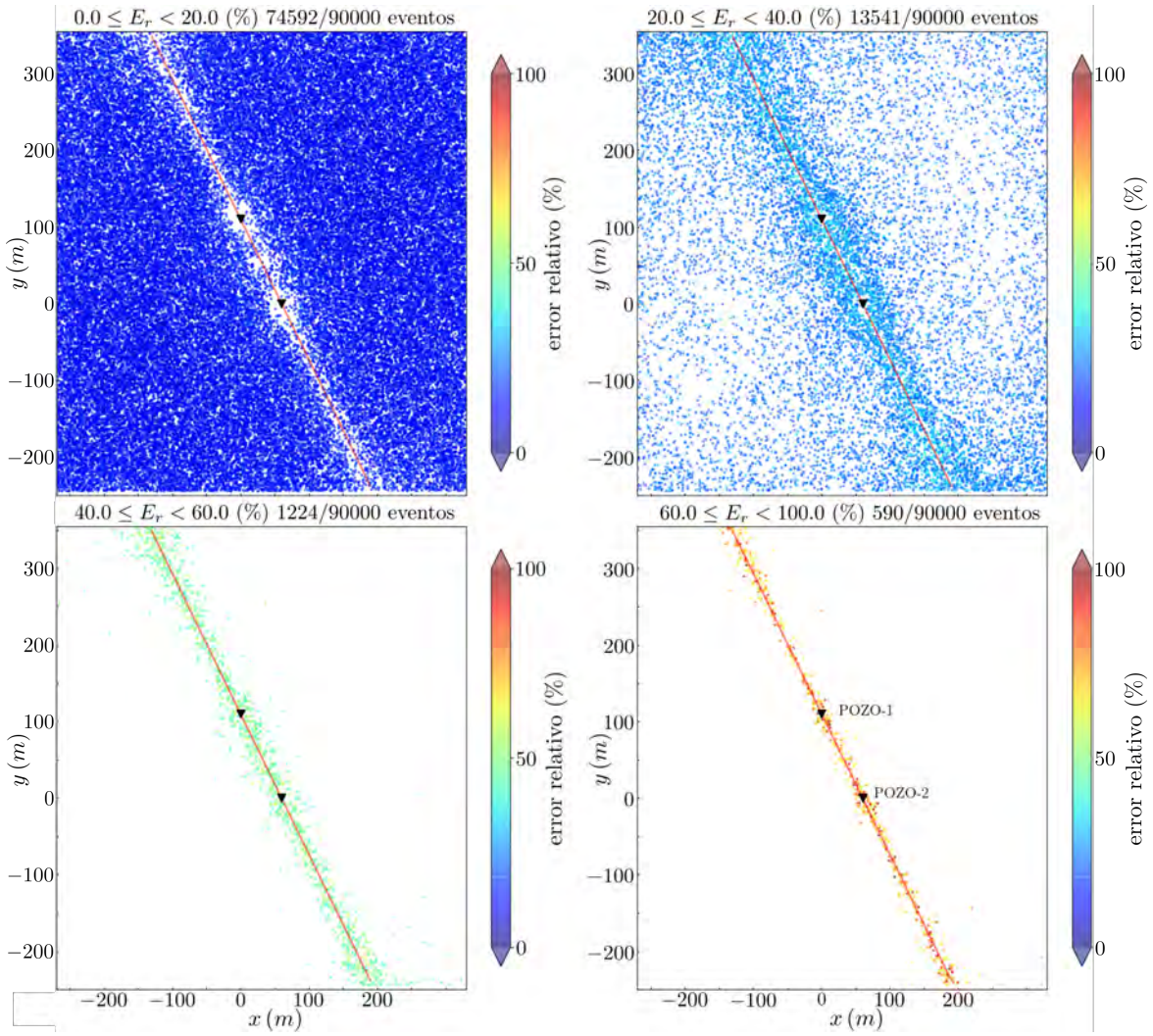


Figura 4.10: Modelo 2: Distribución espacial de los errores relativos para los tensores momento predichos para diferentes rangos de $\bar{E}_r$. Los mayores errores son observados en posiciones cercanas al plano que contiene a los dos pozos monitores verticales.

deberse a que, para el caso anisotrópico, la distancia crítica, d_c, que determina la distancia a partir de la cual las ventanas de tiempo comienzan a superponerse dependería, a través del modelo de velocidades, del ángulo de inclinación θ que forma el rayo de la señal incidente con cada uno de los receptores (Tsvankin, 2012; Grechka, 2020). Esto es

$$d_c = \frac{V_P(\theta)\, V_S(\theta)}{V_P(\theta) - V_S(\theta)} \times 0.015\,\mathrm{s} \tag{4.10}$$

para ambas fases de cizalla. Esta ecuación muestra que la distancia crítica no es constante, y por ello, la proyección del cilindro de radio d_c (en el espacio 3D) que formaba el círculo en una imagen en planta para el caso isotrópico, ya no es tan clara.

Modelo 3: dos pozos inmersos en un medio VTI de dos capas planas

Para este caso, los resultados obtenidos luego de entrenar la red se muestran en la Figura 4.11 y la Tabla 4.4. La red entrenada arroja valores de $\bar{\rho} \sim 0.9783$ y $\bar{E}_r \sim 20.60\,\%$. Estos valores son inferiores a los mostrados por los modelos 1 y 2. Vemos que aproximadamente 47 % de todos los mecanismos de fractura son predichos con un coeficiente de correlación $\bar{\rho} \geq 0.99$ y un error relativo $\bar{E}_r < 14\,\%$, mientras que aproximadamente 26.5 % y 10.2 % del dato restante entregó valores de correlaciones en los intervalos $0.98 \leq \bar{\rho} < 0.99$ ($\bar{E}_r \sim 19.0\,\%$) y $0.97 \leq \bar{\rho} < 0.98$ ($\bar{E}_r \sim 24.5\,\%$), respectivamente. En este caso, cerca del 84.46 % de los eventos fue predicho con una certeza mayor a 0.97.

Como en los dos casos anteriores, las *beach-balls* de la Figura 4.12 muestran que los mecanismos predichas (rojo) son prácticamente indiscernibles de aquellos obtenidas a partir del dato sintético verdadero (verdes), especialmente para el caso $\bar{\rho} \geq 0.99$, que agrupa cerca del 50 % de todos los eventos de testeo. Al igual que para los anteriores modelos, los valores numéricos de los 6 elementos son también muy similares.

De la Figura 4.13 se desprenden resultados comparables a los mostrados para el modelo 2. Por ello, las conclusiones que se desprenden de esta figura son similares a las que se encontraron para este último. Como es de esperarse, la incorporación de un modelo de capas planas y horizontales no es suficiente para resolver el problema de la ambigüedad para aquellos eventos posicionados sobre o en la vecindad del plano que contiene a los dos pozos.

Por otro lado, los valores de $\bar{\rho}$ y $\bar{E}_r$ son inferiores a los obtenidos para los modelos 1 y 2. Esto se desprende del número de eventos contenidos en el sub-grupo con $0.0 \leq \bar{E}_r \leq 20.0\,\%$, que es menor al obtenido para los modelos anteriores. Al igual que para el modelo 2, el fenómeno que se manifiesta como una proyección "circular" para la red entrenada en el modelo 1 no es tan evidente en este último modelo.

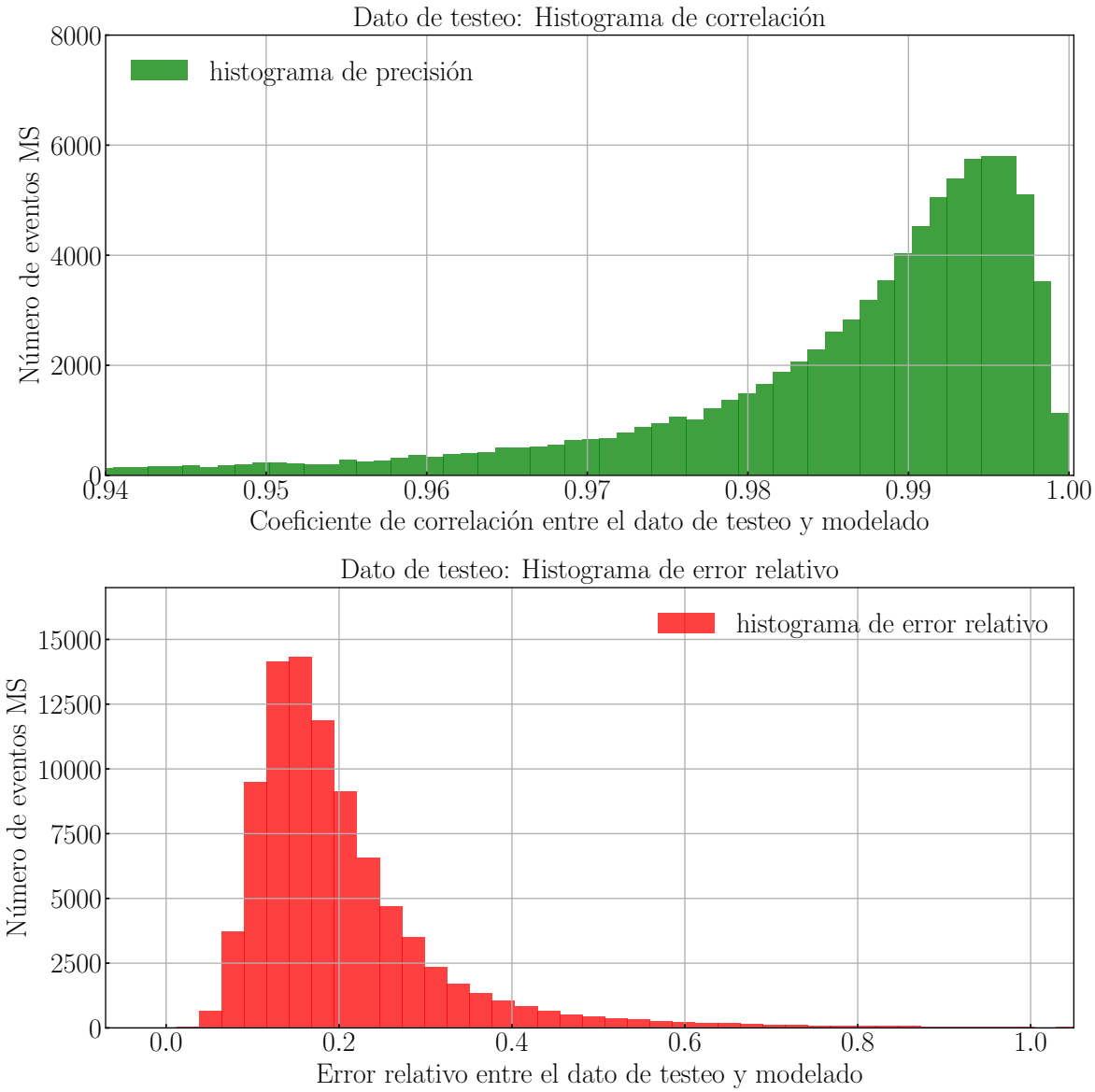


Figura 4.11: Modelo 3: Histogramas de los resultados predichos. Arriba: histograma de coeficientes de correlación. Abajo: histograma de errores relativos. En ambos casos el histograma es calculado utilizando los resultados predichos por el modelo al ser evaluado sobre el dato de testeo.

Certeza			Error relativo	
$\bar{\rho}$	Eventos %	$\bar{E}_r$ %	$\bar{E}_r$ %	Eventos %
$[0.99, 1.00]$	47.6911	13.7417	$[100, \infty)$	0.280
$[0.98, 0.99)$	26.5822	19.0475	$[90, 100)$	0.194
$[0.97, 0.98)$	10.1867	24.5459	$[80, 90)$	0.311
$[0.96, 0.97)$	4.9022	29.2354	$[70, 80)$	0.447
$[0.95, 0.96)$	2.6278	33.2021	$[60, 70)$	0.744
$[0.94, 0.95)$	1.7133	36.5984	$[50, 60)$	1.324
$[0.84, 0.94)$	4.6622	47.5041	$[40, 50)$	2.742
$[0.74, 0.84)$	0.9611	70.7462	$[30, 40)$	6.925
$[0.64, 0.74)$	0.3422	85.9820	$[20, 30)$	24.367
$[0.54, 0.64)$	0.1611	100.5294	$[10, 20)$	54.708
$[0.44, 0.54)$	0.1000	106.3545	$[0, 10)$	7.953
$[0.34, 0.44)$	0.0311	129.5885		
$[0.24, 0.34)$	0.0189	134.6054		
$[0.14, 0.24)$	0.0089	158.03		
$[0.04, 0.14)$	0.0033	139.62		
$[-1.0, 0.04)$	0.0078	210.45		

Tabla 4.4: Modelo 3: Análisis de los histogramas de correlación y error relativo. El dato es separado en sub-grupos con intervalos irregulares que contienen información relevante a partir de ambos histogramas.

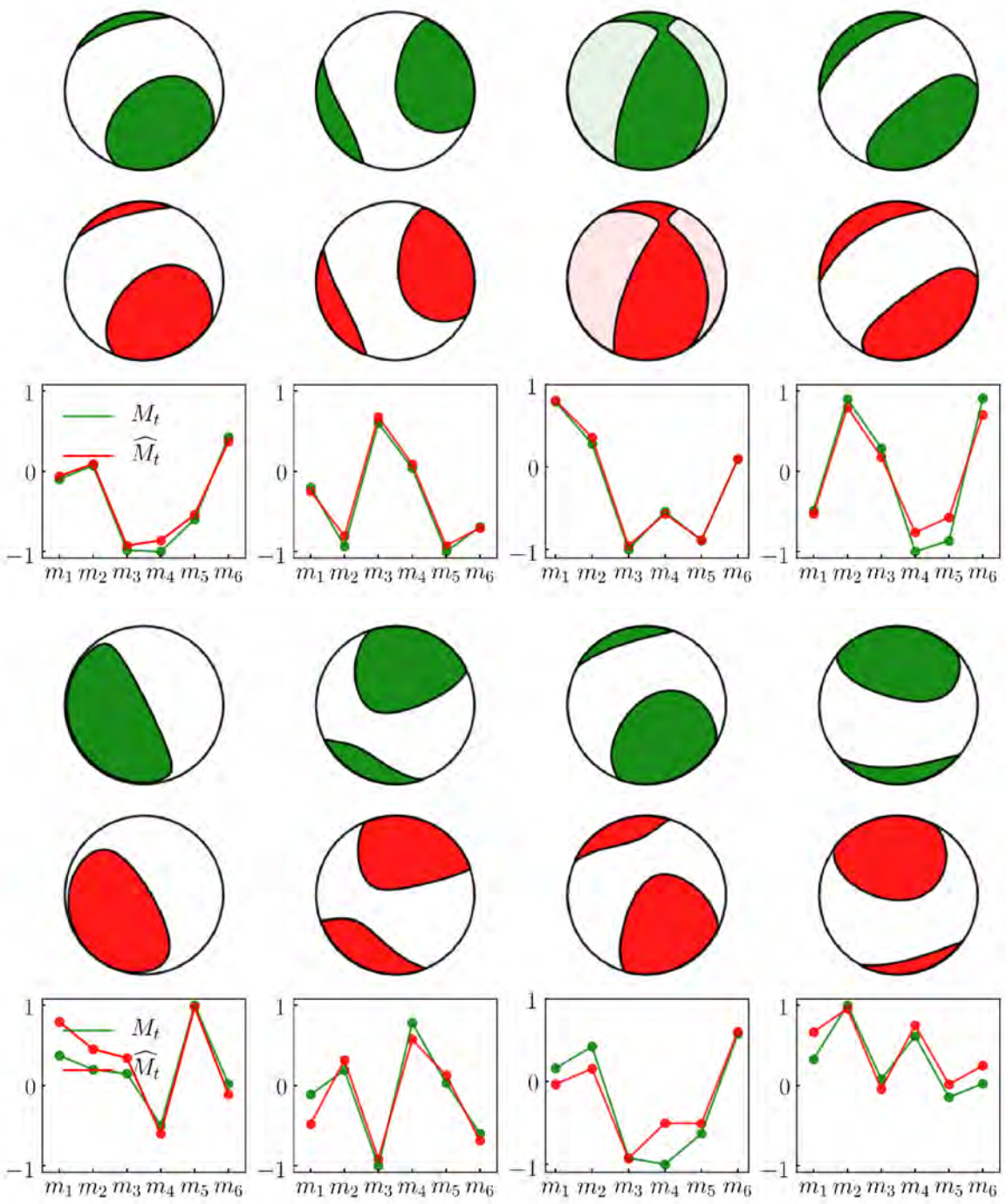


Figura 4.12: Modelo 3: Diagramas *beach-ball* calculados a partir de los TMs (predichos y su correspondiente tensor deseado). Filas 1 y 2: *beach-balls* calculadas a partir de un mecanismo de testeo (verde) y su correspondiente mecanismo predicho (rojo) para 4 (uno por columna) datos pertenecientes al grupo con $0.99 \leq \bar{\rho} \leq 1.0$. Fila 3: valores numéricos para los 6 elementos independientes de cada evento. Filas 4 a 6: idem para 4 eventos del grupo con $0.94 \leq \bar{\rho} < 0.95$.

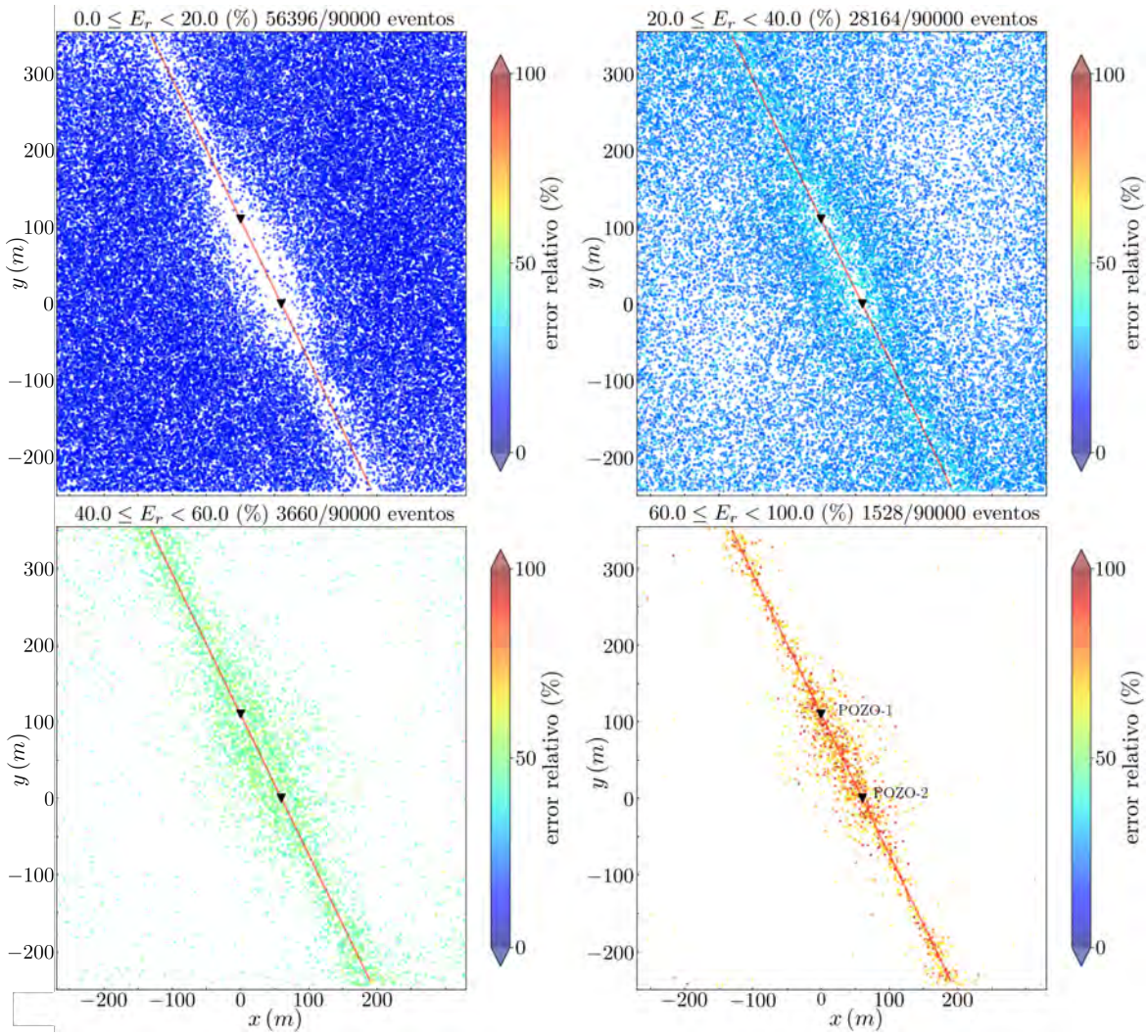


Figura 4.13: Modelo 3: Distribución espacial de los errores relativos para los tensores momento predichos para diferentes rangos de $\bar{E}_r$. Los peores errores son observados en posiciones cercanas al plano que contiene a los dos pozos monitores verticales.

MTI utilizando *single-well* con desviación

En esta sección se estudia el desempeño de la red neuronal bajo circunstancias de monitoreo menos favorables. Para comenzar, consideraremos el modelo de velocidades isótropo descripto por el modelo 1 y una geometría de adquisición de tipo *single-well*. Para este experimento nos valdremos de las conclusiones halladas en Vera Rodríguez et al. (2011) y utilizaremos un arreglo de geófonos desplegados en un pozo curvo, de curvatura mayor a $6°/30$ m. Para mantenernos en el plano realista, y a su vez aumentar las probabilidades de éxito, utilizaremos un arreglo de 12 geófonos de 3 componentes (esta cantidad de geófonos es estándar para un trabajo de monitoreo microsísmico en la región de Vaca Muerta). La separación de receptores será de $\Delta r = 30$ m. Para estar seguros de contar con la curvatura suficiente, se utiliza un ángulo de inclinación inicial igual a $\alpha_0 = 8°$, que es mayor al sugerido en Vera Rodríguez et al. (2011). Así, las coordenadas de los receptores estarán definidas por las expresiones:

$$\begin{aligned} x_i &= x_{i-1} + \sin\left[\alpha_0(i-1)\right]\Delta r, \\ y_i &= 0, \\ z_i &= z_{i-1} + \cos\left[\alpha_0(i-1)\right]\Delta r \end{aligned} \tag{4.11}$$

con $i = 2, \cdots, 12$ y $(x_1, y_1, z_1) = (10, 0, 40)$ m. La Figura 4.14 muestra la geometría de monitoreo diseñada para este experimento.

De la trayectoria diseñada para el pozo, se observa que la misma sigue contenida en un plano (en este caso el plano $y = 0$). A los efectos de la teoría, este plano es un plano de simetría (aun en un medio VTI) y por lo tanto, es razonable asumir que la distribución espacial de los errores en la predicción del mecanismo focal indique la existencia de este plano.

En cuanto al dato, se utilizan la misma cantidad de eventos para los conjuntos de entrenamiento y de testeo. El entrenamiento es llevado a cabo con los mismos exactos parámetros utilizados para los anteriores casos. Los resultados del mismo arrojan los valores de $\bar{\rho} \sim 0.9658$ y $\bar{E}_r \sim 27.66\,\%$. Como es esperable, estos resultados son inferiores a los obtenidos para la red entrenada en el modelo 1 utilizando una geometría *dual-well*. La Figura 4.15 y la Tabla 4.5 resumen los resultados encontrados para el desempeño de esta red. Vemos que el aproximadamente 24.6 % de todos los mecanismos de fractura son predichos con un coeficiente de correlación $\bar{\rho} \geq 0.99$ y un error relativo $\bar{E}_r < 17\,\%$, mientras que aproximadamente 28 % y 16 % del dato restante entregó valores de correlaciones en los intervalos $0.98 \leq \bar{\rho} < 0.99$ ($\bar{E}_r \sim 20.8\,\%$) y $0.97 \leq \bar{\rho} < 0.98$ ($\bar{E}_r \sim 25.4\,\%$), respectivamente. En este caso, cerca del 68 % de los eventos fue predicho con una certeza mayor a 0.97. De la tabla se desprende que el mayor porcentaje de eventos es recuperado en el sub-grupo de $20 \leq \bar{E}_r \leq 30$, mientras que en el desempeño de las redes entrenadas en una geometría de doble pozo monitor se agrupa la mayor cantidad de eventos en el sub-grupo

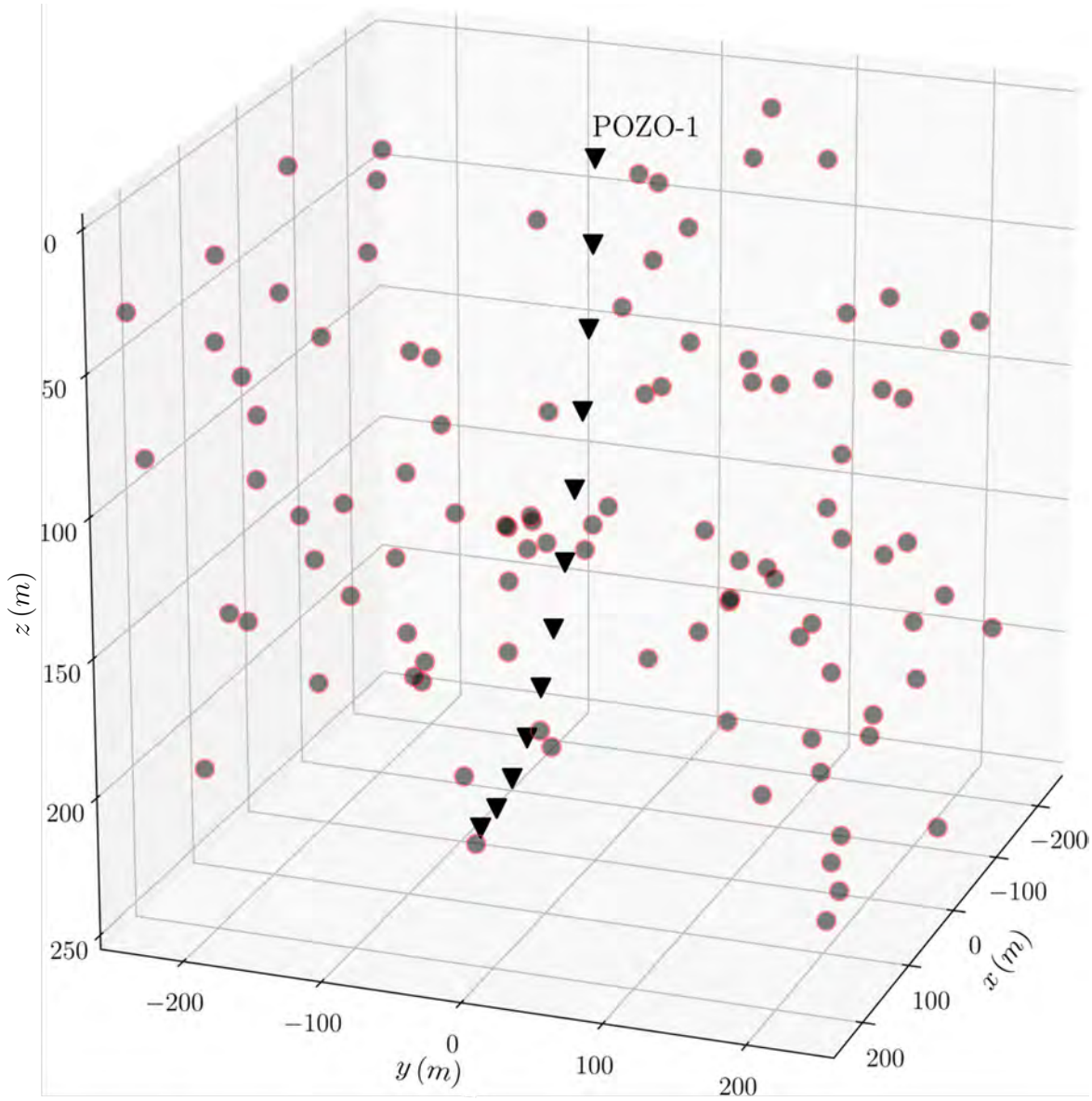


Figura 4.14: Modelo *single-well*: Geometría de monitoreo. Se exhibe el pozo inclinado con curvatura donde se desplegaron 12 geófonos (triángulos negros). Se muestran, a modo únicamente ilustrativo, un total de 100 eventos distribuidos de forma aleatoria (círculos grises).

Certeza			Error relativo	
$\bar{\rho}$	Eventos %	$\bar{E}_r$ %	$\bar{E}_r$ %	Eventos %
$[0.99, 1.00]$	24.6878	17.6816	$[100, \infty)$	0.420
$[0.98, 0.99)$	27.4633	20.7903	$[90, 100)$	0.331
$[0.97, 0.98)$	15.9733	25.4250	$[80, 90)$	0.664
$[0.96, 0.97)$	8.9789	29.8440	$[70, 80)$	1.132
$[0.95, 0.96)$	5.6967	33.7671	$[60, 70)$	1.870
$[0.94, 0.95)$	3.7067	37.6368	$[50, 60)$	3.263
$[0.84, 0.94)$	10.5522	49.1514	$[40, 50)$	6.333
$[0.74, 0.84)$	2.0444	73.3898	$[30, 40)$	15.58
$[0.64, 0.74)$	0.6356	87.8834	$[20, 30)$	36.07
$[0.54, 0.64)$	0.1811	103.9991	$[10, 20)$	33.10
$[0.44, 0.54)$	0.0533	113.1015	$[0, 10)$	1.221
$[0.34, 0.44)$	0.0189	146.5574		
$[0.24, 0.34)$	0.0078	155.7622		
$[0.14, 0.24)$	0.0000	-		
$[0.04, 0.14)$	0.0000	-		
$[-1.0, 0.04)$	0.0000	-		

Tabla 4.5: Modelo *single-well*: Análisis de los histogramas de correlación y error relativo. El dato es separado en sub-grupos con intervalos irregulares que contienen información relevante a partir de ambos histogramas.

de $10 \leq \bar{E}_r \leq 20$. Esto se traduce en histogramas más "aplastados", como puede observarse en la Figura 4.15. Es evidente que, aun cuando la cantidad de geófonos (12) es superior al total de herramientas desplegadas para el modelo 1 (10), la distribución espacial de los mismos juega un papel crucial en la resolución de un problema de MTI. Debido a que la complejidad de resolver un problema de MTI bajo estas condiciones aumenta, es justo aseverar que el desempeño global de la red es inferior a la mostrada para los casos de tipo *multi-well*.

En cuanto a la representación de los mecanismos mediante los diagramas *beach-balls*, se observa que los mecanismos predichos son similares a los verdaderos.

La Figura 4.17 nos muestra la distribución espacial de los errores en la predicción. Nuevamente podemos observar un plano donde la red tiene dificultades para predecir eventos con alta calidad. También podemos ver cómo, en las cercanías del pozo tenemos un ensanchamiento de esta región, posiblemente asociado a la superposición de las ventanas que extraen las amplitudes.

Los análisis del desempeño de una red neuronal entrenada bajo geometrías de tipo

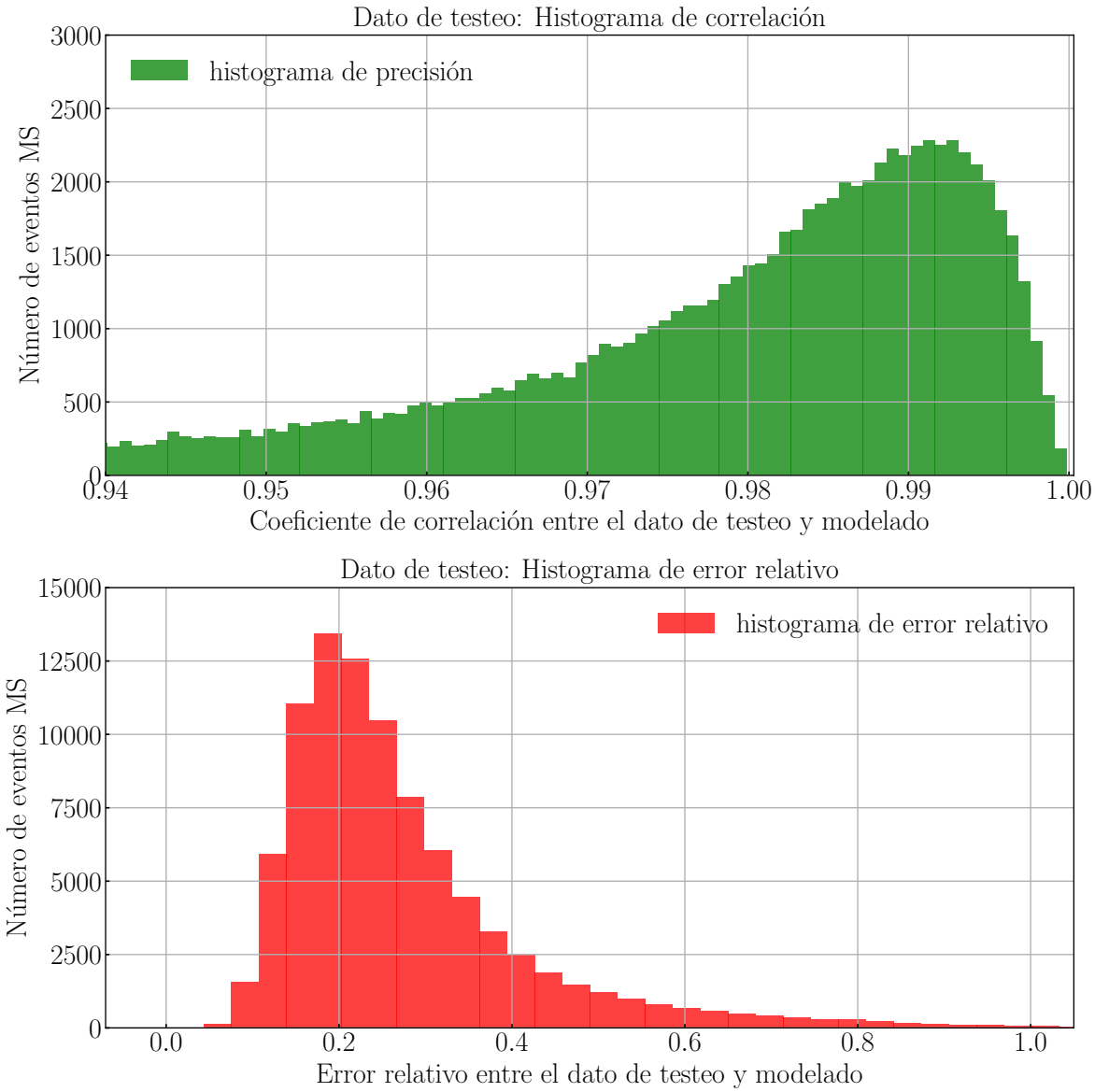


Figura 4.15: Modelo *single-well*: Histogramas de los resultados predichos. Arriba: histograma de coeficientes de correlación. Abajo: histograma de errores relativos. En ambos casos, el histograma es calculado utilizando los resultados predichos por el modelo al ser evaluado sobre el dato de testeo.

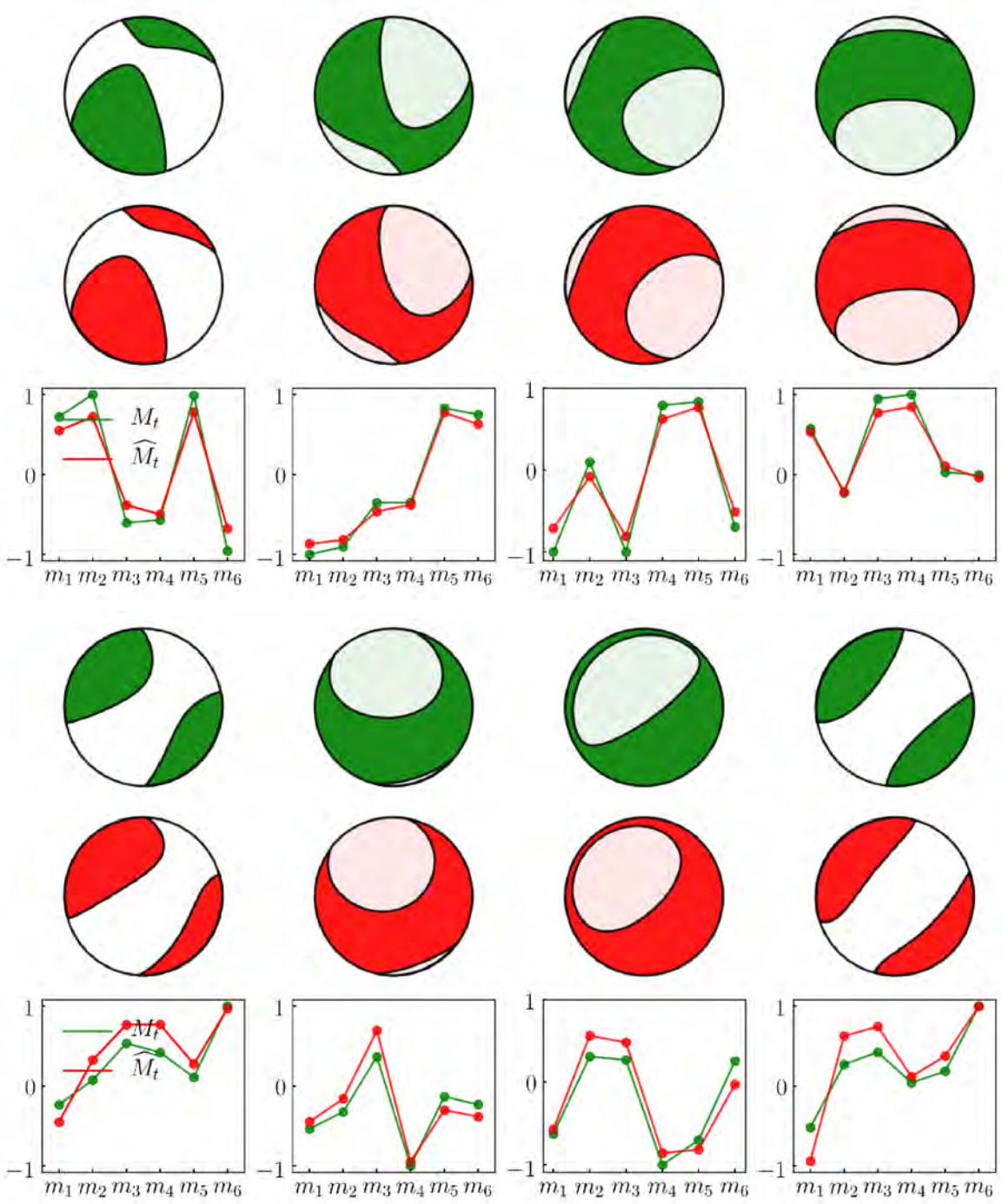


Figura 4.16: Modelo *single-well*: Diagramas *beach-ball* calculados a partir de tensores momento (predichos y su correspondiente tensor deseado). Filas 1 y 2: *beach-balls* calculadas a partir de un mecanismo de testeo (verde) y su correspondiente mecanismo predicho (rojo) para 4 (uno por columna) datos pertenecientes al grupo con $0.99 \leq \bar{\rho} \leq 1.0$. Fila 3: valores numéricos para los 6 elementos independientes de cada evento. Filas 4 a 6: idem para 4 eventos del grupo con $0.94 \leq \bar{\rho} < 0.95$.

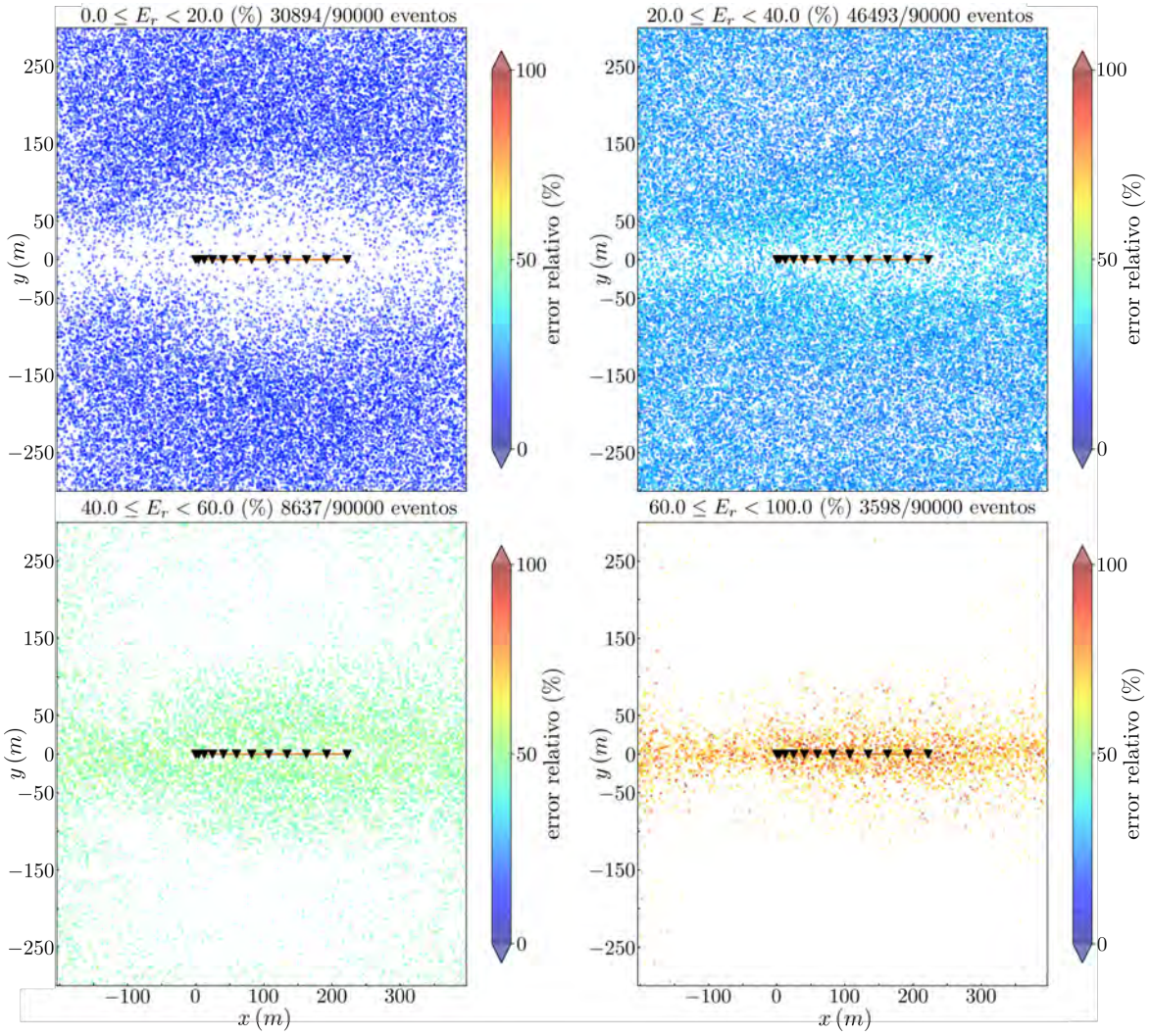


Figura 4.17: Modelo *single-well*: Distribución espacial de los errores relativos para los tensores momento predichos para diferentes rangos de $\bar{E}_r$. Los peores errores son observados en posiciones cercanas a la vecindad del plano que contiene al pozo monitor inclinado.

single-well con a) pozo perfectamente vertical y b) pozo no contenido en un plano de simetría, aunque interesantes, quedan por fuera del alcance de este trabajo. En principio, uno podría predecir que el caso a) será el más desfavorable de todos, en tanto que el b) permitiría obtener los mejores resultados, ya que presentaría menos simetrías. Esto resultaría en un menor número de simetrías y por ende, mayor cantidad de datos independientes, lo que a su vez favorece la resolución de cualquier problema inverso, y en especial de MTI.

4.4. Discusión

De los resultados obtenidos, y considerando los valores porcentuales de la cantidad de eventos cuyos mecanismos son recuperados con correlaciones mayores al 0.9, se puede decir que la red neuronal fue capaz de realizar predicciones del tensor momento exitosamente en los 3 escenarios *dual-well* propuestos. Si bien el desempeño de la red fue aceptable en el escenario *single-well*, el mismo no alcanzó los niveles de precisión alcanzados en los otros 3 casos. En líneas generales, la gran mayoría de los mecanismos se recuperaron con un alto nivel de confidencia. Como era de esperarse, en todos los casos se mostró una dependencia espacial similar de los errores, demarcando un plano con mayor densidad de eventos de baja calidad y coincidente con el plano que contiene a los dos pozos verticales. Esto está en concordancia con los resultados hallados por varios autores. Particularmente, la red entrenada para el modelo 1 tiene mayores dificultades para invertir eventos más cercanos a los receptores. Este fenómeno está relacionado a la cantidad y calidad de los datos extraídos por las ventanas de tiempo. En primer lugar, la cantidad de fases (y por ende, de información) extraídas para un modelo isotrópico es menor que para el caso anisótropo, ya que solo pueden extraerse amplitudes de P y S. En contraparte, para el caso anisotrópico, se tienen fases de P, S_1 y S_2 (o más, para el caso de anisotropías de mayor complejidad). En segundo lugar, recordemos que para cada receptor del arreglo, se puede dibujar una esfera con centro en dicho receptor y con un dado radio "crítico" dentro del cual las ventanas se superpondrán. Dentro de esta esfera, la extracción de las amplitudes son erróneas. El aglutinamiento de estos datos de mala calidad dentro de estas regiones esféricas tiene efecto sobre el entrenamiento, y por lo tanto, sobre el desempeño de la red en esta región. Dado que lo que se tiene es un arreglo de receptores perfectamente vertical, lo que se observa en el primer panel de la Figura 4.5 es el colapso en planta de todas las esferas. Es evidente que de tener un arreglo no-vertical (inclinado, buzante, etc) en el mismo medio isótropo, este fenómeno de la superposición de ventanas aun persistiría, aunque su visualización en planta sería el producto del colapso de cada una de las esferas con centro en los receptores. Por otro lado, si se considera un medio anisotrópico, como los descriptos en los modelos 2 y 3, el fenómeno de superposición de ventanas no se traduciría matemáticamente a una esfera (ni tampoco a un elipsoide), ya que las cantidades $V_P = V_P(\theta)$ y $V_S = V_S(\theta)$ dependen del

ángulo de inclinación θ de la señal (ver Apéndice A o también Grechka (2020)), calculado como el ángulo contenido entre la vertical y el rayo que une la fuente y el receptor*. Por otro lado, existen tres ventanas de tiempo para cada evento, por lo que las zonas de superposición de las 3 ventanas en simultáneo está más restringida que para el caso isótropo.

Resta discutir porqué el desempeño de la red entrenada para el modelo 3, aunque bueno, no es comparable al mostrado por la red del modelo 2. Si consideramos que todas las partes del entrenamiento y los parámetros de la red permanecieron inalteradas, la distinción entre ambos desempeños tiene que tener su origen en la naturaleza del dato. Recordemos que el objetivo de un entrenamiento es discernir un "patrón" general del dato de entrenamiento. En esta línea, no es difícil suponer que el patrón descripto por un conjunto de datos cuyas señales fueron refractadas por una interfaz sea más complejo de descifrar que aquel correspondiente a un medio homogéneo. En efecto, las amplitudes de los eventos pertenecientes al modelo 2 están dominadas, al igual que para el caso isotrópico, por el modelo de velocidades, la geometría fuente-receptor y el tensor momento, mientras que las amplitudes extraídas para el modelo 3 consideran además el efecto de los coeficientes de refracción. Si tenemos en cuenta que los coeficientes de refracción se calculan para un medio anisotrópico, el nivel de complejidad de este dato es claramente superior.

Por otra parte, los métodos de MTI basados en la resolución directa de la ecuación 4.1 necesitan del cálculo de las funciones de Green, a partir del cual se arman las matrices de dicho sistema de ecuaciones. Por las razones ya mencionadas, estos cálculos suelen ser complejos y de alto costo numérico. En contraparte, una vez entrenada sobre un dado medio, una NN entregaría valores del tensor momento sin necesidad de dichos cálculos para la inversión, evitando también el problema de la inestabilidad numérica ya mencionado.

Finalmente, y tal como se mencionó en el capítulo, la arquitectura de la red se eligió luego de realizar numerosos ensayos de tipo "prueba y error". El criterio de aceptación del modelo final no solo se basó en la obtención de buenos valores de error relativo y correlación, sino que también se buscó evitar el fenómeno de *overfitting*.

4.5. Conclusiones

En este capítulo se propone una red neuronal profunda para resolver el problema de MTI en medios 3D isótropos o con anisotropía VTI débil y se prueba utilizando datos sintéticos en un escenario típico de monitoreo de fracturación hidráulica con uno o dos pozos. Como se muestra en los resultados, la NN es capaz predecir la mayoría de los datos de testeo entregando valores de correlación media altos y errores relativos medio bajos

*Este rayo, para un medio de capas no sería recto y, generalmente, presentaría quiebres en sus puntos de refracción

para todos los modelos propuestos. Los resultados también muestran que las inversiones son pobres para aquellos eventos cerca del plano que contiene ambos pozos (o en el caso *single-well*), pero se vuelven muy precisas a medida que se alejan de este plano. En general, demostramos que una NN entrenada es una buena alternativa a los métodos tradicionales de MTI.

Capítulo 5

Conclusiones

5.1. Conclusiones generales

El objetivo principal de esta Tesis consistía en el desarrollo de algoritmos especializados para resolver el tratamiento de la información microsísmica con aplicación al estudio y monitoreo de procesos involucrados en la exploración y explotación de reservorios de hidrocarburos no convencionales. Así, las diferentes estrategias y herramientas que se muestran en este documento significan avances en el estado del arte del procesamiento de las señales microsísmicas y su implementación representa un aporte frente a la técnicas usualmente adoptadas por la industria.

Los resultados de esta Tesis se obtuvieron considerando únicamente geometrías de monitoreo de pozo. Particularmente, solo se utilizaron geometrías con uno o dos pozos verticales, que son las más utilizadas para este tipo de trabajos. Sin embargo, la mayoría de las estrategias aquí desarrolladas pueden aplicarse, con algunas o pocas modificaciones, a geometrías menos convencionales, como pozos verticales altamente desviados, pozos horizontales, arreglos de tipo *multi-well*, o incluso arreglos de superficie. En cuanto a los medios geológicos considerados a lo largo de este trabajo, se utilizaron medios con diversas complejidades, como un medio homogéneo e isótropo de una capa y dos medios con anisotropía VTI de una y dos capas, respectivamente. Es importante mencionar que los métodos desarrollados en esta Tesis pueden ser aplicados en otras geometrías (más de dos capas, por ejemplo) de manera directa.

Cabe mencionar que la mayor parte de los métodos aquí presentados fueron programados enteramente durante el transcurso de esta Tesis. Los mismos fueron programados utilizando lenguajes de programación modernos como Julia o, en menor medida, Python y Fortran. En algunos casos, para el desarrollo de ciertos algoritmos se necesitó de la integración de programas pre-existentes. Esta integración requirió de trabajos de adaptación y, en su mayoría, de traducción a un lenguaje de programación común (Julia y Python).

Como resultado, se cuenta ahora con programas en Julia para: (1) localizar eventos microsísmicos, (2) realizar filtrados mediante los métodos basados en la transformada de Radon, descomposición en valores singulares, y descomposición en modos empíricos, (3) efectuar histogramas y el análisis de sus diferencias, y (4) inversión de tensor momento mediante la implementación de redes neuronales a través del módulo "Flux.jl". También se cuenta con programas para la creación de datos sintéticos en medios VTI considerando capas planas y horizontales bajo cualquier geometría de monitoreo. Parte de estos programas utilizan rutinas del modulo "SeisJulia.jl", ya programadas en este lenguaje. En Pyhton, por su parte, destacamos rutinas para la generación de sintéticos y gráficas de trazas sísmicas. Todas las subrutinas fueron programadas para funcionar considerando datos sintéticos (ya sea limpios o contaminados con ruido sintético o de campo) o reales.

Los resultados son presentados en los Capítulos 2, 3 y 4. Los Apéndices A y B, por su parte, contienen los fundamentos necesarios para la comprensión del Capítulo 4. En el Capítulo 2 se presenta un método que permite comparar el desempeño de diferentes algoritmos de filtrado de ruido. Este método se basa en el análisis de las diferencias de los histogramas de polarización construidos a partir de las señales que fueron filtradas y sus señales limpias. Este análisis se aplica, en principio, a datos sintéticos donde se conoce el dato libre de ruido. Sin embargo, logramos adaptar el método para ser utilizado también como herramienta de evaluación sobre datos reales. Además, demostramos que es posible realizar un análisis cuantitativo y comparativo entre diferentes métodos de filtrado, lo cual nos permite decidir cuál es el que mejor realiza su tarea. Como resultado anexo, encontramos que el método de SVD obtiene los mejores resultados tanto sobre datos sintéticos como sobre datos de campo.

En el Capítulo 3 se desarrolla un algoritmo para la localización de eventos microsísmicos mediante el método de Evolución diferencial. El mismo es un método de optimización heurístico utilizado para encontrar el máximo o mínimo global de una función de costo. Así, lo utilizamos para resolver el problema de la localización espacial de un evento microsísmico, que constituye un problema altamente no-lineal. Su aplicación arrojó resultados satisfactorios. Como parte de las conclusiones encontramos que el método es capaz de localizar eventos con gran precisión y con velocidades de cómputo comparables a las obtenidas mediante otros métodos publicados en trabajos científicos de alcance internacional.

Finalmente, el Capítulo 4 está dedicado al desarrollo y la implementación de la técnica de redes neuronales sobre el problema de la inversión de tensor momento. Los resultados nos permiten asegurar que es posible recuperar los mecanismos de fractura de los eventos microsísmicos generados durante un proceso de fracturación hidráulica de manera precisa. También, como resultado de nuestro análisis encontramos que, bajo una geometría de doble pozo monitor, existe una zona donde la red entrenada tiene dificultades para predecir estos mecanismos. Dicha zona se encuentra definida por las cercanías del plano que contiene los

dos pozos verticales. Este resultado no solo reafirma los resultados numéricos encontrados por otros autores, sino que también está en coincidencia con la teoría que describe el problema de inversión de tensor momento. Por completitud, y en el mismo capítulo, entrenamos una red para el caso de un único pozo monitor desviado, encontrando resultados similares. Así, mostramos que las redes neuronales son una herramienta capaz de realizar esta tarea entregando valores de errores relativos medios bajos y pueden considerarse como una alternativa a los métodos convencionales de inversión de tensor momento.

5.2. Contribuciones científicas

Todos los resultados presentados en esta Tesis fueron presentados parcial o totalmente en documentos científicos y/o presentaciones en congresos. Particularmente, se escribieron 3 artículos en revistas de circulación internacional, los cuales fueron presentados en 3 congresos científicos nacionales e internacionales. Todos ellos fueron escritos con primera autoría. En detalle, la estrategia de evaluación de filtrado de ruido mediante histogramas de polarización, descripta en el Capítulo 2, fue publicada (Brunini et al., 2021a) en la revista de circulación internacional y con referato, *Geophysics*:

> Brunini, G. I., J. I. Sabbione, J. L. Gómez, and D. R. Velis, 2021a, Microseismic denoising assessment by polarization histograms: Geophysics, **86**, KS11–KS22.
>
> https://library.seg.org/doi/abs/10.1190/geo2020-0130.1

Los resultados que llevaron a este trabajo también fueron presentados en forma oral en la *AAPG 2019 International Conference & Exhibition* (ICE) (Brunini et al., 2019) celebrada durante el mes de agosto 2019 en Buenos Aires:

> G. I. Brunini, J. I. Sabbione, J. L. Gómez, D. R. Velis. 2019, Comparative analysis of three denoising methods for microseismic data: Radon transform denoising, reduced-rank filtering, and empirical mode decomposition. AAPG 2019 International Conference & Exhibition (ICE).
>
> https://archive.aapg.org/ICE/2019/buenosaires2019.iceevent.org/technical-program/program/friday-oral-presentations.html

Respecto al trabajo que se desarrolló para la localización de eventos microsísmicos implementando el método de evolución diferencial (Capítulo 3) fue presentado en el congreso *2017 XVII Workshop on Information Processing and Control* (RPIC) y posteriormente aceptado para su publicación como resumen expandido (Brunini et al., 2017) en el *proceedings* de la conferencia (con referato):

Brunini, G. I., J. I. Sabbione, and D. R. Velis, 2017, Differential evolution for microseismic event location: Proceedings of the 2017 XVII Workshop on Information Processing and Control (RPIC), IEEE, 1–6.

https://ieeexplore.ieee.org/abstract/document/8214316

Por otro lado, el trabajo que se presenta en el Capítulo 4, donde se analiza el desempeño de una red neuronal para la inversión del tensor momento en diferentes medios y geometrías de adquisición, fué presentado oralmente en el congreso *2021 XIX Workshop on Information Processing and Control* (RPIC) y publicado como resumen expandido (Brunini et al., 2021b) en el *proceedings* de la conferencia (con referato):

Brunini, G. I., D. R. Velis, and J. I. Sabbione, 2021b, Seismic moment tensor inversion in anisotropic media using deep neural networks: Proceedings of the 2021 XIX Workshop on Information Processing and Control (RPIC), IEEE, 1–6.

https://ieeexplore.ieee.org/abstract/document/9648414

En el transcurso de esta Tesis también se presentaron trabajos en modalidad poster (Brunini, 2020) y presentación corta (Brunini et al., 2020) y se realizó una colaboración en el artículo (Velis et al., 2022):

Velis, D. R., Gómez, J. L., Gelpi, G. R., Brunini, G. I., Pérez, D. O., & Sabbione, J. I. (2022). Aprendizaje automático para análisis y procesamiento de datos sísmicos. Geoacta, 43(2), 7–29.

https://revistas.unlp.edu.ar/geoacta/article/view/14284

Por último, se escribió un resumen expandido (Brunini García et al., 2021) titulado:

Brunini García, G. I., J. I. Sabbione, and D. R. Velis, 2021, Analyzing the time saving of microseismic finite differences modeling when using an expanding box algorithm: Presented at the 17th International Congress of the Brazilian Geophysical Society; 2021. Brazilian Geophysical Society.

https://researchgate.net/publication/358449336_Analyzing_the_time_saving_of_microseismic_finite_differences_modeling_when_using_an_expanding_box_algorithm

En este trabajo se desarrolló un método que permite reducir los tiempos de cómputo en la utilización del método de diferencias finitas para la propagación de ondas microsísmicas en medios 2D (y 3D). Para ello se vale de una estrategia de tipo *expanding-box* y se asume una fuente de tipo explosiva. Este algoritmo no se utilizó para el desarrollo de ninguno de los trabajos presentados en esta tesis, por lo que no fué incluido como un capítulo adicional de la misma.

5.3. Contribuciones tecnológicas y desarrollos

Todos los algoritmos y métodos descriptos en este trabajo de Tesis fueron llevados a cabo mediante la implementación de códigos en lenguaje Julia, y en menor medida Python y Fortran 90. Entre los códigos desarrollados, destacamos:

- Código en lenguaje Fortran para la implementación del algoritmo de evolución diferencial, que consiste en una subrutina principal que llama diferentes subrutinas para ser utilizadas como herramientas auxiliares. Entre estas subrutinas se destacan la implementación de un algoritmo de trazado de rayos anisotrópicos para el calculo de tiempos de arribo sintéticos y la subrutina que aplica el método de evolución diferencial propiamente dicho.

- Código en lenguaje Julia para el cálculo de atributos de polarización utilizando ventanas de tiempo y dato de tres componentes.

- Código en lenguaje Julia para obtener sintéticos mediante trazado de rayos anisotrópico VTI con amplitudes y medios estratificados

- Código en Julia para el filtrado simultáneo de ruido mediante los métodos de RHRT, SVD, EMD y BPF. Para el ensamble de este código fue necesario la traducción de las rutinas de EMD y SVD, que fueron provistas por el Dr. Julián L. Gómez y el Dr. Danilo R. Velis, en los lenguajes Pyhton y Fortran, respectivamente. Además fue necesario la adaptación de la rutina de RHRT, desarrollada originalmente en el lenguaje Julia por el Dr. Juan I. Sabbione.

- Código en Julia para la construcción de histogramas de polarización y sus respectivas diferencias.

- Código en Julia para la implementación de redes neuronales aplicado a la problemática de la inversión de tensor momento mediante cualquier geometría de adquisición. Aquí, destacamos el uso del módulo "Flux.jl" (*Flux: The Julia Machine Learning Library*), que como su nombre lo indica, es una librería programada íntegramente en Julia con múltiples herramientas del aprendizaje automático integradas y que, entre sus aplicaciones, permite la implementación de redes neuronales en forma limpia, eficiente y flexible. La información pertinente a este módulo puede encontrarse en https://fluxml.ai/Flux.jl/stable/.

- Código en Python para graficar arribos de señales microsísmicas (*gathers*) e histogramas, entre otros.

- Destacamos la librería "SeisMain.jl", desarrollada por el grupo *Signal analysis & imaging group* de la universidad de Alberta, Canadá (https://github.com/SeismicJulia/SeisMain.jl). La misma contiene una innumerable cantidad de rutinas de gran utilidad para el procesamiento y visualización de datos sísmicos que están íntegramente desarrolladas en Julia. Muchas de ellas fueron utilizadas en esta Tesis como herramientas que se integraron dentro de los programas principales.

Bibliografía

Aggarwal, C. C., 2018, Neural networks and deep learning: Springer, **10**, 978–3.

Aki, K., and P. G. Richards, 2002, Quantitative seismology: University Science Books.

Akram, J., and D. W. Eaton, 2016, A review and appraisal of arrival-time picking methods for downhole microseismic data arrival-time picking methods: Geophysics, **81**, KS71–KS91.

Akram, J., O. Ovcharenko, and D. Peter, 2017, A robust neural network-based approach for microseismic event detection, *in* SEG Technical Program Expanded Abstracts 2017: Society of Exploration Geophysicists, 2929–2933.

Ali, M., and A. Aftab, 2020, Topic review unconventional reservoirs subjects: Energy & fuels: Energy & Fuel Technology View Times, **66**.

Athey, S., J. Tibshirani, and S. Wager, 2019, Generalized random forests: Annals of Statistics, **47**, 1148–1178.

Baig, A., and T. Urbancic, 2010, Microseismic moment tensors: A path to understanding frac growth: The Leading Edge, **29**, 320–324.

Baig, A. M., T. Urbancic, and G. Viegas, 2012, Do hydraulic fractures induce events large enough to be felt on surface?: CSEG Recorder, **10**, 40–46.

Barredo, S., and L. Stinco, 2013, *in* A Geodynamic View of Oil and Gas Resources Associated to the Unconventional Shale Reservoirs of Argentina: Society of Exploration Geophysicists, 832–841.

Bekara, M., and M. Van der Baan, 2009, Random and coherent noise attenuation by empirical mode decomposition: Geophysics, **74**, V89–V98.

Bezdek, J. C., S. K. Chuah, and D. Leep, 1986, Generalized k-nearest neighbor rules: Fuzzy Sets and Systems, **18**, 237–256.

Binder, G., 2018, Neural networks for moment-tensor inversion of surface microseismic data, *in* SEG Technical Program Expanded Abstracts 2018: Society of Exploration Geophysicists, 2917–2921.

Binder, G., and A. Tura, 2020, Convolutional neural networks for automated microseismic detection in downhole distributed acoustic sensing data and comparison to a surface geophone array: Geophysical Prospecting, **68**, 2770–2782.

Blias, E., and V. Grechka, 2013, Analytic solutions to the joint estimation of microseismic

event locations and effective velocity model: Geophysics, **78**, KS51–KS61.

Breiman, L., 2001, Random forests: Machine learning, **45**, 5–32.

Brunini, G. I., 2020, Caracterización de eventos microsísmicos: Presented at the Encuentro de Becaries de Posgrado de la UNLP (EBEC)(Modalidad virtual, 12 de noviembre de 2020), UNLP.

Brunini, G. I., J. I. Sabbione, J. L. Gómez, and D. R. Velis, 2019, Comparative analysis of three denoising methods for microseismic data: Radon transform denoising, reduced-rank filtering, and empirical mode decomposition: Presented at the AAPG 2019 International Conference & Exhibition (ICE), AAPG.

———, 2021a, Microseismic denoising assessment by polarization histograms: Geophysics, **86**, KS11–KS22.

Brunini, G. I., J. I. Sabbione, and D. R. Velis, 2017, Differential evolution for microseismic event location: Proceedings of the 2017 XVII Workshop on Information Processing and Control (RPIC), IEEE, 1–6.

Brunini, G. I., D. R. Velis, and J. I. Sabbione, 2020, Caracterización de eventos microsísmicos: Procesamiento y algoritmos: Investigación Joven, **7**, 101–102.

———, 2021b, Seismic moment tensor inversion in anisotropic media using deep neural networks: Proceedings of the 2021 XIX Workshop on Information Processing and Control (RPIC), IEEE, 1–6.

Brunini García, G. I., J. I. Sabbione, and D. R. Velis, 2021, Analyzing the time saving of microseismic finite differences modeling when using an expanding box algorithm: Presented at the 17th International Congress of the Brazilian Geophysical Society; 2021. Brazilian Geophysical Society, Brazilian Geophysical Society.

Bzdok, D., M. Krzywinski, and N. Altman, 2018, Machine learning: supervised methods: Nature methods, **15**, 5.

Carbone, O., G. Vergani, and A. Giusiano, 2020a, Neuquén. a un siglo del descubrimiento del petróleo.¿ por qué fue estatal?: Revista Facultad de Ciencias Exactas, Físicas y Naturales, **7**, 173–184.

———, 2020b, Neuquén. a un siglo del descubrimiento del petróleo. ¿por qué fue estatal?: Revista de la Facultad de Ciencias Exactas, Físicas y Naturales, **7**, 173–184.

Carcione, J. M., 2007, Wave fields in real media: Wave propagation in anisotropic, anelastic, porous and electromagnetic media: Elsevier.

Castano, A. F., C. H. Sondergeld, and C. S. Rai, 2010, Estimation of uncertainty in microseismic event location associated with hydraulic fracturing: Presented at the Tight Gas Completions Conference, OnePetro.

Cerveny, V., 2005, Seismic ray theory: Cambridge University Press.

Chapman, C., 2004, Fundamentals of seismic wave propagation: Cambridge University Press.

Chen, K., and M. D. Sacchi, 2014, Robust reduced-rank filtering for erratic seismic noise attenuation: Geophysics, **80**, V1–V11.

Chen, S. S., D. L. Donoho, and M. A. Saunders, 2001, Atomic decomposition by basis pursuit: SIAM Review, **43**, 129–159.

Chen, Y., 2018, Fast waveform detection for microseismic imaging using unsupervised machine learning: Geophysical Journal International, **215**, 1185–1199.

———, 2020, Automatic microseismic event picking via unsupervised machine learning: Geophysical Journal International, **222**, 1750–1764.

Chorney, D., P. Jain, M. Grob, and M. van der Baan, 2012, Geomechanical modeling of rock fracturing and associated microseismicity: The Leading Edge, **31**, 1348–1354.

Cipolla, C., M. Mack, and S. Maxwell, 2010, Reducing exploration and appraisal risk in low permeability reservoirs using microseismic fracture mapping—part 2: Presented at the SPE Latin American and Caribbean Petroleum Engineering Conference, OnePetro.

Cipolla, C., S. Maxwell, M. Mack, and R. Downie, 2012, A practical guide to interpreting microseismic measurements: Presented at the SPE/EAGE European Unconventional Resources Conference & Exhibition-From Potential to Production.

Cipolla, C. L., R. E. Lewis, S. C. Maxwell, and M. G. Mack, 2011, Appraising unconventional resource plays: Separating reservoir quality from completion effectiveness: Presented at the International petroleum technology conference, International Petroleum Technology Conference.

Colominas, M. A., G. Schlotthauer, and M. E. Torres, 2014, Improved complete ensemble EMD: A suitable tool for biomedical signal processing: Biomedical Signal Processing and Control, **14**, 19–29.

Downie, R., E. Kronenberger, and S. C. Maxwell, 2010, Using microseismic source parameters to evaluate the influence of faults on fracture treatments-a geophysical approach to interpretation: Presented at the SPE Annual Technical Conference and Exhibition, OnePetro.

Drew, J. E., H. D. Leslie, P. N. Armstrong, and G. Michard, 2005, Automated microseismic event detection and location by continuous spatial mapping: Presented at the SPE annual technical conference and exhibition, OnePetro.

Du, J., and N. R. Warpinski, 2011, Uncertainty in fpss from moment-tensor inversion: Geophysics, **76**, WC65–WC75.

Du, J., U. Zimmer, and N. Warpinski, 2011, Fault-plane solutions from moment tensor inversion for microseismic events using single-well and multi-well data: CSEG Recorder, **36**, 22–28.

Duncan Peter, M., and L. Eisner, 2010, Reservoir characterization using surface microseismic monitoring: Geophysics, **75**, 139–146.

Eaton, D. W., and F. Forouhideh, 2011, Solid angles and the impact of receiver-array

geometry on microseismic moment-tensor inversion: Geophysics, **76**, WC77–WC85.

Eisner, L., D. Abbott, W. B. Barker, J. Lakings, and M. P. Thornton, 2008, Noise suppression for detection and location of microseismic events using a matched filter, *in* SEG Technical Program Expanded Abstracts 2008: Society of Exploration Geophysicists, 1431–1435.

Eisner, L., P. M. Duncan, W. M. Heigl, and W. R. Keller, 2009, Uncertainties in passive seismic monitoring: The Leading Edge, **28**, 648–655.

Eisner, L., V. Grechka, and S. Williams-Stroud, 2010a, Future of microseismic analysis: Integration of monitoring and reservoir simulation: Presented at the AAPG Hedberg Conference.

Eisner, L., M. Thornton, and J. Griffin, 2011a, Challenges for microseismic monitoring, *in* SEG Technical Program Expanded Abstracts 2011: Society of Exploration Geophysicists, 1519–1523.

Eisner, L., M. P. Thornton, and J. Griffin, 2011b, Challenges for microseismic monitoring: 81st Annual International Meeting, Expanded Abstracts, Society of Exploration Geophysicists, 1519–1523.

Eisner, L., S. William-Stroud, A. Hill, P. Duncan, and M. Thornton, 2010b, Beyond dots in the box: microseismicity-constrained fracture models for reservoir simulation: The Leading Edge, **29**, 326–333.

Flinn, E., 1965, Signal analysis using rectilinearity and direction of particle motion: Proceedings of the IEEE, **53**, 1874–1876.

Gentleman, R., and V. J. Carey, 2008, Unsupervised machine learning, *in* Bioconductor case studies: Springer, 137–157.

Géron, A., 2019, Hands-on machine learning with scikit-learn, keras, and tensorflow: Concepts, tools, and techniques to build intelligent systems: O'Reilly Media.

Ghahramani, Z., 2003, Unsupervised learning: Summer School on Machine Learning, Springer, 72–112.

Golub, G., and C. Van Loan, 1989, Matrix computations, 2nd ed.: John Hopkins University Press.

Gómez, J. L., and D. R. Velis, 2016, A simple method inspired by empirical mode decomposition for denoising seismic data: Geophysics, **81**, V403–V413.

Gómez, J. L., D. R. Velis, and J. I. Sabbione, 2020, Noise suppression in 2D and 3D seismic data with data-driven sifting algorithms: Geophysics, **85**, V1–V10.

Goodfellow, I., Y. Bengio, A. Courville, and Y. Bengio, 2016, Deep learning: MIT press Cambridge.

Goodway, B., 2012, Introduction to this special section: Passive seismic and microseismic - Part 1: The Leading Edge, **31(11)**, 1296–1299.

Grechka, V., 2015a, Moment tensor inversion of single-well microseismic data: Is it fea-

sible?, *in* SEG Technical Program Expanded Abstracts 2015: Society of Exploration Geophysicists, 2506–2511.

———, 2015b, On the feasibility of inversion of single-well microseismic data for full moment tensor: Geophysics, **80**, KS41–KS49.

———, 2020, Anisotropy and microseismics: Theory and practice: Society of Exploration Geophysicists.

Grechka, V. I., and W. M. Heigl, 2017, Microseismic monitoring: Society of Exploration Geophysicists Tulsa, OK.

Gupta, I., C. Rai, D. Devegowda, and C. H. Sondergeld, 2021, Fracture hits in unconventional reservoirs: A critical review: SPE Journal, **26**, 412–434.

Hafner, J., H. S. Sawhney, W. Equitz, M. Flickner, and W. Niblack, 1995, Efficient color histogram indexing for quadratic form distance functions: IEEE Transactions on Pattern Analysis and Machine Intelligence, **17**, 729–736.

Hamerly, G., and C. Elkan, 2004, Learning the k in k-means: Advances in neural information processing systems, **16**, 281–288.

Han, J., and M. van der Baan, 2015, Microseismic and seismic denoising via ensemble empirical mode decomposition and adaptive thresholding: Geophysics, **80**, KS69–KS80.

Hardebeck, J. L., and P. M. Shearer, 2003, Using s/p amplitude ratios to constrain the focal mechanisms of small earthquakes: Bulletin of the Seismological Society of America, **93**, 2434–2444.

Hastie, T., R. Tibshirani, and J. Friedman, 2009, The elements of statistical learning: data mining, inference, and prediction: Springer Science & Business Media.

He, Z., J. Li, L. Liu, and Y. Shen, 2017, Three-dimensional empirical mode decomposition (TEMD): A fast approach motivated by separable filters: Signal Processing, **131**, 307–319.

Hecht-Nielsen, R., 1992, Theory of the backpropagation neural network, *in* Neural networks for perception: Elsevier, 65–93.

Hestenes, M., and E. Stiefel, 1952, Methods of conjugate gradients for solving linear systems: Journal of Research of the National Bureau of Standards, **49**, 409–436.

Holditch, S. A., and H. Madani, 2010, Global unconventional gas-it is there, but is it profitable?: Journal of petroleum Technology, **62**, 42–48.

Howley, T., M. G. Madden, M.-L. O'Connell, and A. G. Ryder, 2005, The effect of principal component analysis on machine learning accuracy with high dimensional spectral data: International Conference on Innovative Techniques and Applications of Artificial Intelligence, Springer, 209–222.

Hu, Y. H., J.-N. Hwang, and S. W. Perry, 2002, Handbook of Neural Network Signal Processing : The Journal of the Acoustical Society of America, **111**, 2525–2526.

Huang, L., J. Li, H. Hao, and X. Li, 2018, Micro-seismic event detection and location

in underground mines by using convolutional neural networks (cnn) and deep learning: Tunnelling and Underground Space Technology, **81**, 265–276.

Huang, N. E., Z. Shen, S. R. Long, M. C. Wu, H. H. Shih, Q. Zheng, N.-C. Yen, C. C. Tung, and H. H. Liu, 1998, The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis: Proceedings of the Royal Society of London A: mathematical, physical and engineering sciences, The Royal Society, 903–995.

Huang, N. E., and Z. Wu, 2008, A review on Hilbert-Huang transform: Method and its applications to geophysical studies: Reviews of Geophysics, **46**, 1–23.

Ionescu, C., O. Vantzos, and C. Sminchisescu, 2015, Matrix backpropagation for deep networks with structured layers: Proceedings of the IEEE International Conference on Computer Vision, 2965–2973.

Jacot, A., F. Gabriel, and C. Hongler, 2018, Neural tangent kernel: Convergence and generalization in neural networks: arXiv preprint arXiv:1806.07572.

Jakubovitz, D., R. Giryes, and M. R. Rodrigues, 2019, Generalization error in deep learning, *in* Compressed Sensing and Its Applications: Springer, 153–193.

Jones, J. P., D. W. Eaton, and E. Caffagni, 2016, Quantifying the similarity of seismic polarizations: Geophysical Journal International, **204**, 968–984.

Jurkevics, A., 1988, Polarization analysis of three-component array data: Bulletin of the Seismological Society of America, **78**, 1725–1743.

Kamata, M., L. Nutt, and W. Underhill, 2008, Improving sensor technology brings a new level of reservoir understanding, *in* SEG Technical Program Expanded Abstracts 2008: Society of Exploration Geophysicists, 168–172.

Kanamori, H., 1977, The energy release in great earthquakes: Journal of geophysical research, **82**, 2981–2987.

Kassambara, A., 2017, Practical guide to cluster analysis in r: Unsupervised machine learning: Sthda.

Keller, J. M., M. R. Gray, and J. A. Givens, 1985, A fuzzy k-nearest neighbor algorithm: IEEE transactions on systems, man, and cybernetics, 580–585.

Kendall, J.-M., A. Butcher, A. L. Stork, J. P. Verdon, R. Luckett, and B. J. Baptie, 2019, How big is a small earthquake? challenges in determining microseismic magnitudes: First Break, **37**, 51–56.

Kendall, M., S. Maxwell, G. Foulger, L. Eisner, and Z. Lawrence, 2011, Microseismicity: Beyond dots in a box — Introduction: Geophysics, **76**, WC1–WC3.

Kingma, D. P., and J. Ba, 2014, Adam: A method for stochastic optimization: arXiv preprint arXiv:1412.6980.

Kingsford, C., and S. L. Salzberg, 2008, What are decision trees?: Nature biotechnology, **26**, 1011–1013.

Köhn, H.-F., and L. J. Hubert, 2014, Hierarchical cluster analysis: Wiley StatsRef: Statis-

tics Reference Online, 1–13.

Kotsiantis, S. B., I. Zaharakis, and P. Pintelas, 2007, Supervised machine learning: A review of classification techniques: Emerging artificial intelligence applications in computer engineering, **160**, 3–24.

Lagos, S. R., J. I. Sabbione, and D. R. Velis, 2014, Very fast simulated annealing and particle swarm optimization for microseismic event location, *in* SEG Technical Program Expanded Abstracts 2014: Society of Exploration Geophysicists, 2188–2192.

Lanusse, I., D. Garcia, M. Di Benedetto, and G. Bottesi, 2012, Vaca Muerta formation: From world class source rock to world class shale play: Presented at the American Association of Petroleum Geologists. International Conference.

Leaney, W. S., 2014, Microseismic source inversion in anisotropic media: PhD thesis, University of British Columbia.

Likas, A., N. Vlassis, and J. J. Verbeek, 2003, The global k-means clustering algorithm: Pattern recognition, **36**, 451–461.

Ling, H., and K. Okada, 2007, An efficient earth mover's distance algorithm for robust histogram comparison: IEEE transactions on Pattern Analysis and Machine Intelligence, **29**, 840–853.

Lloyd, S., M. Mohseni, and P. Rebentrost, 2013, Quantum algorithms for supervised and unsupervised machine learning: arXiv preprint arXiv:1307.0411.

Loussaief, S., and A. Abdelkrim, 2018, Convolutional neural network hyper-parameters optimization based on genetic algorithms: International Journal of Advanced Computer Science and Applications, **9**.

Maimon, O., and L. Rokach, 2005, Introduction to supervised methods, *in* Data mining and knowledge discovery handbook: Springer, 149–164.

Marín-Reyes, P. A., J. Lorenzo-Navarro, and M. Castrillón-Santana, 2016, Comparative study of histogram distance measures for re-identification: arXiv preprint arXiv:1611.08134.

Maxwell, S., 2005, A brief guide to passive seismic monitoring: CSEG National Convention, 177–178.

———, 2009a, Microseismic location uncertainty: CSEG Recorder, 177–188.

———, 2009b, Microseismic location uncertainty: CSEG Recorder, **34**, 41–46.

———, 2011, What does microseismic tell us about hydraulic fractures?: 81st Annual International Meeting, Expanded Abstracts, Society of Exploration Geophysicists, 1565–1569.

———, 2014, Microseismic imaging of hydraulic fracturing: improved engineering of unconventional shale reservoirs: Society of Exploration Geophysicists. Distinguished Instructor Series No. 17.

Maxwell, S., M. Jones, R. Parker, S. Miong, S. Leaney, D. Dorval, D. DÁmico, J. Logel, E. Anderson, and K. Hammermaster, 2009, Fault activation during hydraulic fracturing, *in*

SEG Technical Program Expanded Abstracts 2009: Society of Exploration Geophysicists, 1552–1556.

Maxwell, S. C., J. Rutledge, R. Jones, and M. Fehler, 2010, Petroleum reservoir characterization using downhole microseismic monitoring: Geophysics, **75**, A129–A137.

Maxwell, S. C., and T. I. Urbancic, 2001, The role of passive microseismic monitoring in the instrumented oil field: The Leading Edge, **20**, 636–639.

McCulloch, W. S., and W. Pitts, 1943, A logical calculus of the ideas immanent in nervous activity: The bulletin of mathematical biophysics, **5**, 115–133.

Michaud, G., and S. Leaney, 2008, Continuous microseismic mapping for real-time event detection and location, *in* SEG Technical Program Expanded Abstracts 2008: Society of Exploration Geophysicists, 1357–1361.

Mirjalili, S., 2019, Evolutionary algorithms and neural network, *in* Studies in computational intelligence: Springer.

Mishachev, N., 2017, Backpropagation in matrix notation: arXiv preprint arXiv:1707.02746.

Montalbetti, J. F., and E. R. Kanasewich, 1970, Enhancement of teleseismic body phases with a polarization filter: Geophysical Journal International, **21**, 119–129.

Montgomery, D. C., E. A. Peck, and G. G. Vining, 2021, Introduction to linear regression analysis: John Wiley & Sons.

Mousavi, S. M., C. A. Langston, and S. P. Horton, 2016, Automatic microseismic denoising and onset detection using the synchrosqueezed continuous wavelet transform: Geophysics, **81**, V341–V355.

Neri, F., and V. Tirronen, 2010, Recent advances in differential evolution: a survey and experimental analysis: Artificial Intelligence Review, **33**, 61–106.

Neyshabur, B., S. Bhojanapalli, D. McAllester, and N. Srebro, 2017, Exploring generalization in deep learning: arXiv preprint arXiv:1706.08947.

Nolen-Hoeksema, R. C., and L. J. Ruff, 1999, Moment tensor inversion of microseismic events from hydrofractures, *in* SEG Technical Program Expanded Abstracts 1999: Society of Exploration Geophysicists, 1779–1782.

———, 2001, Moment tensor inversion of microseisms from the b-sand propped hydrofracture, m-site, colorado: Tectonophysics, **336**, 163–181.

Ortiz, A. C., D. E. Hryb, J. R. Martínez, and R. A. Varela, 2016, Hydraulic fracture height estimation in an unconventional vertical well in the Vaca Muerta formation, Neuquen basin, Argentina: Presented at the SPE Hydraulic Fracturing Technology Conference, OnePetro.

Otharán, G., 2020, Sedimentología y análisis de facies de la formación vaca Muerta (tithoniano-valanginiano), cuenca neuquina. el rol de los flujos de fango en la depositación de espesas sucesiones de lutitas: PhD thesis, Universidad Nacional del Sur.

Ovcharenko, O., J. Akram, and D. Peter, 2018, Feasibility of moment tensor inversion from a single borehole data using artificial neural networks: Search and Discovery.

Parolai, S., 2009, Denoising of seismograms using the S transform: Bulletin of the Seismological Society of America, **99**, 226–234.

Patterson, J., and A. Gibson, 2017, Deep learning: A practitioner's approach: O'Reilly Media, Inc.

Pei, D., and N. Warpinski, 2015, Downhole microseismic moment tensor inversion by damped least-squares, *in* SEG Technical Program Expanded Abstracts 2015: Society of Exploration Geophysicists, 2532–2536.

Pele, O., and M. Werman, 2010, The quadratic-chi histogram distance family: European Conference on Computer Vision, Springer, 749–762.

Petersen, K., and M. Pedersen, 2008, The matrix cookbook, vol. 7: Technical University of Denmark, **15**.

Pinnegar, C. R., and D. W. Eaton, 2003, Application of the S transform to prestack noise attenuation filtering: Journal of Geophysical Research: Solid Earth, **108**.

Poliannikov, O., A. Malcolm, M. Prange, and H. Djikpesse, 2012, Checking up on the neighbors: Quantifying uncertainty in relative event location: The Leading Edge, **31**, 1490–1494.

Press, W. H., S. Teukolsky, W. Vetterling, and B. Flannery, 1992, Numerical recipes in FORTRAN: The art of scientific computing, 2nd ed.: Cambridge University Press.

Price, K., R. M. Storn, and J. A. Lampinen, 2006, Differential evolution: a practical approach to global optimization: Springer Science & Business Media.

Qian, N., 1999, On the momentum term in gradient descent learning algorithms: Neural networks, **12**, 145–151.

Qin, A. K., and P. N. Suganthan, 2005, Self-adaptive differential evolution algorithm for numerical optimization: Evolutionary Computation, 2005. The 2005 IEEE Congress on, IEEE, 1785–1791.

Qu, S., Z. Guan, E. Verschuur, and Y. Chen, 2020, Automatic high-resolution microseismic event detection via supervised machine learning: Geophysical Journal International, **222**, 1881–1895.

Quinlan, J. R., 1986, Induction of decision trees: Machine learning, **1**, 81–106.

Rilling, G., P. Flandrin, and P. Gonçalves, 2003, On empirical mode decomposition and its algorithms: IEEE-EURASIP Workshop on Nonlinear Signal and Image Processing, NSIP-03, Grado (I), 8–11.

Rojas, R., 1996, The backpropagation algorithm, *in* Neural networks: Springer, 149–182.

Rokach, L., and O. Maimon, 2005, Decision trees, *in* Data mining and knowledge discovery handbook: Springer, 165–192.

Romero-Sarmiento, M.-F., S. Ramiro-Ramirez, G. Berthe, M. Fleury, and R. Littke, 2017,

Geochemical and petrophysical source rock characterization of the Vaca Muerta Formation, Argentina: Implications for unconventional petroleum resource estimations: International Journal of Coal Geology, **184**, 27–41.

Rosenblatt, F., 1958, The perceptron: a probabilistic model for information storage and organization in the brain.: Psychological review, **65**, 386.

Ross, Z. E., M.-A. Meier, E. Hauksson, and T. H. Heaton, 2018, Generalized seismic phase detection with deep learning: Bulletin of the Seismological Society of America, **108**, 2894–2901.

Ruder, S., 2016, An overview of gradient descent optimization algorithms: arXiv preprint arXiv:1609.04747.

Rumelhart, D. E., G. E. Hinton, and R. J. Williams, 1985, Learning internal representations by error propagation: Technical report, California Univ San Diego La Jolla Inst for Cognitive Science.

———, 1986, Learning representations by back-propagating errors: Nature, **323**, 533–536.

Rutledge, J. T., and W. S. Phillips, 2003, Hydraulic stimulation of natural fractures as revealed by induced microearthquakes, Carthage Cotton Valley gas field, east Texas: Geophysics, **68**, 441–452.

Sabbione, J., D. Velis, and M. Sacchi, 2013a, Microseismic data denoising via an apex-shifted hyperbolic Radon transform: SEG Expanded Abstracts, 2155–2161.

Sabbione, J. I., and M. D. Sacchi, 2016, Restricted model domain time Radon transforms: Geophysics, **81**, A17–A21.

Sabbione, J. I., M. D. Sacchi, and D. R. Velis, 2014, Radon transform-based microseismic event detection and signal-to-noise ratio enhancement: Journal of Applied Geophysics, **113**, 51–63.

———, 2015, Radon transform-based microseismic event detection and signal-to-noise ratio enhancement: Journal of Applied Geophysics, **113**, 51–63.

Sabbione, J. I., and D. R. Velis, 2010, Automatic first breaks picking: new strategies and algorithms: Geophysics, **75**, V67–V76.

———, 2012, An automatic method for microseismic events detection based on earthquake phase pickers: 82nd Annual International Meeting, Expanded Abstracts, Society of Exploration Geophysicists, 1–5.

———, 2013, A robust method for microseismic event detection based on automatic phase pickers: Journal of Applied Geophysics, **99**, 42–50.

Sabbione, J. I., D. R. Velis, and M. D. Sacchi, 2013b, Microseismic data denoising via an apex-shifted hyperbolic Radon transform, *in* SEG Technical Program Expanded Abstracts 2013: Society of Exploration Geophysicists, 2155–2161.

Sadowski, P., 2016, Notes on backpropagation: Department of Computer Science University of California Irvine.

Saldungaray, P., and T. T. Palisch, 2012, Hydraulic fracture optimization in unconventional reservoirs: Presented at the SPE Middle East unconventional gas conference and exhibition, OnePetro.

Schaul, T., I. Antonoglou, and D. Silver, 2013, Unit tests for stochastic optimization: arXiv preprint arXiv:1312.6055.

Schimmel, M., and J. Gallart, 2007, Frequency-dependent phase coherence for noise suppression in seismic array data: Journal of Geophysical Research: Solid Earth, **112**, B04303.

Seber, G. A., and A. J. Lee, 2012, Linear regression analysis: John Wiley & Sons.

Shao, J., Y. Wang, Y. Yao, S. Wu, Q. Xue, and X. Chang, 2019, Simultaneous denoising of multicomponent microseismic data by joint sparse representation with dictionary learning: Geophysics, **84**, KS155–KS172.

Shearer, P. M., 1999a, Introduction to seismology: Cambridge University Press.

———, 1999b, Introduction to Seismology: Cambridge University Press.

———, 2009, Introduction to Seismology: Cambridge University Press.

Shemeta, J., and P. Anderson, 2010, It's a matter of size: Magnitude and moment estimates for microseismic data: The Leading Edge, **29**, 296–302.

Simon, D., 2013, Evolutionary optimization algorithms: John Wiley & Sons.

Snyman, J. A., 2005, Practical mathematical optimization: Springer.

Song, F., H. S. Kuleli, M. N. Toksöz, E. Ay, and H. Zhang, 2010, An improved method for hydrofracture-induced microseismic event detection and phase picking: Geophysics, **75**, A47–A52.

Stanton, A., and M. D. Sacchi, 2016, Efficient geophysical research in Julia: CSEG GeoConvention 2016, 1–3.

Stinco, L. P., and S. P. Barredo, 2014, Vaca Muerta formation: An example of shale heterogeneities controlling hydrocarbon accumulations: Unconventional Resources Technology Conference, Denver, Colorado, 25-27 August 2014, Society of Exploration Geophysicists, American Association of Petroleum, 2854–2868.

Stockwell, R. G., L. Mansinha, and R. Lowe, 1996, Localization of the complex spectrum: the S transform: IEEE transactions on Signal Processing, **44**, 998–1001.

Storn, R., 1996, On the usage of differential evolution for function optimization: Proceedings of North American Fuzzy Information Processing, IEEE, 519–523.

Storn, R., and K. Price, 1996, Minimizing the real functions of the ICEC'96 contest by differential evolution: Evolutionary Computation, 1996., Proceedings of IEEE International Conference on, IEEE, 842–844.

Stratta, E., 2013, 100 años refinando petróleo argentino: Petrotecnia, **54**, 92–97.

Suárez-Rivera, R., S. J. Green, J. McLennan, and M. Bai, 2006, Effect of layered heterogeneity on fracture initiation in tight gas shales: Presented at the SPE Annual Technical

Conference and Exhibition, Society of Petroleum Engineers.

Taner, M. T., F. Koehler, and R. Sheriff, 1979, Complex seismic trace analysis: Geophysics, **44**, 1041–1063.

Thorson, J. R., and J. F. Claerbout, 1985, Velocity-stack and slant-stack stochastic inversion: Geophysics, **50**, 2727–2741.

Tieleman, T., and G. Hinton, 2012, Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude: COURSERA: Neural networks for machine learning, **4**, 26–31.

Torres, M. E., M. A. Colominas, G. Schlotthauer, and P. Flandrin, 2011, A complete ensemble empirical mode decomposition with adaptive noise: 2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 4144–4147.

Tsvankin, I., 2012, Seismic signatures and analysis of relection data in anisotropic media: Society of Exploration Geophysicists.

Van Der Baan, M., D. Eaton, and M. Dusseault, 2013a, Effective and sustainable hydraulic fracturing: chapter: Microseismic monitoring developments in hydraulic fracture stimulation, Intech.

———, 2013b, Microseismic monitoring developments in hydraulic fracture stimulation: Presented at the ISRM International Conference for Effective and Sustainable Hydraulic Fracturing, International Society for Rock Mechanics and Rock Engineering.

Vavryčuk, V., 2001, Inversion for parameters of tensile earthquakes: Journal of Geophysical Research: Solid Earth, **106**, 16339–16355.

———, 2007, On the retrieval of moment tensors from borehole data: Geophysical Prospecting, **55**, 381–391.

Vavryčuk, V., and D. Kühn, 2012, Moment tensor inversion of waveforms: a two-step time-frequency approach: Geophysical Journal International, **190**, 1761–1776.

Velis, D., J. I. Sabbione, and M. D. Sacchi, 2015, Fast and automatic microseismic phase-arrival detection and denoising by pattern recognition and reduced-rank filtering: Geophysics, **80**, WC25–WC38.

Velis, D. R., J. L. Gómez, G. R. Gelpi, G. I. Brunini, D. O. Pérez, and J. I. Sabbione, 2022, Aprendizaje automático para análisis y procesamiento de datos sísmicos: Geoacta, **43**, 7–29.

Vera Rodríguez, I., D. Bonar, and M. Sacchi, 2012, Microseismic data denoising using a 3C group sparsity constrained time-frequency transform: Geophysics, **77**, V21–V29.

Vera Rodríguez, I., Y. J. Gu, and M. D. Sacchi, 2011, Resolution of seismic-moment tensor inversions from a single array of receivers: Bulletin of the Seismological Society of America, **101**, 2634–2642.

Verkhovtseva, N., and J. Shaffner, 2013, Advantages and disadvantages of array depth placement and longer tool string apertures: Presented at the GeoConvention 2013: Inte-

gration, geoscience engineering partnership.

Vidale, J. E., 1986, Complex polarization analysis of particle motion: Bulletin of the Seismological Society of America, **76**, 1393–1405.

Wamriew, D. S., M. Charara, and E. Maltsev, 2020, Deep neural network for real-time location and moment tensor inversion of borehole microseismic events induced by hydraulic fracturing: Presented at the SPE Russian Petroleum Technology Conference, Society of Petroleum Engineers.

Wang, H., Q. Zhang, G. Zhang, J. Fang, and Y. Chen, 2020, Self-training and learning the waveform features of microseismic data using an adaptive dictionary: Geophysics, **85**, KS51–KS61.

Warpinski, N., 2009, Microseismic monitoring: Inside and out: Journal of Petroleum Technology, **61**, 80–85.

Warpinski, N. R., and J. Du, 2010a, Source-mechanism studies on microseismicity induced by hydraulic fracturing: Presented at the SPE Annual Technical Conference, Proceedings, Society of Petroleum Engineers.

———, 2010b, Source-mechanism studies on microseismicity induced by hydraulic fracturing: Presented at the SPE Annual Technical Conference and Exhibition, Society of Petroleum Engineers.

Warpinski, N. R., S. L. Wolhart, and C. A. Wright, 2001, Analysis and prediction of microseismicity induced by hydraulic fracturing: Presented at the SPE Annual Technical Conference and Exhibition, Society of Petroleum Engineers.

Weng, X., O. Kresse, C.-E. Cohen, R. Wu, and H. Gu, 2011, Modeling of hydraulic-fracture-network propagation in a naturally fractured formation: SPE Production & Operations, **26**, 368–380.

Wold, S., K. Esbensen, and P. Geladi, 1987, Principal component analysis: Chemometrics and intelligent laboratory systems, **2**, 37–52.

Wu, Y., X. Zhao, R. Zinno, H. Wu, V. Vaidya, M. Yang, and J. Qin, 2016, The application of microseismic monitoring in unconventional reservoirs, *in* Unconventional Oil and Gas Resources Handbook: Elsevier, 243–287.

Wu, Z., and N. E. Huang, 2009, Ensemble empirical mode decomposition: a noise-assisted data analysis method: Advances in Adaptive Data Analysis, **1**, 1–41.

Yilmaz, O., 2001, Seismic data analysis: processing, inversion, and interpretation of seismic data: Society of Exploration Geophysicists. Investigations in Geophysics.

Yrigoyen, M., 1993, The history of hydrocarbons exploration and production in Argentina: Journal of Petroleum Geology, **16**, 371–382.

Yu, X., S. Leaney, J. Rutledge, and C. Chapman, 2016, Multievent moment-tensor inversion for ill-conditioned geometries: Geophysics, **81**, KS11–KS24.

Yu, X., J. Rutledge, and S. Leaney, 2015a, Event grouping and multi-event moment-tensor

inversion for ill-posed monitoring geometry, *in* SEG Technical Program Expanded Abstracts 2015: Society of Exploration Geophysicists, 2522–2526.

———, 2015b, Event grouping and multi-event moment-tensor inversion for ill-possed monitoring geometry: 85th Annual International Meeting, Expanded Abstracts, Society of Exploration Geophysicists, 2522–2526.

Zeiler, M. D., 2012, Adadelta: an adaptive learning rate method: arXiv preprint arXiv:1212.5701.

Zhang, C., S. Bengio, M. Hardt, B. Recht, and O. Vinyals, 2021, Understanding deep learning (still) requires rethinking generalization: Communications of the ACM, **64**, 107–115.

Zhang, C., and M. van der Baan, 2018, Multicomponent microseismic data denoising by 3D shearlet transform: Geophysics, **83**, A45–A51.

———, 2019, Microseismic denoising and reconstruction by unsupervised machine learning: IEEE Geoscience and Remote Sensing Letters, 1–5.

Zhao, Z., and L. Gross, 2017, Using supervised machine learning to distinguish microseismic from noise events, *in* SEG Technical Program Expanded Abstracts 2017: Society of Exploration Geophysicists, 2918–2923.

Zhou, Y., and G. Wu, 2020, Unsupervised machine learning for waveform extraction in microseismic denoising: Journal of Applied Geophysics, **173**, 103879.

Zhu, L., E. Liu, and J. H. McClellan, 2015, Full waveform microseismic inversion using differential evolution algorithm: 2015 IEEE Global Conference on Signal and Information Processing (GlobalSIP), 591–595.

Zhu, W., S. M. Mousavi, and G. C. Beroza, 2019, Seismic signal denoising and decomposition using deep neural networks: IEEE Transactions on Geoscience and Remote Sensing, **57**, 9476–9488.

Zimmer, U., 2010, Localization of microseismic events using headwaves and direct waves, *in* SEG Technical Program Expanded Abstracts 2010: Society of Exploration Geophysicists, 2196–2200.

Zimmer, U., and J. Jin, 2011, Fast search algorithms for automatic localization of microseismic events: CSEG Recorder, **36**, 40–46.

Apéndice A

Tensor momento: El problema directo

Introducción

Sea una fuerza f "puntual" de volumen, unidireccional, y de magnitud variable en el tiempo responsable de producir un desplazamiento en el subsuelo $\mathbf{u}(\mathbf{x}, t)$, con origen en un punto O del espacio dentro de un medio homogéneo, infinito, isotrópico y elástico. Sin pérdida de generalidad, consideremos al punto O como el origen de un sistema de coordenadas cartesiano y la dirección de la fuerza coincidente con el eje x de dicho sistema. Dicho esto, la relación entre la fuerza y el desplazamiento es descripta por la "ecuación de movimiento", expresada como:

$$\rho \ddot{u}_i = f_i + (\lambda + \mu)\, u_{j,ji} + \mu\, u_{i,jj}, \tag{A.1}$$

donde f_i es la magnitud de la fuerza en la dirección i, con $i = 1, 2, 3$ (cualquiera de las 3 direcciones ortogonales del sistema cartesiano), u_i es la componente i-ésima del vector de desplazamiento $\mathbf{u}$, ρ es la densidad del medio, y λ y μ son los módulos de Lamé.

Comenzamos describiendo la solución de la ecuación A.1 para una fuerza puntual, cuya expresión es:

$$\mathbf{f}^{vol}(\mathbf{x}, t) = \delta(\mathbf{x} - \xi)\, \mathbf{I} \cdot \mathbf{f}(\xi, \tau) * \delta(t - \tau), \tag{A.2}$$

donde δ es la función "delta" de Dirac, siendo ξ y τ las coordenadas espaciales y tiempo de origen de la fuente, respectivamente. El tensor $\mathbf{I}$ es la matriz identidad $I^{3\times3}$ y $\mathbf{f}^{vol}$ es una fuerza puntual cuyo supra-índice vol indica que es una "densidad de fuerza" con unidades de "fuerza/volumen" (Grechka and Heigl, 2017). Esto último se deriva de considerar una fuente sísmica encerrada en un volumen de roca vol, delimitado por una superficie S, y que actúa sobre el volumen exterior a esta superficie. La expresión A.2 indica que la fuerza tiene una dependencia temporal dada por la función unidimensional "delta", $\delta(t - \tau)$,

y una dependencia espacial gobernada por la función $\delta(\mathbf{x}-\xi)$, que es, lógicamente, tridimensional.

Para resolver esta ecuación, se deben imponer condiciones iniciales nulas $\mathbf{u}(\mathbf{x},0)=\dot{\mathbf{u}}(\mathbf{x},0)=0$ para $\mathbf{x}\neq 0$. Bajo estas condiciones, se deduce que la solución de esta ecuación tiene la forma (Grechka and Heigl, 2017):

$$u_i(\mathbf{x},t) = G_{ij} * f_j = \mathbf{G}(\mathbf{x},t;\xi,\tau)\cdot\left[*\mathbf{f}(\xi,\tau)\right]. \tag{A.3}$$

donde $\mathbf{G}(\mathbf{x},t;\xi,\tau)$ y G_{ij} son el tensor de Green y sus componentes (funciones de Green), respectivamente. Con el fin de que la suma de fuerza externas a nuestro sistema sean nulas, esto es:

$$\int_{vol}\mathbf{f}(\xi,\tau)d\xi = 0, \tag{A.4}$$

podemos proponer que $\mathbf{f}^{vol}$ sea, además, una fuerza de tipo dipolar o "cupla", compuesta por un par de fuerzas de igual magnitud y en direcciones opuestas, cada una de ellas actuando sobre los extremos de un segmento recto. Así, estas dos fuerzas deberían cumplir que:

$$\mathbf{f}(\xi,\tau) = \mathbf{f}(\xi-\Delta\xi,\tau), \tag{A.5}$$

donde $\Delta\xi$ debe tomarse lo suficientemente pequeño como para que la hipótesis de una fuerza de tipo puntual siga siendo válida. Si además se quiere conservar el momento angular, es necesario considerar estas dos fuerzas paralelas a $\Delta\xi$, o "cupla paralela". Si esto último no se cumpliese, se tendría un torque efectivo lo que violaría la conservación del momento angular. Para evitar esto, se considera una "doble cupla", es decir, dos cuplas de igual magnitud actuando en direcciones opuestas.

Considerando una cupla paralela, la ecuación A.3 se reescribe como:

$$\begin{aligned}\mathbf{u}(\mathbf{x},t) = {}& \mathbf{G}(\mathbf{x},t;\xi,\tau)\cdot\left[*\mathbf{f}(\xi,\tau)\right] + \\ & + \mathbf{G}(\mathbf{x},t;\xi-\Delta\xi,\tau)\cdot\left[*\mathbf{f}(\xi-\Delta\xi,\tau)\right].\end{aligned} \tag{A.6}$$

Expandiendo $\mathbf{G}(\mathbf{x},t;\xi-\Delta\xi,\tau)$ en series de Taylor alrededor de $\Delta\xi=0$ y conservando únicamente los términos de primer orden (despreciando los términos de orden superior), se obtiene:

$$G_{ik}(\mathbf{x},t;\xi-\Delta\xi,\tau)\Big|_{\Delta\xi=0} \approx G_{ik}(\mathbf{x},t;\xi,\tau) + \xi_j\frac{\partial G_{ik}}{\partial\xi_j}. \tag{A.7}$$

Reemplazando en A.6 y utilizando A.5:

$$\mathbf{u}(\mathbf{x},t) = \nabla_\xi\mathbf{G}(\mathbf{x},t;\xi,\tau) : \left[*\mathbf{f}(\xi,\tau)\Delta\xi\right], \tag{A.8}$$

donde : denota el doble producto punto, que actúa como contracción de los últimos índices, lo que permite escribir la ecuación A.8 como:

$$u_i(\mathbf{x},t) = \frac{\partial G_{ij}(\mathbf{x},t;\xi,\tau)}{\partial\xi_k} * f_j(\xi,\tau)\Delta\xi_k. \tag{A.9}$$

La cantidad $[\mathbf{f}(\xi,\tau)\Delta\xi]$ en la ecuación A.8 define el "tensor momento" y es una matriz de dimensión 3×3 que es función tanto de la posición ξ de la fuente, como del tiempo τ. Esta cantidad transporta toda la información pertinente a la fuente y solo es capaz de ser observada, como tal, a una distancia significativamente mayor a la longitud de onda que caracteriza la fractura (Aki and Richards, 2002). En su forma general, se escribe como:

$$\mathbf{M}(\xi,\tau)=\mathbf{f}(\xi,\tau)\Delta\xi=\begin{pmatrix} m_{11} & m_{12} & m_{13}\\ m_{12} & m_{22} & m_{23}\\ m_{13} & m_{23} & m_{33}\end{pmatrix}, \tag{A.10}$$

que es una matriz simétrica representando 9 cuplas, de las cuales solo 6 son independientes.

Los 6 elementos independientes de esta matriz $\mathbf{M}=\mathbf{M}(\xi,\tau)$ representan las incógnitas en un problema de MTI. Esto es, dado un desplazamiento $\mathbf{u}(\mathbf{x},t)$, registrado en un receptor, nos interesa conocer cuál es tensor momento $\mathbf{M}$ responsable de dicho desplazamiento. Es por este motivo que el problema de MTI puede plantearse como un problema de inversión. Aquí, si se tiene en cuenta que el desplazamiento observado es una combinación lineal de acciones de una cantidad determinada de cuplas, la inversión solo estaría limitada a conocer las funciones de Green.

Las funciones de Green, como puede deducirse de la ecuación A.3, son el nexo entre el desplazamiento del medio y la fuerza actuando sobre el mismo. Si se considera una fuente caracterizada por un tensor momento $\mathbf{M}$, toda la deformación que sufra la señal (emitida por dicha fuente) a lo largo de su trayectoria, estará determinada por estas funciones de Green.

Encontrar las expresiones analíticas de la funciones de Green es una tarea algebraica laboriosa (ver Aki and Richards (2002); Chapman (2004); Cerveny (2005)). Sus partes constitutivas dependen de las posiciones de la fuente y el receptor, y de la naturaleza del medio que atraviesa el rayo a lo largo de toda su trayectoria. Esto implica que a mayor complejidad del medio (anisotropía, inhomogeneidades, etc), mayor será la dificultad para encontrar dichas expresiones. En este sentido, es claro que el escenario más sencillo para desarrollar estas expresiones es un medio elástico, isótropo, homogéneo y semi-infinito. Si el lector está interesado en dichas deducciones, puede referirse a los textos recientemente citados. A continuación mostramos las expresiones resueltas y describimos el significado de sus partes.

Para comenzar, las funciones de Green para "campo lejano" correspondientes a un medio homogéneo e isótropo se escriben como:

$$\begin{aligned}\mathbf{G}(\mathbf{x},t;\xi,\tau)=&\frac{\delta\left(t-\tau-\mathbf{r}\cdot\mathbf{p}^{P}\right)}{4\pi\rho R\,V_P^2}\mathbf{U}^{P}\left[\mathbf{U}^{P}\right]^{T}+\\ &+\frac{\delta\left(t-\tau-\mathbf{r}\cdot\mathbf{p}^{S}\right)}{4\pi\rho R\,V_S^2}\mathbf{U}^{S}\left[\mathbf{U}^{S}\right]^{T},\end{aligned} \tag{A.11}$$

donde ρ es la densidad del medio y $\mathbf{r} = |\mathbf{x} - \xi|$ es el vector fuente-receptor, siendo $|\mathbf{r}| = R$ lo suficientemente grande como para satisfacer las condiciones de "campo lejano", esto es, una distancia mucho más grande que la longitud de onda de la fuente (Yu et al., 2015a). Además, $\mathbf{p}^{P,S} = \mathbf{n}/V_{P,S}$ son los vectores columna de lentitud para las velocidades P y S, respectivamente, siendo $\mathbf{n} = \mathbf{r}/R$ el vector normal al frente de ondas (esto último solo para el caso isotrópico). Los vectores columna $\mathbf{U}^{P,S}$ son los llamados "vectores de polarización" correspondientes a las fases P y S, respectivamente, y son de gran importancia en el desarrollo de estas funciones. Los mismos se obtienen de resolver la ecuación de Kelvin-Christoffel:

$$[\mathbf{\Gamma}(\mathbf{n}) - \rho\, V^2\, \mathbf{I}]\cdot\mathbf{U} = 0, \tag{A.12}$$

donde $\Gamma_{ij} = l_{iI} C_{IJ} l_{JI}$ es el tensor simétrico de Christoffel (Carcione, 2007), de dimensión 3×3 con $l_{1,2,3}$ los cosenos directores que determinan la normal al frente de ondas, $\mathbf{n}$. La matriz $\mathbf{C}$ es, por su parte, la matriz de elasticidad (ver Carcione (2007): su ecuación 1.32, Capítulo 1.), que define el medio y contiene las constantes elásticas del mismo. Si se normaliza la ecuación A.12 por la densidad del medio se obtiene:

$$\left[\frac{\mathbf{\Gamma}(\mathbf{n})}{\rho} - V^2\, \mathbf{I}\right]\cdot\mathbf{U} = 0, \tag{A.13}$$

que es equivalente a pensar en una matriz de elasticidad-normalizada $\tilde{\mathbf{C}} = \mathbf{C}/\rho$. La ecuación A.13 constituye un problema de autovalores y autovectores, donde los primeros son las velocidades de fase (al cuadrado) de la onda plana propagándose por el medio y los segundos son los ya mencionados vectores de polarización. Como es de esperar, la naturaleza de estas cantidades cambia con la complejidad del medio y su análisis lo hace en concordancia. La matriz $\tilde{\mathbf{C}}$ es una matriz definida positiva, por lo que los autovalores V^2 son cantidades estrictamente positivas, y así la ecuación A.13 define ondas de cuerpo planas (Grechka and Heigl, 2017). Se desprende de esta misma ecuación que las 3 posibles velocidades de fase dependen de la dirección normal al frente de ondas (excepto para el caso isotrópico), donde, ordenadas de mayor a menor

$$V_1(\mathbf{n}) \geq V_2(\mathbf{n}) \geq V_3(\mathbf{n}). \tag{A.14}$$

La mayor de estas cantidades se corresponde con la velocidad de fase de la onda P, $V_1 = V_P$, y $V_{2,3}$ son las velocidades de las ondas de cizalla rápida y lenta, respectivamente. Para el caso particular de un medio isótropo, se cumple que $V_2(\mathbf{n}) \equiv V_3(\mathbf{n})$ para cualquier dirección normal $\mathbf{n}$. Más aún, en el caso isotrópico, las velocidades de fase son coincidentes con las velocidades de grupo, y la dirección normal $\mathbf{n}$ es paralela a la dirección del rayo $\mathbf{r}$, lo que en general, no se cumple para medios anisotrópicos*. Por otra parte, debido a ser $\tilde{\mathbf{C}}$ definida

*Todas estas propiedades son analizadas en profundidad en diferentes textos como, por ejemplo, Grechka and Heigl (2017) o Carcione (2007), siendo el primero algo más condensado que el segundo. En este texto, y por simplicidad, elegimos seguir el análisis de Grechka and Heigl (2017).

positiva, los vectores de polarización $\mathbf{U}^{P,S_1,S_2}$ son siempre perpendiculares entre si, lo que no implica que sean paralelos o no a la normal $\mathbf{n}$, excepto para el caso isotrópico, donde si se cumple que $\mathbf{U}^P \cdot \mathbf{n} = 1$ y $\mathbf{U}^{S_1,S_2} \cdot \mathbf{n} = 0$.

Volviendo a la ecuación A.11, se puede decir que la dirección del movimiento de las partículas del medio, a medida que la onda se propaga por el mismo, está definido por los vectores de polarización $\mathbf{U}^P$, paralelo a $\mathbf{n}(= \mathbf{r})$ y $\mathbf{U}^S$ perpendicular a $\mathbf{n}$. La amplitud de dicho movimiento está determinada por el factor:

$$A^{P,S} = \frac{1}{4\pi\, R\, V_{P,S}^2}, \tag{A.15}$$

donde se evidencia que la amplitud decae con $\propto R^{-1}$. Si se incluye esta cantidad, se puede reescribir la ecuación A.3 para una fuente puntal como:

$$\begin{aligned}
\mathbf{u}(\mathbf{x},t) &= \mathbf{G}(\mathbf{x},t;\xi,\tau)\cdot[*\mathbf{f}(\xi,\tau)] = \\
&= A^P \delta\left(t-\tau-\mathbf{r}\cdot\mathbf{p}^P\right)\mathbf{U}^P\left[\mathbf{U}^P\right]^T\cdot[*\mathbf{f}(\xi,\tau)] + \\
&\quad + A^S \delta\left(t-\tau-\mathbf{r}\cdot\mathbf{p}^S\right)\left[\mathbf{U}^S\right]^T\mathbf{U}^S\cdot[*\mathbf{f}(\xi,\tau)] = \\
&= A^P \delta\left(t-\tau-\mathbf{r}\cdot\mathbf{p}^P\right)\mathbf{U}^P R^P(\xi,\tau) + \\
&\quad + A^S \delta\left(t-\tau-\mathbf{r}\cdot\mathbf{p}^S\right)\mathbf{U}^S R^S(\xi,\tau),
\end{aligned} \tag{A.16}$$

donde la cantidad escalar:

$$R^{P,S}(\xi,\tau) = \left[\mathbf{U}^{P,S}\right]^T\cdot[*\mathbf{f}(\xi,\tau)], \tag{A.17}$$

se define como el patrón de radiación de la señal y determina la relación entre su amplitud y la dirección de propagación de la misma.

Si continuamos considerando un medio homogéneo e isótropo, pero incluimos el modelo de fuente dipolar descripto por las ecuaciones A.8 y A.9 obtenemos:

$$\begin{aligned}
\mathbf{u}(\mathbf{x},t) &= \nabla_\xi \mathbf{G}(\mathbf{x},t;\xi,\tau) : [*\mathbf{M}(\xi,\tau)] = \\
&= \nabla_\xi\left\{A^P \delta\left(t-\tau-\mathbf{r}\cdot\mathbf{p}^P\right)\mathbf{U}^P\left[\mathbf{U}^P\right]^T\right\} : [*\mathbf{M}(\xi,\tau)] + \\
&\quad + \nabla_\xi\left\{A^S \delta\left(t-\tau-\mathbf{r}\cdot\mathbf{p}^S\right)\mathbf{U}^S\left[\mathbf{U}^S\right]^T\right\} : [*\mathbf{M}(\xi,\tau)] = \\
&= \left\{A^P \nabla_\xi\delta\left(t-\tau-\mathbf{r}\cdot\mathbf{p}^P\right)\mathbf{U}^P\left[\mathbf{U}^P\right]^T\right\} : [*\mathbf{M}(\xi,\tau)] + \\
&\quad + \left\{A^S \nabla_\xi\delta\left(t-\tau-\mathbf{r}\cdot\mathbf{p}^S\right)\mathbf{U}^S\left[\mathbf{U}^S\right]^T\right\} : [*\mathbf{M}(\xi,\tau)].
\end{aligned} \tag{A.18}$$

Si ahora se considera que:

$$\mathbf{r}\cdot\mathbf{p}^{P,S} = \frac{1}{R}\left[\frac{(x_1-\xi_1)^2}{V_{P,S}} + \frac{(x_2-\xi_2)^2}{V_{P,S}} + \frac{(x_3-\xi_3)^2}{V_{P,S}}\right], \tag{A.19}$$

se puede deducir que:

$$\begin{aligned}\nabla_\xi \delta\left(t-\tau-\mathbf{r}\cdot\mathbf{p}^{P,S}\right) = \\ = \delta'\left(t-\tau-\mathbf{r}\cdot\mathbf{p}^{P,S}\right)\nabla_\xi(t-\tau-\mathbf{r}\cdot\mathbf{p}^{P,S}) = \\ = \delta'\left(t-\tau-\mathbf{r}\cdot\mathbf{p}^{P,S}\right)\cdot\mathbf{p}^{P,S}.\end{aligned} \tag{A.20}$$

que permite reescribir la ecuación A.18 como:

$$\begin{aligned}\mathbf{u}(\mathbf{x},t) = \nabla_\xi \mathbf{G}(\mathbf{x},t;\xi,\tau)\colon[*\mathbf{M}(\xi,\tau)] = \\ = A^P\delta'\left(t-\tau-\mathbf{r}\cdot\mathbf{p}^P\right)\mathbf{U}^P\left[\mathbf{U}^P\right]^T\cdot\left[*\mathbf{M}(\xi,\tau)\cdot\mathbf{p}^P\right] + \\ + A^S\delta'\left(t-\tau-\mathbf{r}\cdot\mathbf{p}^S\right)\mathbf{U}^S\left[\mathbf{U}^S\right]^T\cdot\left[*\mathbf{M}(\xi,\tau)\cdot\mathbf{p}^S\right] = \\ = A^P\delta'\left(t-\tau-\mathbf{r}\cdot\mathbf{p}^P\right)\mathbf{U}^P R^P + \\ + A^S\delta'\left(t-\tau-\mathbf{r}\cdot\mathbf{p}^S\right)\mathbf{U}^S R^S,\end{aligned} \tag{A.21}$$

donde:

$$R^{P,S} = \left[\mathbf{U}^{P,S}\right]^T\cdot\left[*\mathbf{M}(\xi,\tau)\cdot\mathbf{p}^{P,S}\right], \tag{A.22}$$

es el patrón de radiación correspondiente a una fuente de tipo doble-dipolo (Grechka and Heigl, 2017).

Las ecuaciones A.16 y A.21, correspondientes a fuerzas de tipo puntual y doble dipolo, respectivamente, permiten calcular explícitamente el desplazamiento recibido por un receptor localizado en un punto $\mathbf{x}$ en un medio homogéneo e isótropo.

La deducción de las expresiones equivalentes para medios anisotrópicos (y homogéneos de una sola capa) es más laboriosa. Por este motivo, nos limitaremos a mostrar sus expresiones finales y describiremos sus partes en forma simplificada. Así, para el caso de una fuente puntual, se tiene que:

$$\begin{aligned}\mathbf{u}(\mathbf{x},t) = \\ = A^P\hat{\delta}\left(t-\tau-\mathbf{r}\cdot\mathbf{p}^P\right)\mathbf{U}^P R^P(\xi,\tau) + \\ + A^{S_1}\hat{\delta}\left(t-\tau-\mathbf{r}\cdot\mathbf{p}^{S_1}\right)\mathbf{U}^S R^{S_1}(\xi,\tau) + \\ + A^{S_2}\hat{\delta}\left(t-\tau-\mathbf{r}\cdot\mathbf{p}^{S_2}\right)\mathbf{U}^S R^{S_2}(\xi,\tau),\end{aligned} \tag{A.23}$$

y para el caso de una fuente de tipo doble-dipolo:

$$\begin{aligned}\mathbf{u}(\mathbf{x},t) = \\ = A^P\hat{\delta}'\left(t-\tau-\mathbf{r}\cdot\mathbf{p}^P\right)\mathbf{U}^P R^P(\xi,\tau) + \\ + A^{S_1}\hat{\delta}'\left(t-\tau-\mathbf{r}\cdot\mathbf{p}^{S_1}\right)\mathbf{U}^S R^{S_1}(\xi,\tau) + \\ + A^{S_2}\hat{\delta}'\left(t-\tau-\mathbf{r}\cdot\mathbf{p}^{S_2}\right)\mathbf{U}^S R^{S_2}(\xi,\tau).\end{aligned} \tag{A.24}$$

donde las expresiones para los patrones de radiación R^P, R^{S_1} y R^{S_2} son equivalentes a las expresiones mostradas en las ecuaciones A.17 y A.22.

A pesar de las semejanzas, existen varias diferencias entre estas últimas ecuaciones y las correspondientes a un medio isótropo. En primer lugar, se puede ver que ya no existen únicamente dos fases, sino que hay al menos tres, apareciendo los desplazamientos S_1 y S_2 (S rápida y S lenta, respectivamente). En el caso más general, el número máximo de fases presentes en un desplazamiento puede llegar a ser de 19 (Grechka and Heigl, 2017). Por otro lado, es importante tener en cuenta que para el caso anisotrópico, los vectores del rayo, $\mathbf{r}$, y la lentitud, $\mathbf{p}$, ya no son, en general, paralelos. Esto implica que las velocidades de grupo,$\mathbf{V}^g$, tampoco son, en general, paralelas a las velocidades de fase $\mathbf{V}$. En cuanto a las cantidades A^{P,S_1,S_2}, las mismas se expresan como:

$$A^{P,S_1,S_2} = \frac{1}{4\pi R|\mathbf{V}^g_{P,S_1,S_2}|\sqrt{|K_{P,S_1,S_2}|}} \tag{A.25}$$

donde K_{P,S_1,S_2} son las curvaturas gaussianas correspondientes a cada fase (Cerveny, 2005). Finalmente, la función $\hat{\delta}$ se define como:

$$\hat{\delta}(t-\tau-\mathbf{r}\cdot\mathbf{p}) = Re\left[\exp\left(\frac{-i\pi\zeta_0}{2}\right)\delta^A(t-\tau-\mathbf{r}\cdot\mathbf{p})\right], \tag{A.26}$$

donde δ^A es la delta analítica (Grechka, 2015a) y ζ_0 es el índice KMAH evaluado en la posición de la fuente (Cerveny, 2005).

Por completitud, diremos que la ecuación A.24 deriva, para el caso de una fuente de tipo doble-dipolo en un medio anisotrópico de capas planas, en la expresión:

$$\begin{aligned}\mathbf{u}(\mathbf{x},t) &= \nabla_\xi \mathbf{G}(\mathbf{x},t;\xi,\tau)\colon[*\mathbf{M}(\xi,\tau)] = \\ &= \sum_j \frac{Re\left[\mathcal{R}^j e^{[-i\pi(\zeta_0-\zeta)/2]}\,\delta^A(t-\tau-\mathbf{r}^j\cdot\mathbf{p}^j)\right]}{4\pi\sqrt{\rho(\mathbf{x})V_j(\mathbf{x})\rho(\xi)V_j(\xi)}\,\mathcal{L}^j}\times \\ &\quad\times \mathbf{U}^j(\xi)\left[*\,\mathbf{U}^j(\xi)\cdot\mathbf{M}(\xi,\tau)\cdot\mathbf{p}^j(\xi)\right], \quad j = P,\, S_1,\, S_2,\end{aligned} \tag{A.27}$$

donde

$$\mathcal{R}^j = \prod_{l=1}^{L-1}\mathcal{R}^j_l\sqrt{\frac{\rho_{l+1}\,|\mathbf{g}^j_{l+1}\cdot\mathbf{i}_l|}{\rho_l\,|\mathbf{g}^j_l\cdot\mathbf{i}_l|}}, \quad L\geq 2, \tag{A.28}$$

es el coeficiente de reflexión/transmisión para el rayo y $\mathcal{R}^j_1 = 1$, con L el número de interfaces que atraviesa el rayo en su trayectoria. Además, $\mathbf{g}^j_l$ son las velocidades de grupo para cada fase correspondientes a cada interfaz l e $\mathbf{i}_l$ es la normal a cada interfaz (en capas planas paralelas será un vector unitario vertical). Por último, $\mathcal{L}^j$ es el decaimiento geométrico calculado para cada trayectoria del rayo correspondiente a cada fase (Aki and Richards, 2002; Grechka and Heigl, 2017; Cerveny, 2005).

Finalmente, las ecuaciones A.21, A.24 y A.27 nos permiten calcular los desplazamientos para una fuente de tipo doble-dipolo para un medio homogéneo e isótropo de una sola capa, anisotrópico de una sola capa, y anisotrópico de varias capas (debe reducirse al de una sola capa si se considera $L = 1$), respectivamente, y serán las que nos permitan generar nuestros datos sintéticos en el futuro.

Las expresiones mencionadas anteriormente constituyen la roca fundamental sobre las cuales se basan los métodos tradicionales de MTI, ya que, transformadas al dominio de la frecuencia, ω, establecen una relación lineal entre los desplazamientos $\mathbf{u}(\mathbf{x}, \omega)$ y el tensor momento $\mathbf{M}(\xi, \omega)$. Así, para el caso anisotrópico,

$$\begin{aligned} \mathbf{u}(\mathbf{x}, \omega) &= \mathcal{F}\left[\frac{\partial G_{ij}(\mathbf{x}, t; \xi, \tau)}{\partial \xi_k} * M_{jk}(\xi, \tau)\right] = \\ &= \frac{\partial G_{ij}(\mathbf{x}, \xi, \omega)}{\partial \xi_k} M_{jk}(\xi, \omega) \end{aligned} \tag{A.29}$$

que puede reescribirse en forma matricial como:

$$\mathbf{d} = \hat{\mathbf{G}}\,\mathbf{m}, \tag{A.30}$$

que es la misma expresión que encontramos en 4.1. Aquí, $\mathbf{m} = [m_{11}, m_{22}, m_{33}, m_{23}, m_{13}, m_{12}]^T$ es un vector columna con las 6 componentes independientes del tensor momento $\mathbf{M}$. La matriz $\hat{\mathbf{G}}$, por su parte, se construye como:

$$\begin{aligned} G_{ij} &= G_{ij,j} \quad j = 1, 2, 3 \\ G_{i4} &= G_{i2,3} + G_{i3,2} \\ G_{i5} &= G_{i1,3} + G_{i3,1} \\ G_{i6} &= G_{i1,2} + G_{i2,1}, \end{aligned} \tag{A.31}$$

donde G_{ij} son las componentes de "campo lejano" de las funciones de Green correspondientes al medio sobre el cual se está trabajando. Por último, $\mathbf{d}$ es la señal que arriba a cada uno de los receptores (ver: Vavryčuk and Kühn (2012)).

Apéndice B

Redes neuronales

Introducción

Las redes neuronales artificiales (ANN, por sus siglas en inglés) son una categoría particular dentro de las estrategias del aprendizaje automático y su funcionamiento está basado en los complejos sistemas biológicos que podrían encontrarse en cerebros humanos (u otros organismos vivos). Un cerebro animal está compuesto típicamente por una infinidad de neuronas interconectadas unas con otras por medio de extremidades complejas (axones). Estas conexiones permiten la comunicación entre neuronas mediante impulsos eléctricos (sinapsis). Lo cierto es que, aun cuando hay mucho por descubrir sobre la composición, estructura y funcionamiento de un cerebro animal, se sabe que las neuronas están organizadas en forma de *clusters* o capas apiladas consecutivamente. Teniendo esta imagen en mente, resulta que una ANN es una recreación artificial y simplificada de esta estructura. Las mismas suelen estar compuestas por neuronas artificiales, que funcionan como sus unidades elementales que, estando conectadas por alguna relación matemática, se comunican recibiendo entradas y entregando salidas. En cualquier caso, para el universo de las ANN estas entradas y salidas no son más que paquetes de datos que, dependiendo del problema, pueden ser desde muy simples a más complejos. Así, dos neuronas artificiales interconectadas pueden entenderse de manera que la primera recibe un dato como entrada, lo procesa y lo envía como entrada de la segunda, que a su vez, realiza otra serie de operaciones produciendo su propia salida. Dependiendo de la forma en la que estas unidades se agrupan y relacionan entre sí, se habla de diferentes "arquitecturas" de ANN. Por ello, la arquitectura de una ANN nos brinda información del aspecto físico y mecánico de la red. En otras palabras, de su forma, tamaño, tipo y funcionamiento.

El perceptrón

Una de las arquitecturas más sencillas para una red neuronal es el "perceptrón", introducido por primera vez por Rosenblatt (1958). La unidad elemental de un perceptrón es conocida como "unidad lógica de umbral", comúnmente referida como *Linear Threshold Unit* (LTU o TLU, por sus siglas en inglés). Esta estructura matemática fue diseñada por McCulloch and Pitts (1943) con el objetivo de simular ciertas actividades de un sistema nervioso biológico. Matemáticamente, estas unidades quedan definidas por dos operaciones fundamentales. En primer lugar, cuando esta neurona es estimulada (recibe un dato de entrada), se realiza una combinación lineal o suma pesada de los elementos que componen la entrada $\mathbf{x}$ para producir una salida z. Esta operación puede escribirse de la siguiente manera:

$$z = w_1 x_1 + \cdots + w_n x_n = \sum_{i=1}^{n} w_i \, x_i = \mathbf{x^T} \, \mathbf{w}, \tag{B.1}$$

donde x_i, w_i, son los elementos de la entrada $\mathbf{x}$ y pesos $\mathbf{w}$ del perceptrón, respectivamente. Esta instancia transforma el dato de entrada en una cantidad más uniforme, cuyo rango puede variar entre $-\infty$ e ∞, dependiendo de los valores numéricos de sus partes y los pesos. En segundo lugar, este nuevo dato "intermedio" z es utilizado como entrada de una función "salto" (de tipo *step*). La función más común suele ser la *Heaviside step function*, definida como:

$$H(z) = \begin{cases} 0, & \text{si } z \leq 0 \\ 1, & \text{si } z > 0, \end{cases} \tag{B.2}$$

aunque existen otras, como la función signo. En este contexto, estas funciones son denominadas "funciones de activación". Cuando se aplica una de estas funciones a z, se dice que la neurona se "activa" si se supera el umbral (en este caso $z = 0$). La idea detrás de esta expresión se hereda del comportamiento de las neuronas biológicas, ya que se sabe que las mismas son capaces de recibir mensajes de sus pares en forma de impulsos eléctricos, pero solo son capaces de retransmitirlo cuando el mismo supera una cierta cantidad umbral (de ahí, *threshold*) que es particular para cada neurona. Cuando esto sucede, se dice que la neurona fue "activada" por dicho impulso. De igual manera, estas funciones de activación son los umbrales de decisión para las LTUs: son las "encargadas" de decidir si la cantidad z es suficiente o no para activar la unidad, y así, definir qué tipo de dato (o información) transmitirá o dejará de transmitir. En el caso particular de la $H(z)$ definida por la ecuación B.2, este umbral es $z = 0$, y la salida O_{ltu} de esta unidad será el resultado de componer estas operaciones, es decir

$$O_{ltu} = H(\mathbf{x^T w}). \tag{B.3}$$

Aun cuando esta definición de perceptrón nos permitiría construir funciones lógicas elementales para (por ejemplo) clasificar una entrada en diferentes clases, su aplicación

es bastante limitada. Para que un perceptrón sea completamente funcional, es necesario agregar un término de *bias*. Puede pensarse a este *bias* como una neurona con un único peso w_b que se suma al resultado de la combinación lineal definida en B.1. Considerando este peso "extra", la ecuación B.1 se reemplazaría como:

$$z' = z + w_b = \sum_{i=1}^{n+1} w_i x_i = \mathbf{x^T w}, \tag{B.4}$$

donde ahora la entrada será $\mathbf{x} = [x_1, x_2, \ldots, x_n, 1]$ y los pesos del perceptrón serán $\mathbf{w} = [w_1, w_2, \ldots, w_n, w_{n+1}]$, con $w_{n+1} = w_b$. La función de esta neurona de *bias* es tan sencilla como importante: permite trasladar la salida de mi perceptrón (luego de aplicar la función de activación) según sea necesario. Para el caso de una función $H(z)$, por ejemplo, el peso de la neurona de *bias* es el que definirá el umbral de activación (se traslada la salida):

$$H(z') = H(z + w_b) = \begin{cases} 0, & \text{si } z \leq -w_b \\ 1, & \text{si } z > -w_b. \end{cases} \tag{B.5}$$

La Figura B.1 muestra una LTU típica con una entrada $\mathbf{x}_j \in \mathbb{R}^{n+1}$ y una función de activación f.

Es importante mencionar que, pese a que la LTU fue originalmente diseñada con estas funciones de activación de tipo "salto", en la actualidad las mismas no se limitan a este tipo de funciones. Así, hoy en día existen numerosas funciones de activación en la bibliografía, y mientras que la mayoría de sus cualidades y utilidades pueden variar, tienen en común el ser transformaciones no-lineales y diferenciables. Esta última propiedad, como veremos más adelante, es de vital importancia para el funcionamiento de una ANN. Además, como se puede ver en las ecuaciones B.2 y B.5, las salidas de una función salto pueden ser únicamente 1 o 0, mientras que las salidas de otras funciones de activación pueden ser un número real que varíe en un rango continuo (cerrado o no) de valores.

Habiendo definido las partes fundamentales y el funcionamiento de una unidad LTU, cabe mencionar que un perceptrón es el agrupamiento de uno o varias LTUs (formando una única capa, o *layer*).

La Figura B.2 muestra un perceptrón formado por 4 LTUs alimentado por una entrada $\mathbf{x}_j \in \mathbb{R}^{n^*}$. En forma estricta, se conoce como *layer* al agrupamiento de varias** LTUs dispuestos de forma tal que todos los datos de entrada a esta capa estén conectados a cada una de las LTUs que la componen. En el marco de una *layer*, suele también referirse a cada unidad que la compone como un "nodo".

---

*Es importante notar que aquí, la dimensión de $\mathbf{x}_j$ es n ya que no se considera el elemento de *bias*.

**Notar que esto no implica que no pueda existir un perceptrón (o layer) formado por una única LTU, aunque este no es el caso general.

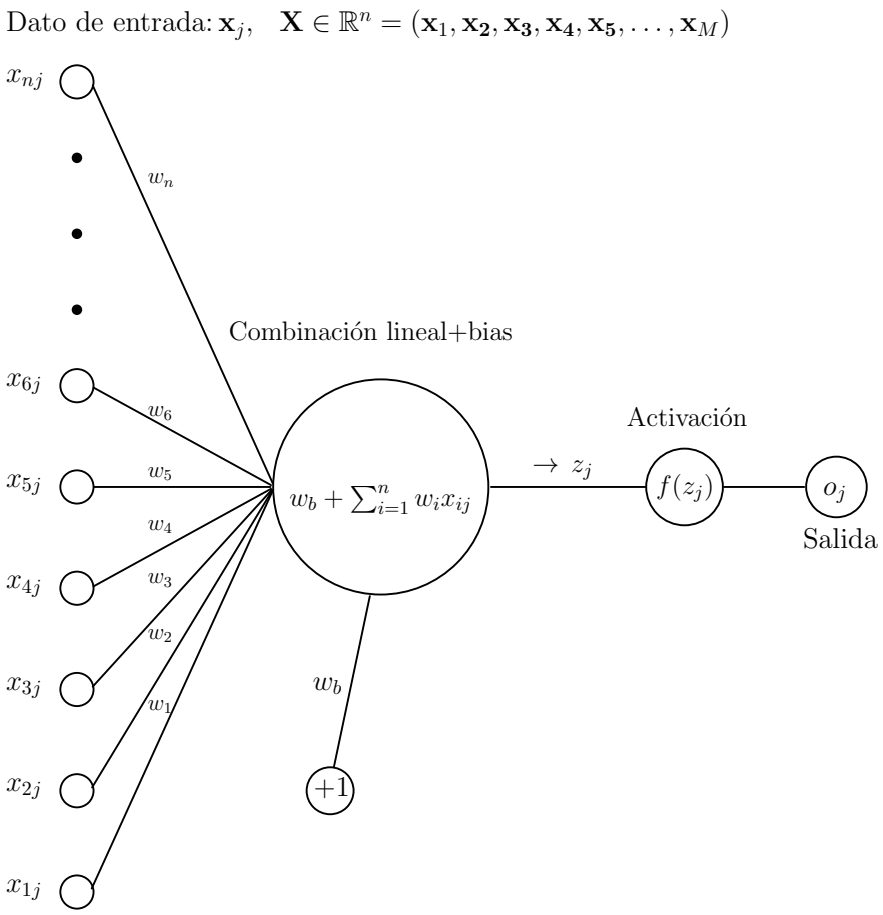


Figura B.1: Una LTU típica. La entrada es un elemento $\mathbf{x}_j$ cualquiera de un conjunto de instancias $\mathbf{X} \in \mathbb{R}^n$. Las componentes de x_i de cada instancia $\mathbf{x}_j$ se multiplican por pesos w_i y luego se suma un término de bias, w_b, para producir una cantidad escalar z_j. Esta salida luego es evaluada en una función de activación no-lineal $f(\,)$, obteniendo una salida final escalar o_j.

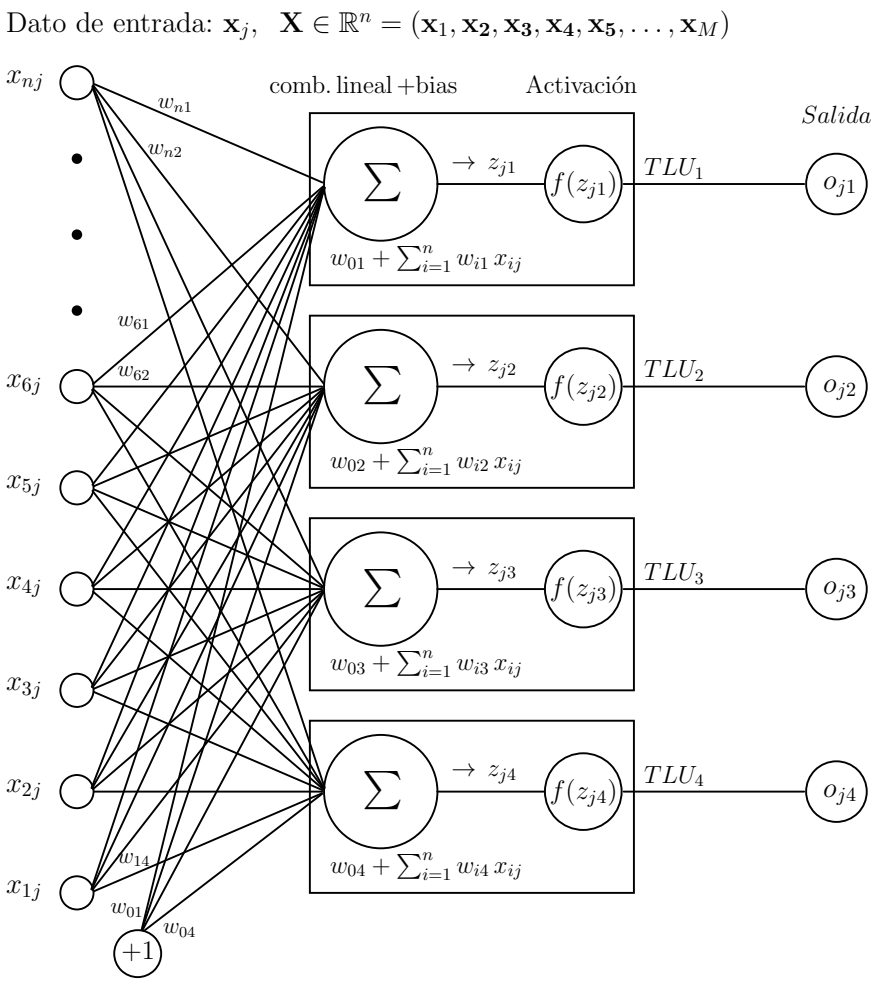


Figura B.2: Representación típica de una *layer* de 4 nodos, o equivalentemente, un perceptrón formado por 4 LTUs. La entrada es un elemento $\mathbf{x}_j$ cualquiera de un conjunto de instancias $\mathbf{X} \in \mathbb{R}^n$. Las n componentes de $\mathbf{x}_j$ se distribuyen sobre todas las unidades de la *layer* y se multiplican con sus respectivos pesos w_{ip}, $p = 1, \ldots, 4$. Luego, un término adicional de *bias*, w_{0p}, es sumado para producir una cantidad z_{jp}. Es importante notar que cada una de las LTU_p produce una única salida de la dimensión que corresponda.

Tipos de capas de una red neuronal

En este punto, tenemos en claro el concepto básico de un nodo, llamado también LTU, y de cómo varias LTUs pueden agruparse como una capa para formar un perceptrón. Ahora podemos entrar en detalle en el concepto de *layer* o "capa".

En principio, una ANN puede estar constituida por una innumerable cantidad de capas, que a su vez, pueden estar compuestas por una innumerable cantidad de nodos. Sin embargo, la expresión más simple de una ANN está compuesta por una "capa de entrada" y una "capa de salida". Este tipo de ANN se denomina "red de capa simple".

La capa de entrada, o *input layer* (IL), como su nombre lo indica, es la primer capa con la que interactúa el dato al atravesar una ANN. Estrictamente hablando, no es un perceptrón tal como fue definido anteriormente. En esta instancia, es importante aclarar que, a pesar de que la definición de la IL puede variar según la bibliografía, su función siempre es la misma. Así, esta capa puede presentarse como una de dos formas: o bien es una capa de neuronas simples que reciben el dato de entrada y lo distribuyen sin cambios sobre el "cuerpo" de la red, o sencillamente se presenta como el dato de entrada en sí mismo. En ambos casos, su salida es el dato inalterado. En lo que sigue, adoptaremos la primera convención.

La capa de salida, o *output layer* (OL), es la última capa que compone a la red, y su salida es aquella magnitud que se quiere predecir o inferir. Esto último implica que la cantidad de unidades que contenga esta capa estará definida por la dimensión de la salida deseada.

En general, y debido a su simplicidad, las aplicaciones de este tipo de ANNs son bastante limitadas. Por ello, en la práctica, cuando se habla de una ANN, se asume que la misma tiene una arquitectura de tipo "capa múltiple".

A diferencia de la anterior, estas últimas contienen un número finito de capas intermedias entre la IL y OL, a las que se denomina como capas ocultas, o *hidden layers* (HLs). La cantidad de HLs que tenga una ANN define su *profundidad*, mientras que la cantidad de unidades que componen cada capa define su "ancho". En general, una ANN que contenga más de 5 o 6 HLs se clasifica como profunda (*deep neural network*, DNN). Debido a que la IL no realiza ningún trabajo sobre el dato de entrada, las HLs constituyen el cuerpo principal de una ANN, condensando la gran mayoría de sus nodos y ejerciendo la mayor cantidad de trabajo/procesamiento sobre el dato.

En el futuro, nos referiremos a una ANN como la composición de una IL seguida de un apilamiento de HLs y finalizando con una OL. Así:

$$ANN = IL + \text{una o más } HL + OL. \tag{B.6}$$

Cuando todas las neuronas de una capa están conectadas a todas las neuronas de la capa siguiente, se habla de "capas completamente conectadas", o simplemente, "capas densas"

(*fully connected* o *dense layers*). Por otro lado, si la información o dato de entrada de la ANN fluye en una sola dirección, esto es, desde la capa de entrada hacia la capa de salida, se habla de una ANN "alimentada hacia adelante" (*feed-forward*)*. En esta Tesis, se asume que nuestras ANN estarán constituidas por capas densas y alimentadas hacia adelante.

Entrenamiento de una red neuronal

Una red neuronal puede pensarse como una serie de operaciones matriciales encadenadas una detrás de otra que modifican un dato de entrada y lo transforman en otro de salida. Estas operaciones, que combinan transformaciones lineales y no-lineales, pueden cambiar tanto la naturaleza del dato (por ejemplo, realizando un cambio de dominio) como su forma (por ejemplo, reduciendo o ampliando su dimensión). Así definida, una ANN no es más que un conjunto de operaciones estáticas. Sin embargo, la verdadera fortaleza de una ANN reside en su capacidad de ser "entrenada" para resolver un problema determinado. Esto es, su capacidad de recibir un dato y aprender a distinguir sus patrones distintivos. Así, se busca que una red neuronal sea capaz de aprender o "generalizar" (Neyshabur et al., 2017; Jacot et al., 2018; Zhang et al., 2021; Jakubovitz et al., 2019) a partir de un dado conjunto de datos, de forma tal que nos permita hacer predicciones sobre otros datos de la misma clase. En la práctica, el entrenamiento de una red implica ajustar los valores de los pesos que la componen de manera tal que dada una entrada, la salida producida sea una buena estimación (de acuerdo a algún criterio definido) al valor deseado.

Características del dato de entrenamiento

Para entrenar una red neuronal se requiere de un conjunto (o *set*) de datos que componen el llamado "dato de entrenamiento". En la bibliografía, los elementos individuales de este conjunto son llamados "instancias del dato", o simplemente "instancias". Las características que se buscan en este conjunto es que sea lo más grande y representativo (del problema) posible. La primer característica se explica por sí misma. En cuanto a la representatividad, es más sencillo entenderlo mediante un ejemplo.

Supongamos que queremos enseñarle a nuestra ANN a clasificar las diferentes especies de árboles que aparecen en una determinada imagen. Para ello, no solo necesitamos enseñarle los patrones distintivos que hacen a la estructura de un árbol en general, sino también aquellos que hacen a cada una de las especies de árboles que existen en la naturaleza. Así, es imperioso que nuestra ANN tenga "contacto" con muestras de cada una de estas especies, y preferentemente, en cantidad. Tampoco serviría enseñarle con un conjunto de

*A las redes neuronales cuyas neuronas no están todas conectadas, o donde la información no fluye únicamente hacia adelante se las conoce como *non-fully connected* y *non-feed-forward*, respectivamente.

imágenes donde solo aparece una única especie de árbol. Si así fuere, la ANN aprendería las características o reglas fundamentales que hacen a la estructura de esa especie particular, pero no sería capaz de distinguir entre dos especies diferentes. Con este último conjunto (con imágenes de una sola especie) solo se lograría entrenar la red para clasificar imágenes en categorías "árbol" y "no-árbol", pero no aprendería a distinguir entre las categorías "fresno", "arce", o "álamo", por mencionar algunas. El conjunto que se busca es aquel que contenga imágenes de todas las especies existentes (de ser posible), en diferentes estaciones (con o sin hojas), y con diferente luz (de mañana, tarde y noche), por nombrar algunas características.

Entre otras cualidades buscadas en un dato de entrenamiento, se destacan también la calidad del mismo y el tipo de información que brinda el conjunto. En relación a la calidad del dato, es evidente que a mayor presencia de ruido o errores, la ANN tendrá más dificultades para aprender un patrón. Es evidente que la presencia de una foto de un "perro" en el conjunto de fotos de árboles sería contraproducente y no sumaría más que ruido. En cuanto a la información que transporte o contenga este conjunto, y continuando con el mismo ejemplo, resulta evidente que si se pretende entrenar la red para clasificar árboles, es irrelevante asignarle a cada fotografía un casillero que indique su pertenencia al planeta "Tierra". Tampoco sería relevante información que indique la pertenencia de dicho árbol al "reino vegetal". En este sentido, siempre es preferible que el dato contenga información útil para realizar la tarea específica para la cual se está entrenando la red.

Tipos de entrenamiento: supervisado y no-supervisado

Existen muchas formas diferentes a partir de las cuales una ANN puede aprender los patrones o reglas de un problema. Las dos formas más conocidas son mediante un aprendizaje "supervisado" (Bzdok et al., 2018; Maimon and Rokach, 2005; Kotsiantis et al., 2007) o uno "no-supervisado" (Gentleman and Carey, 2008; Lloyd et al., 2013; Ghahramani, 2003).

En el aprendizaje supervisado, cada instancia d_i del conjunto de entrenamiento D cuenta con dos elementos: una entrada x y la salida verdadera y. Así, puede escribirse $d_i = (x_i, y_i)$. Siguiendo con el ejemplo anterior, cada una de las instancias debería estar compuesta por un archivo x con la "foto" (puede además tener más información) de un árbol y una "etiqueta" y que especifique qué tipo de especie aparece en dicha foto. Así, al entrenar, la ANN simplemente "vería" las fotos y asociaría ese árbol a la especie etiquetada. Es natural que al cabo de "ver" muchas fotos etiquetadas como "fresno", comenzará a identificar las características distintivas que constituyen a esa especie. Por ejemplo, la forma, el color de sus hojas, y el color de su corteza, entre otras.

En un aprendizaje de tipo no-supervisado, en cambio, el dato no contiene estas etiquetas. En este caso, la ANN deberá ser capaz de aprender las diferencias y similitudes que existen entre cada instancia del dato y clasificarlos según las características que observa

de cada imagen. En el ejemplo de los árboles, al cabo de un tiempo aprenderá a clasificar árboles según su color, hoja, corteza y forma (entre otros), pero será incapaz de distinguir la especie. Solo podrá agruparlos por similitudes, o separarlos por sus diferencias.

Algunas de los algoritmos más comunes de aprendizaje supervisado son: *k-nearest neighbors* (Keller et al., 1985; Bezdek et al., 1986), *decision trees* (Quinlan, 1986; Rokach and Maimon, 2005; Kingsford and Salzberg, 2008), *random Forest* (Breiman, 2001; Athey et al., 2019) y regresión (logística, lineal, no-lineal, etc), al cual le prestaremos particular atención debido a su importancia en este trabajo. Para el aprendizaje no-supervisado se tienen algoritmos tales como: *k-means* (Likas et al., 2003; Hamerly and Elkan, 2004), *principal component analysis* (Wold et al., 1987; Howley et al., 2005), *hierarchical cluster analysis* (Kassambara, 2017; Köhn and Hubert, 2014), entre otras. Como es de suponer, cada uno de estos algoritmos se destaca en un cierto tipo de problema, y por ello, al momento de elegir qué tipo de aprendizaje se debe adoptar, es imperioso saber cuál abordar y la solución que se quiere obtener. Cabe mencionar que también existen las estrategias de aprendizaje "semi-supervisadas", donde se tienen instancias con y sin etiquetas y de "aprendizaje reforzado", donde la red es motivada a aprender según un esquema de retribuciones y penalidades.

Regresión lineal y no-lineal

Nos enfocaremos aquí en explicar el funcionamiento de un entrenamiento de tipo supervisado. Esto significa que cada instancia de nuestro dato estará compuesta por una entrada y su respectiva salida deseada (etiqueta). En particular, estaremos interesados en algoritmos regresivos y por esta razón, es importante detenernos y comprender los fundamentos de un problema de regresión clásico.

Un problema de regresión consiste en hallar los parámetros β de un modelo f que mejor estimen la relación entre un conjunto de variables dependientes $\mathbf{Y}$, con otro conjunto de variables independientes $\mathbf{X}$. Esto es:

$$\hat{\mathbf{Y}} = f(\mathbf{X}, \beta), \tag{B.7}$$

donde $\hat{\mathbf{Y}} \sim \mathbf{Y}$. Para poder encontrar el "mejor" modelo se debe transformar el problema en un problema de optimización*. Típicamente, el método más utilizado para resolver dicha tarea es mediante mínimos cuadrados, donde se busca minimizar una función de costo $\mathcal{E}(\beta)$, generalmente expresada como el promedio de los errores al cuadrado. O sea,

$$\beta = \arg\min_{\beta} \mathcal{E}(\beta) = \arg\min_{\beta} \frac{1}{M} \sum_{j} \mathcal{L}(y_j, x_j), \tag{B.8}$$

*Veremos que el entrenamiento de redes neuronales está íntimamente relacionado a un problema de optimización clásico.

donde

$$\mathcal{E}(\beta) = \frac{1}{M} \sum_j \Big[y_j - f(x_j, \beta) \Big]^2, \tag{B.9}$$

y $\mathcal{L}(x, y) = (y - \hat{y})^2 = (y - f(x, \beta))^2$, con x_j, y_j, los elementos de $\mathbf{X}$ e $\mathbf{Y}$, respectivamente y M la cantidad de dichos elementos.

La forma más sencilla en la que se presenta un problema de regresión es su forma lineal (Montgomery et al., 2021; Seber and Lee, 2012). En una regresión lineal, se busca la combinación lineal de los parámetros que mejor ajuste a la relación entre $\mathbf{X}$ e $\mathbf{Y}$. Para el caso de un conjunto $\mathit{\Gamma}$ de variables unidimensionales:

$$\mathit{\Gamma} = \{(x_1, y_1), (x_2, y_2), (x_3, y_3), \ldots, (x_M, y_M)\}, \tag{B.10}$$

el modelo buscado se denomina regresión lineal "simple", y tiene la forma de una recta:

$$\hat{y}_j = f(x_j, \beta) = \beta_1 x_j + \beta_0, \quad j = 1, \ldots, M, \tag{B.11}$$

donde $\beta = (\beta_0, \beta_1)$, siendo β_0 y β_1 el término independiente (*bias*) y la pendiente de la recta, respectivamente. En cambio, cuando el conjunto de variables $\mathbf{X}$ e $\mathbf{Y}$ contiene magnitudes vectoriales, $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_M)$ e $\mathbf{Y} = (\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_M)$, el problema toma la forma:

$$\begin{aligned} \hat{\mathbf{y}}_j &= \beta_0 + \beta_1\, x_{1j} + \beta_2\, x_{2j} + \beta_3\, x_{3j} + \cdots + \beta_n\, x_{nj} = \\ &= \sum_{i=0}^{n} \beta_i x_{ij} = \mathbf{x}_j \cdot \beta, \quad j = 1, \ldots, M, \end{aligned} \tag{B.12}$$

donde $\mathbf{x}_j = (1, x_{1j}, \ldots, x_{nj})$ es la j-ésima instancia del conjunto $\mathbf{X}$ (vector fila) y $\beta = (\beta_0, \ldots, \beta_n)$ es el vector columna de parámetros del modelo lineal[**]. En forma más general, la ecuación B.12 puede expresarse como:

$$\hat{\mathbf{Y}}_{[M,Q]} = f(\mathbf{X}, \beta) = \mathbf{X}_{[M,P]}\, \beta_{[P,Q]}. \tag{B.13}$$

Notar que si el conjunto $\mathbf{X}$ tiene dimensión $M \times P$, con $P = n + 1$, e $\mathbf{Y}$ tiene dimensión $M \times Q$, luego, β deberá tener dimensión $P \times Q$. Luego $f : \mathbb{R}^P \to \mathbb{R}^Q$. Este tipo de regresión suele llamarse regresión lineal "múltiple".

Encontrar el conjunto de parámetros que minimicen la función de costo $\mathcal{E}$ definida en la ecuación B.9 es equivalente a encontrar sus puntos críticos respecto de cada uno de los parámetros. El problema se reduce a resolver las ecuaciones:

$$\frac{\partial \mathcal{E}}{\partial \beta_i} = 0, \ \ \forall i. \tag{B.14}$$

[**]Nota: Las instancias de entrenamiento (o elementos del conjunto $\mathbf{X}$) se identifican con el subíndice j, mientras que las componentes de cada instancia lo hacen con el subíndice i.

Para el caso particular de una regresión lineal con variables unidimensionales, la ecuación B.14 lleva a

$$\begin{aligned}\beta_1 &= \frac{\sum_j (x_j - \bar{x})(y_j - \bar{y})}{\sum_j (x_j - \bar{x})^2}, \\ \beta_0 &= \bar{y} - \beta_1 \bar{x},\end{aligned} \tag{B.15}$$

donde $\bar{x}$ e $\bar{y}$ son los promedios de las variables en los conjuntos $\mathbf{X}$ e $\mathbf{Y}$, respectivamente

Para el caso de una regresión múltiple, se puede reescribir el problema de minimización B.9 como:

$$\begin{aligned}\arg\min_{\beta} \mathcal{E}(\beta) = \arg\min_{\beta} \frac{1}{M} \sum_j \mathcal{L}(\mathbf{y}_j, \mathbf{x}_j) = \\ = \arg\min_{\beta} \left[\frac{1}{M} \sum_j \Big(\mathbf{y}_j - \mathbf{x}_j \cdot \beta) \Big)^2 \right] = \\ = \arg\min_{\beta} \left[\frac{1}{M} (\mathbf{Y} - \mathbf{X}\beta)^T (\mathbf{Y} - \mathbf{X}\beta) \right].\end{aligned} \tag{B.16}$$

Ahora, si diferenciamos utilizando álgebra de matrices (Petersen and Pedersen, 2008) respecto de β, se obtienen las ecuaciones "normales" (Hastie et al., 2009):

$$\mathbf{X}^T(\mathbf{Y} - \mathbf{X}\beta) = 0, \tag{B.17}$$

quedando β determinado de la forma:

$$\beta = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y}, \tag{B.18}$$

donde

$$\beta_{P\times Q} = \begin{pmatrix} \beta_{0,1} & \beta_{0,2} & \beta_{0,3} & \cdots & \beta_{0,Q} \\ \beta_{1,1} & \beta_{1,2} & \beta_{1,3} & \cdots & \beta_{1,Q} \\ \beta_{2,1} & \beta_{2,2} & \beta_{2,3} & \cdots & \beta_{2,Q} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \beta_{n,1} & \beta_{n,2} & \beta_{n,3} & \cdots & \beta_{n,Q} \end{pmatrix},$$

$$\mathbf{X}_{M\times P} = \begin{pmatrix} 1 & x_{1,1} & x_{1,2} & \cdots & x_{1,n} \\ 1 & x_{2,1} & x_{2,2} & \cdots & x_{2,n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{M,1} & x_{M,2} & \cdots & x_{M,n} \end{pmatrix}, \mathbf{Y}_{M\times 1} = \begin{pmatrix} y_{1,1} & y_{1,2} & \cdots & y_{1,Q} \\ y_{2,1} & y_{2,2} & \cdots & y_{2,Q} \\ \vdots & \vdots & \ddots & \vdots \\ y_{M,1} & y_{M,2} & \cdots & y_{M,Q} \end{pmatrix}$$

con $P = n + 1$, la dimensión del dato de entrada (que define el número de parámetros en β).

En cualquiera de los casos anteriores, asumimos que el error cometido en la estimación del modelo será aceptable, desde el punto de vista estadístico, siempre que los datos se comporten linealmente[*]. Sin embargo, esto no siempre es posible. Cuando la relación entre variables no puede expresarse en términos de una relación lineal entre los parámetros de un modelo, el problema se convierte en una regresión de tipo no-lineal. En este caso, al menos una de las derivadas que se desprenden al buscar los puntos críticos (en este caso, pueden existir varios puntos estacionarios del problema) depende de alguno de los parámetros de β (Montgomery et al., 2021). A diferencia de la regresión lineal, este tipo de problemas puede no tener una solución cerrada y, por ende, la misma suele buscarse de manera iterativa.

Descenso de gradiente: descenso de gradiente estocástico, momento & ADAM

El "descenso de gradiente" (DG) es uno de los métodos más utilizados a la hora de resolver problemas de optimización no-lineal. Es un algoritmo de optimización de primer orden utilizado para encontrar mínimos locales de una función diferenciable $f(\beta)$[**]. La idea principal del método es encontrar un mínimo dirigiéndose iterativamente en una dirección en la que la función tome valores cada vez más pequeños. Así, el método ira generando una serie de valores β_0, β_1, β_2, ..., en una dirección que "descienda", o disminuya los valores de mi función de costo.

En principio, el algoritmo debe iniciarse con una aproximación al mínimo β_0 y converge cuando ya no se detecte un descenso, o en un punto β tal que $\nabla_\beta f$ sea suficientemente próximo a cero (Snyman, 2005). Esto último implicaría haber llegado a un punto crítico o mínimo (local o global), o en el peor escenario, un punto "silla". En la práctica, suele terminarse el algoritmo en la iteración n que alcance uno o alguna combinación de ciertos criterios de corte. Estos pueden ser que $\| \beta_{n+1} - \beta_n \| \leq \varepsilon_1$, $\| \nabla_\beta f_n \| \leq \varepsilon_2$ o $\| f_{n+1} - f_n \| \leq \varepsilon_3$.

Debido a que el gradiente de la función indica la dirección de máxima velocidad de crecimiento, si se considera una función $f : \mathbb{R}^N \to \mathbb{R}$, el método propone moverse a un nuevo conjunto de parámetros según (Goodfellow et al., 2016):

$$\beta_{n+1} = \beta_n - \alpha \nabla_\beta f(\mathbf{x}, \beta_n), \quad n \geq 0, \tag{B.19}$$

donde $\nabla_\beta f = \left[\frac{\partial f}{\partial \beta_1}, \ldots, \frac{\partial f}{\partial \beta_N}\right]$ es un vector que contiene todas las derivadas parciales de f. La cantidad α se denomina "paso" o *step size*. Define el tamaño del paso y es uno de los parámetros más importantes del algoritmo. Valores muy grandes de α pueden provocar oscilaciones o inestabilidades, mientras que valores muy pequeños implican un decrecimiento

[*]Estrictamente, para que el modelo sea estadísticamente válido deben hacerse varias suposiciones sobre el dato, los residuos y el modelo (ver referencias citadas).

[**]Por simplicidad, solo expresamos $f()$ en función de sus parámetros

lento. En la práctica, se suelen realizar varias optimizaciones utilizando diferentes valores de α y se elige aquel que dio mejores resultados. Generalmente, el valor óptimo es un escalar positivo pequeño, $\alpha \ll 1$.

En DG , los valores de los elementos en β se actualizan como:

$$\beta_{n+1} = \beta_n - \alpha \nabla_\beta \mathcal{E}(\beta_n), \quad n \geq 1, \tag{B.20}$$

donde

$$\nabla_\beta \mathcal{E}(\beta_n) = \frac{1}{M}\left[\sum_{i=1}^{M} \frac{\partial \mathcal{L}(y_i, x_i)}{\partial \beta_0}, \sum_{i=1}^{M} \frac{\partial \mathcal{L}(y_i, x_i)}{\partial \beta_1}, \ldots, \sum_{i=1}^{M} \frac{\partial \mathcal{L}(y_i, x_i)}{\partial \beta_N}\right]. \tag{B.21}$$

Como puede observarse, en cada iteración n, este algoritmo de optimización requiere de la evaluación de la función de error $\mathcal{L}$ en todas las instancias del dato. Este tipo de optimización se denomina descenso de gradiente de tipo *batch*.

En este punto, será útil definir el concepto de "época" (*epoch*) y distinguirlo del de una "iteración". En las circunstancias del aprendizaje automático se denomina como "época" al conjunto de operaciones que ejecuta un algoritmo al utilizar la totalidad del dato de entrenamiento. Para el caso anterior vemos que una época es equivalente a una iteración, ya que el algoritmo evalúa todo el dato en cada iteración. Volviendo a la ecuación B.21, si el dato es relativamente pequeño (generalmente unos cientos o miles de datos), el costo computacional de cada época puede ser aceptable. Sin embargo, cuando el dato es grande (centenas de miles o millones de datos), el costo computacional requerido por cada época se vuelve inviable, en especial cuando un entrenamiento suele involucrar muchas épocas. La solución a este problema es aplicar "descenso de gradiente estocástico" (SGD, por sus siglas en inglés). En SGD, los parámetros en β se actualizan luego de evaluar la función de error en cada uno de los datos. En este caso, para terminar una época necesitaríamos iterar sobre cada una de las instancias del dato de entrenamiento. De igual manera, la cantidad de actualizaciones que se realizaron sobre los parámetros al completar una época es igual al número de instancias en el conjunto de entrenamiento. A simple vista, esto no parecería ofrecer una mejora en el rendimiento del algoritmo, sin embargo, a diferencia de lo que ocurre en GD, para las últimas iteraciones de cada época se utilizan parámetros que ya fueron optimizados en iteraciones anteriores. Esto es, mientras que en GD de tipo *batch* se obtiene una única actualización de parámetros por cada época, en SGD se tienen tantas actualizaciones como instancias en el conjunto de entrenamiento. Habiendo dicho esto, en la práctica tampoco suele utilizarse el método SGD en forma estricta. En cambio, se procede dividiendo el dato en subconjuntos de tamaño fijo (generalmente en el orden de cientos o pocos miles de datos) denominados *mini-batchs* y luego se calcula el valor promedio del gradiente en cada uno de estos subconjuntos. En forma precisa, en cada iteración se calcula el gradiente a partir de un subconjunto $\mathbb{A}$ con M' instancias que son extraídas en forma

aleatoria (con distribución uniforme):

$$\nabla_\beta \mathcal{E}_{M'} = \frac{1}{M'} \nabla_\beta \left[\sum_{j=1}^{M'} \mathcal{L}(y_j, x_j) \right]. \tag{B.22}$$

Notar que, mientras el costo computacional de calcular los gradientes sobre el dato completo depende de la cantidad total de instancias, mientras que utilizando *mini-batches*, el costo computacional es lineal con el tamaño (fijo) del *mini-batch*. Este tipo de optimización también suele llamarse descenso de gradiente de tipo *mini-batch* (*mini-batch*-GD). Con esta estrategia, se dice que una época estará completa una vez que se hayan utilizado todos los *mini-batchs* extraídos del dato. Es importante mencionar que si el tamaño del *mini-batch* es igual a 1, entonces *mini-batch*-GD=SGD. De igual forma, si el tamaño del *mini-batch* es igual al conjunto de entrenamiento, *mini-batch*-GD=GD. En general, si el tamaño del *mini-batch* es muy pequeño, el promedio del gradiente suele fluctuar generando oscilaciones indeseadas en el conjunto de parámetros que se está optimizando, mientras que si se elije muy grande se compromete la eficiencia que el método ofrece.

A pesar de que *mini-batch*-GD ofrece algunas ventajas de rendimiento sobre GD y SGD, todos estos presentan dificultades para encontrar el valor óptimo en regiones "alargadas". Esto es, en regiones sobre la superficie de la función de error donde la pendiente es mucho más empinada en una dirección frente a las demás (Ruder, 2016) causando oscilaciones en las sucesivas aproximaciones del vector gradiente. Para prevenir este problema, se suele modificar el algoritmo agregando un término de "momento" (Qian, 1999), cuyo objetivo es acelerar los gradientes en la dirección correcta, evitando estas oscilaciones y por ende, resultando en una convergencia más rápida. Este término está relacionado con el concepto de *exponentially weighted averages* (EWA), que son promedios móviles que suelen utilizarse para suavizar series de datos. La expresión matemática para un promedio de este tipo está dada por:

$$v_t = \zeta v_{t-1} + (1 - \zeta) s_t \tag{B.23}$$

donde s_t es la serie de datos que se quiere suavizar y v_t es la serie suavizada. El parámetro $\zeta \in [0, 1]$ está relacionado con la cantidad de muestras pasadas que se utilizan en el suavizado. Tomar valores de $\zeta \simeq 1$ implica utilizar muchas muestras pasadas, resultando en series más suaves, y valores $\zeta \simeq 0$ implica tomar menos muestras pasadas, lo cual resulta en series más ruidosas. Si la serie de puntos que se desea suavizar es la serie de los gradientes $\nabla_\beta \mathcal{E}(\beta_n)$, podemos utilizar esta idea para modificar la ecuación B.20 quedando como:

$$\begin{aligned} v_n &= \gamma v_{n-1} + \alpha \nabla_\beta \mathcal{E}(\beta_n) \\ \beta_{n+1} &= \beta_n - v_n, \quad n \geq 1. \end{aligned} \tag{B.24}$$

Notar que v_n es un valor suavizado que acumula muestras pasadas y por ende, incrementa su valor para las dimensiones donde el gradiente apunta en la misma dirección y disminuye

en las demás. Así, se reducen las oscilaciones y se acelera el rendimiento de GD. El valor más utilizado para γ es usualmente cercano a 0.9.

El último método que veremos será "ADAM" (Kingma and Ba, 2014)*, que combina el término de momento con tasas de aprendizaje adaptivas, implementadas en métodos como "Adadelta" (Zeiler, 2012) o "RMSprop" (Tieleman and Hinton, 2012). Los autores de este método proponen realizar un EWA de los gradientes así como también del cuadrado de los gradientes (esto último como en Adadelta y RMSprop). La implementación del método queda descripta por las ecuaciones

$$\begin{aligned} m_n &= \gamma_1 m_{n-1} + (1-\gamma_1)\nabla_\beta \mathcal{E}(\beta_n) \\ v_n &= \gamma_2 v_{n-1} + (1-\gamma_2)\nabla^2_\beta \mathcal{E}(\beta_n) \\ \beta_{n+1} &= \beta_n - \alpha \frac{m_n}{\sqrt{v_n}+\epsilon}, \quad n \geq 1 \end{aligned} \tag{B.25}$$

con $\gamma_1 = 0.9$, $\gamma_2 = 0.999$ y $\epsilon \sim 10^{-8}$.

Como vimos, existe una gran variedad de métodos que pueden ser utilizados para entrenar una NN. Dicho esto, los métodos de la familia de "descenso de gradiente" no son los únicos métodos utilizados para esta tarea. En efecto, en los últimos tiempos aumento considerablemente la utilización de métodos de la familia de los algoritmos de tipo "evolutivos"*, como *differential evolution* y *particle swarm optmization*. Si el lector está interesado en este tema, puede referirse a Mirjalili (2019). En este libro se trata en detalle la implementación de varios de estos métodos en la inteligencia artificial y se detallan los desafíos de su implementación en este campo. A pesar de que no existe consenso sobre que método elegir para cada uno de los problemas que se ataquen, los algoritmos evolutivos han probado tener ventajas en algunos casos particulares (Mirjalili, 2019). Este tipo de métodos también resultan atractivos para atacar el problema de la optimización de los hiper-parámetros de entrenamiento (Loussaief and Abdelkrim, 2018). Aun así, la elección de los métodos de tipo GD sigue siendo la más aceptada. Esto último se debe a su simplicidad, que los convierte en métodos fácilmente programables e interpretables.

Por otro lado, tampoco existe consenso sobre cuál es el mejor método para utilizar dentro de a familia del GD. De este conjunto, los más utilizados son aquellos de la familia del SGD con momento y sus variaciones (Schaul et al., 2013). Por su parte, Goodfellow et al. (2016) indican que una elección razonable para un algoritmo de optimización puede ser alguna variación de SGD con momento (por ejemplo ADAM) y una tasa de aprendizaje variable decreciente.

---

*ADAM: *Adaptive moment*.

*Si bien se utilizan estos métodos desde la década del 90 para el desarrollo de este campo, el aumento de su aplicación es considerable en los últimos años.

Procedimiento general de un entrenamiento

Consideremos un conjunto $\mathbb{D}$ formado por M instancias "pares" de entrenamiento, $(\mathbf{x}_j, \mathbf{y}_j)$, $j = 1, \ldots, M$:

$$\Gamma = \Big\{(\mathbf{x}_1, \mathbf{y}_1), (\mathbf{x}_2, \mathbf{y}_2), \ldots, (\mathbf{x}_M, \mathbf{y}_M)\Big\}, \tag{B.26}$$

con $\mathbf{x}_j \in \mathbb{R}^P$ e $\mathbf{y}_j \in \mathbb{R}^Q$, siendo P y Q las dimensiones de ambas cantidades vectoriales.

Sea $NN(\mathbf{x}; \mathbf{w})$ una red neuronal de pesos $\mathbf{w}$ alimentada hacia adelante y formada por L capas densas $l_k = \{w_{ip}^k\}$, de las cuales $L-1$ serán capas ocultas y la última será la capa de salida[**]. Sea además, w_{ip}^k con $p = 1, \ldots, W_{k-1}$, el peso que conecta el nodo i de la capa l_{k-1} con el nodo p de la capa l_k y siendo w_{i0}^k ($p = 0$) el *bias* de esta última capa. Finalmente, sea ϕ_k la función de activación que conecta las capas l_k y l_{k+1}.

Al atravesar las capas de esta red neuronal, cada elemento $\mathbf{x}_j$ del dato de entrada se mapeará a una salida $\hat{\mathbf{y}}_j$ de forma:

$$NN(\mathbf{x}_j; \mathbf{w}) = \hat{\mathbf{y}}_j. \tag{B.27}$$

Dentro de NN, cada una de las capas l_k es responsable de aplicar un promedio pesado de los elementos de su entrada seguido de una activación no-lineal ϕ_k produciendo una salida que será la entrada de la capa siguiente, l_{k+1}. Esta operación se repetirá a lo largo de todas las capas. Visto de otra manera, el proceso que sufre el dato al atravesar la totalidad de la red puede pensarse como el equivalente a aplicar una función especial creada como una composición de funciones no-lineales. En la bibliografía, suele denominarse a esta función como "función de red" (Rojas, 1996).

El objetivo de un entrenamiento es modificar el conjunto de pesos $\mathbf{w}$ de forma tal que cada entrada $\mathbf{x}_j$ se mapee unívocamente a la salida deseada $\mathbf{y}_j$. Matemáticamente, esto se traduce en minimizar una función que mida la distancia entre la salida deseada $\mathbf{y}_j$, y la obtenida $\hat{\mathbf{y}}_j$, $\forall j$. Si nos reducimos a un problema de tipo regresivo, y siguiendo la ecuación B.16, esta función será:

$$\mathcal{E}(\mathbf{w}) = \frac{1}{M} \sum_{j=1}^{M} \mathcal{L}_j, \tag{B.28}$$

donde:

$$\mathcal{L}_j = \frac{1}{Q} \sum_{p=1}^{Q} (y_{jp} - \hat{y}_{jp})^2, \tag{B.29}$$

con y_{jp} e $\hat{y}_{jp}$ las p-ésimas componentes de las salidas deseadas y aproximadas, respectivamente.

[**]Recordar que la capa de entrada, IL, no suele considerarse como una capa densa: $L = (L-1)HL + OL$.

Se considera que un entrenamiento es exitoso si, ajustados los pesos del conjunto $\mathbf{w}$, la red que es capaz de generalizar. Esto es, que pueda predecir una salida "correcta" aun siendo evaluada en instancias $\mathbf{x}_i$ desconocidas (no pertenecientes al dato de entrenamiento).

Así descripto, es evidente que un entrenamiento constituye un problema de optimización. El método utilizado para atacarlo es alguna variante de descenso de gradiente y la mayoría de las veces se utiliza un SGD de tipo *mini-batch*. De acuerdo a las ecuaciones B.20 y B.21, cada iteración n del algoritmo requerirá del gradiente de la función de costo $\mathcal{E}$, lo que implica calcular la derivada respecto de cada uno de los pesos w_{ip}^k de la red, es decir $\partial\mathcal{E}/\partial w_{ip}^k$. Una vez construido el gradiente y siguiendo la ecuación B.20, los pesos de la red se modifican según (omitiendo el índice k):

$$w_{ip}^{n+1} = w_{ip}^n - \alpha\,\Delta w_{ip}^n, \quad \text{con } \Delta w_{ip}^n = \frac{\partial \mathcal{E}^n}{\partial w_{ip}^n}. \tag{B.30}$$

Vemos que para que esto sea válido, primero se debe garantizar que tanto $\mathcal{E}$ como cada una de las funciones de activación que componen a NN sean diferenciables.

Pensando ahora en una NN cualquiera, es fácil ver que el gradiente de la función de costo $\mathcal{E}$ respecto de los pesos de la OL se pueden calcular en forma directa. Esto es equivalente a pensar que sus pesos están inmediatamente "por debajo" de la evaluación de la función de costo. Sin embargo, a medida que nos adentramos en las profundidades de la red, el cálculo de las derivadas parciales de $\mathcal{E}$ respecto de los pesos w_{ip}^k debe realizarse mediante la aplicación reiterativa de la regla de la cadena. Si la red que se entrena es "superficial" (con pocas capas) o cuenta con un puñado de pesos, es posible realizar dichas derivadas parciales manualmente y en forma analítica. A medida que la red crece en profundidad y ancho, realizar las derivaciones analíticamente puede volverse inviable aun para una computadora moderna. Por esta razón, el problema del cómputo del gradiente de la función de costo es el principal desafío a la hora de entrenar una red neuronal.

En la actualidad, el algoritmo mediante el cual se calculan los gradientes para ser utilizados en la optimización es llamado: *Backpropagation*.

El método de *Backpropagation* fue presentado a lo largo de la segunda mitad del siglo XX en varias oportunidades y de forma independiente por varios autores, sin embargo, no fue hasta la publicación de Rumelhart et al. (1986) que el algoritmo tomó relevancia. Si el lector desea tener una noción más detallada de la evolución de este campo de estudio, se recomienda leer también Goodfellow et al. (2016).

En términos matemáticos, el método no es más que una implementación muy eficiente y limpia de realizar la regla de la cadena del cálculo sobre una composición de funciones y por ello, su aplicación no está limitada al entrenamiento de redes neuronales. Sin embargo, en lo que sigue solo nos enfocaremos a su análisis en el marco de esta disciplina. Por otro lado, al ser un método central en el entrenamiento de una NN, el mismo se encuentra ampliamente documentado, y por ello, nos limitaremos a describir las partes fundamentales del algoritmo

y evitaremos los desarrollos más estrictos que suelen relacionarse con este algoritmo. Si el lector desea una explicación más detallada puede referirse a libros específicos de inteligencia artificial, aprendizaje automático o redes neuronales como Rojas (1996); Hu et al. (2002); Goodfellow et al. (2016); Patterson and Gibson (2017); Aggarwal (2018); Géron (2019), o también artículos como Hecht-Nielsen (1992); Ionescu et al. (2015); Sadowski (2016); Mishachev (2017), por mencionar algunos.

Nos disponemos ahora a proveer una deducción de las ecuaciones fundamentales del algoritmo de *Backpropagation*. Por simplicidad, consideraremos la dimensión de la salida deseada $Q = 1$, es decir que será un valor escalar real[*].

Con esto en cuenta, la ecuación B.29 tomará la forma

$$\mathcal{L}_j = (y_j - \hat{y}_j)^2. \tag{B.31}$$

En virtud de nuestra nueva notación, también será de utilidad re-definir la ecuación B.4 como

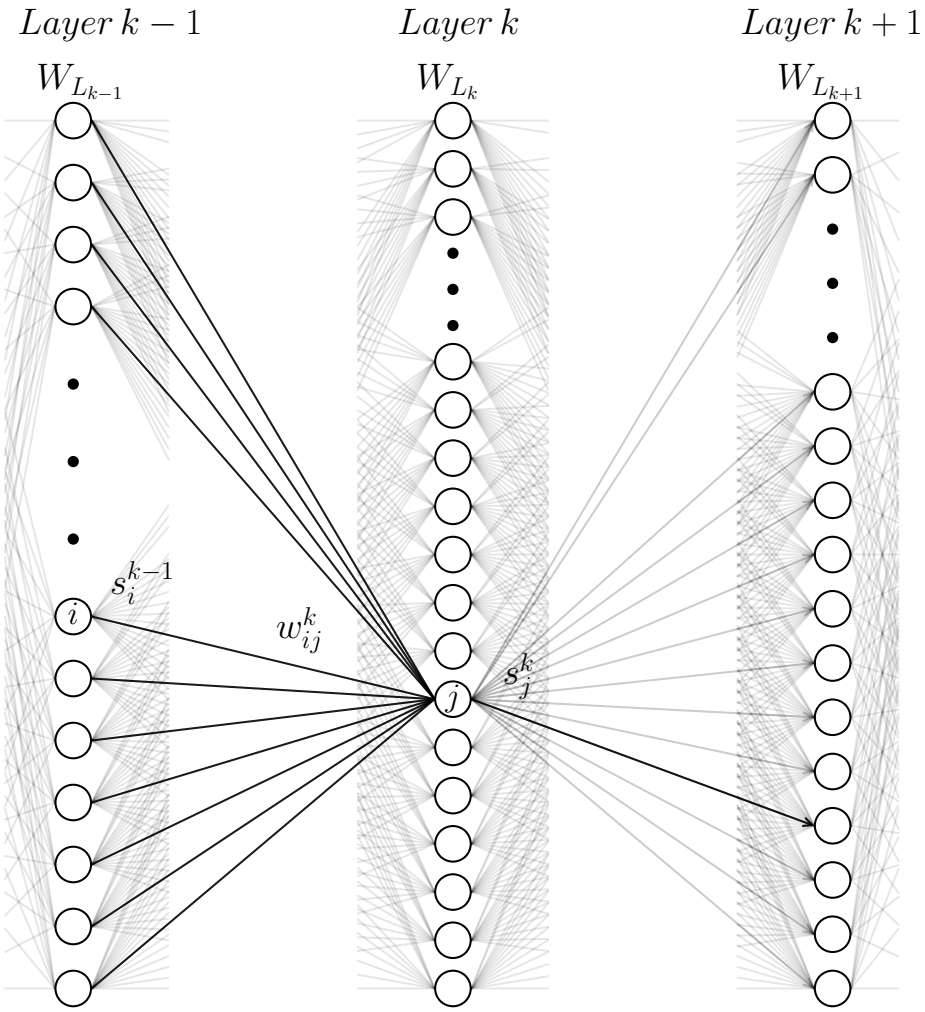
$$z_p^k = w_{0j}^k + \sum_{l=1}^{W_{k-1}} w_{lp}^k s_l^{k-1} = \sum_{p=0}^{W_{k-1}} w_{lp}^k s_l^{k-1}, \tag{B.32}$$

válida para todo par de datos $(\mathbf{x}_j, \mathbf{y}_j)$ con $s_l^{k-1} = \phi^{k-1}(z_l^{k-1})$ la salida correspondiente al nodo l de la capa $k-1$ y definiendo $s_0^{k-1} = 1$, $\forall k$.

La ecuación B.32 representa el núcleo de funcionamiento de una NN ya que define en forma explícita el flujo (procesamiento y transferencia) del dato a través de la red. En términos sencillos, expresa la relación matemática que existe entre la salida de la capa $k-1$, esto es s_l^{k-1}, como entrada de la capa siguiente k. Para el caso particular de la primer HL ($k = 1$), su dato de entrada estaría definido por la salida de la IL. Sabemos que por definición, la IL no es exactamente un perceptrón, ya que sin ejercer operación alguna sobre el dato de entrada, solo lo distribuye sobre los nodos de la primer HL. Haciendo abuso del lenguaje podría pensarse como un "perceptrón" con $\phi^0()$ siendo la identidad y $w_{ip}^0 \equiv 1\ \forall i, p$. En este caso la ecuación B.32 quedaría

$$\begin{aligned} z_p^1 = w_{0p}^1 + \sum_{l=1}^{W_0} w_{lp}^1 s_l^0 = w_{0p}^1 + \sum_{l=1}^{W_0} w_{lj}^1 \phi^0(z_l^0) = \\ = w_{0p}^1 + \sum_{l=1}^{W_0} w_{lp}^1 z_l^0 = w_{0p}^1 + \sum_{l=1}^{W_0} w_{lp}^1 \left(w_{lp}^0 x_l\right) = \\ = w_{0p}^1 + \sum_{p=1}^{W_0} w_{lp}^1 x_l = \end{aligned} \tag{B.33}$$

[*]La deducción de las ecuaciones pertinentes al caso más general, esto es, el caso matricial, siguen el mismo principio pero demandan considerablemente más espacio, por lo que su deducción queda por fuera del alcance de esta Tesis. Si el lector está interesado en la misma, puede referirse a algunas de las referencias citadas en el texto.



$$s_j^k = f(z_j^k) = f(w_{0j}^k + \sum_{l=1}^{W_{L_{k-1}}} w_{lj}^k s_l^{k-1})$$

Figura B.3: Tres capas consecutivas de una red neuronal de múltiples capas. Se muestra cómo las salidas de todos los nodos de la capa $k-1$, s_l^{k-1} son usados como entrada de cada uno de los nodos de la siguiente capa. En particular, se indica cómo la salida s_i^{k-1} del nodo i de la capa $k-1$ se conecta mediante el peso w_{ip}^k con el nodo p de la capa k. También, se muestra cómo la salida escalar (única) del nodo p de la capa k se distribuye a todos los nodos de la capa siguiente.

donde $s_l^0 = \phi^0(z_l^0) = z_l^0 = x_l$ es la componente l-ésima del dato de entrada y W_0 es el número de pesos de la IL, que por ende, debe coincidir con la dimensión de $\mathbf{x}$, P.

Con estas expresiones, podemos utilizar la ecuación B.28 y realizar el cálculo del gradiente:

$$\frac{\partial \mathcal{E}(\mathbf{w})}{\partial w_{ip}^k} = \frac{1}{M}\frac{\partial}{\partial w_{ip}^k}\left[\sum_{d=1}^{M}(y_d - \hat{y}_d)^2\right] = \frac{1}{M}\sum_{d=1}^{M}\frac{\partial (y_d - \hat{y}_d)^2}{\partial w_{ip}^k}, \tag{B.34}$$

lo cual implica encontrar las expresiones para las derivadas:

$$\frac{\partial \mathcal{L}_d}{\partial w_{ip}^k} = \frac{\partial (y_d - \hat{y}_d)^2}{\partial w_{ip}^k}. \tag{B.35}$$

Utilizando regla de la cadena y la expresión B.32 se obtiene:

$$\begin{aligned}\frac{\partial \mathcal{L}_d}{\partial w_{ip}^k} &= \frac{\partial \mathcal{L}_d}{\partial z_p^k}\frac{\partial z_p^k}{\partial w_{ip}^k} = \\ &= \delta_{dp}^k \frac{\partial z_p^k}{\partial w_{ip}^k}.\end{aligned} \tag{B.36}$$

El primer factor $\delta_{dp}^k = \partial \mathcal{L}_d / z_p^k$ es comúnmente denominado error y describe cómo cambia la función de costo respecto del promedio pesado correspondiente al nodo p de la capa k. El segundo factor, en cambio, nos muestra como cambia este mismo promedio pesado con respecto cada uno de sus pesos individuales. Si se profundiza en este último factor, obtenemos:

$$\begin{aligned}\frac{\partial \mathcal{L}_d}{\partial w_{ip}^k} &= \delta_{dp}^k \frac{\partial}{\partial w_{ip}^k} z_p^k = \delta_{dp}^k \frac{\partial}{\partial w_{ip}^k}\left(\sum_{l=0}^{W_{k-1}} w_{lp}^k s_l^{k-1}\right) = \\ &= \delta_{dj}^k \sum_{l=0}^{W_{k-1}} \frac{\partial\left(w_{lp}^k s_l^{k-1}\right)}{\partial w_{ip}^k} = \delta_{dp}^k \sum_{l=0}^{W_{k-1}} \frac{\partial\left(w_{lp}^k\right)}{\partial w_{ip}^k} s_l^{k-1} = \\ &= \delta_{dp}^k s_i^{k-1},\end{aligned} \tag{B.37}$$

donde usamos que:

$$\frac{\partial\left(w_{lp}^k\right)}{\partial w_{ip}^k} = \begin{cases} 1 & l = i \\ 0 & l \neq i. \end{cases} \tag{B.38}$$

Nuevamente se tiene una relación entre el error en el nodo p de la capa k, δ_{dp}^k, y la salida correspondiente al nodo i de la capa $k-1$, s_i^{k-1}. La Figura B.3 muestra gráficamente la relación que existe entre los nodos de una capa y la siguiente. Esto no podría ser de otra manera, ya que w_{ip}^k es la conexión entre ambos nodos.

La ecuación B.38 permite calcular el gradiente buscado, pero aun no es suficientemente explícita, ya que no se conoce la forma de δ_{dp}^k. Como vemos, esta ecuación muestra que la relación entre un cambio en la función de error $\mathcal{L}_d$ y un cambio en los pesos w_{ip}^k de la capa k depende de la salida de la capa anterior s_i^{k-1}. Por ende, para conocer el gradiente de $\mathcal{E}$ respecto de un peso se debe conocer el valor de las salidas de las capas anteriores. En lo siguiente, deduciremos la expresión de δ_{dp}^k y veremos que la misma depende de los valores en capas siguientes. Es por esta razón que el gradiente debe calcularse comenzando por las derivadas parciales desde la función de costo hacia atrás. Esto es, pasando primero por la OL y luego adentrándose en las profundidades de las HLs. Siguiendo con la ecuación B.31, para la OL($k = L$) se tiene:

$$\begin{aligned}\delta_{d1}^L = \frac{\partial \mathcal{L}_d}{\partial z_1^L} &= \frac{\partial}{\partial z_1^L}[y_d - \hat{y}_d]^2 = \\ &= \frac{\partial}{\partial z_1^L}\left[y_d - \phi^L(z_1^L)_d\right]^2 = \\ &= -2\left[y_d - \phi^L(z_1^L)_d\right]\frac{\partial \phi^L(z_1^L)_d}{\partial z_1^L}.\end{aligned} \tag{B.39}$$

Notar que hemos considerado que la salida $z_p^L = z_1^L \in \mathbb{R}^1$. Si se reemplaza la expresión B.39 en B.36 se obtiene:

$$\begin{aligned}\frac{\partial \mathcal{L}_d}{\partial w_{i1}^L} = \frac{\partial \mathcal{L}_d}{\partial z_1^L}\frac{\partial z_1^L}{\partial w_{i1}^L} &= \\ &= -2\left[y_d - \phi^L(z_1^L)_d\right]\frac{\partial \phi^L(z_1^L)_d}{\partial z_1^L}\frac{\partial z_1^L}{\partial w_{i1}^L} = \\ &= -2\left[y_d - \phi^L(z_1^L)_d\right]\frac{\partial \phi^L(z_1^L)_d}{\partial z_1^L}\, s_i^{L-1}.\end{aligned} \tag{B.40}$$

A continuación, para obtener el gradiente de $\mathcal{L}_d$ respecto de todos los pesos, se debe realizar el cálculo de $\partial \mathcal{L}_d / \partial w_{ip}^k$ para el resto de las capas. Esto significa que se debe comenzar por la última de las HLs ($k = L-1$) y con la regla de la cadena hasta llegar al dato de entrada. Así, derivaremos las capas $k = L-1, L-2, \ldots, 1$. Para hacer esto, nos ayudaremos planteando

los diferentes términos de error de la siguiente manera:

$$\begin{aligned}
\delta_1^L &= \frac{\partial \mathcal{L}_d}{\partial z_1^L} \\
\delta_p^{L-1} &= \sum_{l=1}^{W_L} \frac{\partial \mathcal{L}_d}{\partial z_l^L} \frac{\partial z_l^L}{\partial z_p^{L-1}} \\
\delta_p^{L-2} &= \sum_{l=1}^{W_{L-1}} \frac{\partial \mathcal{L}_d}{\partial z_l^{L-1}} \frac{\partial z_l^{L-1}}{\partial z_p^{L-2}} \\
&\ddots \\
\delta_p^1 &= \sum_{l=1}^{W_2} \frac{\partial \mathcal{L}_d}{\partial z_l^2} \frac{\partial z_l^2}{\partial z_p^1}.
\end{aligned} \tag{B.41}$$

Así, está claro que existe una relación recursiva "hacia atrás", que puede expresarse como:

$$\delta_p^k = \sum_{l=1}^{W_{k+1}} \delta_l^{k+1} \frac{\partial z_l^{k+1}}{\partial z_p^k} \quad k < L. \tag{B.42}$$

Utilizamos ahora la ecuación B.32 para reemplazar z_l^{k+1} en la ecuación B.42 obteniendo:

$$\begin{aligned}
\delta_p^k &= \sum_{l=1}^{W_{k+1}} \delta_l^{k+1} \frac{\partial}{\partial z_p^k} \left(w_{0l}^{k+1} + \sum_{r=1}^{W_k} w_{rl}^{k+1} s_r^k \right) = \\
&= \sum_{l=1}^{W_{k+1}} \delta_l^{k+1} \left[\frac{\partial w_{0l}^{k+1}}{\partial z_p^k} + \sum_{r=1}^{W_k} \frac{\partial \left(w_{rl}^{k+1} s_r^k \right)}{\partial z_p^k} \right] = \\
&= \sum_{l=1}^{W_{k+1}} \delta_l^{k+1} \left[\frac{\partial \left(w_{pl}^{k+1} s_p^k \right)}{\partial z_p^k} \right] = \\
&= \sum_{l=1}^{W_{k+1}} \delta_l^{k+1} w_{pl}^{k+1} \frac{\partial \phi^k(z_p^k)}{\partial z_p^k} = \\
&= \frac{\partial \phi^k(z_p^k)}{\partial z_p^k} \sum_{l=1}^{W_{k+1}} \delta_l^{k+1} w_{pl}^{k+1}, \quad k < L.
\end{aligned} \tag{B.43}$$

donde se considera que $\partial w_{0l}^{k+1} / \partial z_p^k = 0$ (el término de *bias* no depende de la ninguna cantidad z_p). Finalmente, se escribe:

$$\frac{\partial \mathcal{L}_d}{\partial w_{ip}^k} = \delta_p^k s_i^{k-1} = \begin{cases} -2 \left(y_d - \phi^L(z_1^L)_d \right) \dfrac{\partial \phi^L(z_1^L)}{\partial z_1^L} s_i^{L-1} & k = L \\ \left[\dfrac{\partial \phi^k(z_p^k)}{\partial z_p^k} \displaystyle\sum_{l=1}^{W_{k+1}} \delta_l^{k+1} w_{pl}^{k+1} \right] s_i^{k-1} & k < L. \end{cases} \tag{B.44}$$

Habiendo entendido esta última ecuación, si se quiere obtener la expresión de $\partial\mathcal{L}_d/\partial w_{ip}^k$ para el caso $y \in \mathbb{R}^{Q>1}$ solo se debe considerar:

$$\delta_p^L = \frac{\partial \mathcal{L}_d}{\partial \phi_p^L} \frac{\partial \phi_p^L}{\partial z_p^L}, \tag{B.45}$$

obteniendo:

$$\frac{\partial \mathcal{L}_d}{\partial w_{ip}^k} = \delta_p^k s_i^{k-1} = \begin{cases} -2\left((y_p)_d - \phi^L(z_p^L)_d\right) \dfrac{\partial \phi^L(z_p^L)}{\partial z_p^L}\, s_i^{L-1} & k = L \\ \left[\dfrac{\partial \phi^k(z_p^k)}{\partial z_p^k} \displaystyle\sum_{l=1}^{W_{k+1}} \delta_l^{k+1} w_{pl}^{k+1}\right] s_i^{k-1} & k < L. \end{cases} \tag{B.46}$$

Una vez calculada esta cantidad $\partial\mathcal{L}_d/\partial w_{ip}^k$, se puede optimizar utilizando descenso de gradiente como:

$$\begin{aligned} (w_{ip}^k)^{(n+1)} &= (w_{ip}^k)^n + \alpha \frac{\partial \mathcal{E}(w_{ip})}{\partial (w_{ip}^k)^n} = \\ &= (w_{ip}^k)^n + \frac{\alpha}{M} \sum_{d=1}^{M} \frac{\partial \mathcal{L}_d}{\partial (w_{ip}^k)^n}, \end{aligned} \tag{B.47}$$

o descenso de gradiente de tipo *mini-batch* como:

$$(w_{ip}^k)^{(n+1)} = (w_{ip}^k)^n + \frac{\alpha}{M'} \sum_{d=1}^{M'} \frac{\partial \mathcal{L}_d}{\partial (w_{ip}^k)^n}, \tag{B.48}$$

donde M' es el número de instancias que componen el *mini-batch*.

A excepción del cálculo explícito de las derivadas parciales de las funciones de activación, que dependerán de qué tipo de funciones se utilicen como tal, la ecuación B.44 define el núcleo del algoritmo de *Backpropagation*. A simple vista, su deducción requirió solamente de la propagación de la regla de la cadena del cálculo elemental. Sin embargo, si uno se propusiese calcular los valores de $\partial\mathcal{L}_d/\partial w_{ip}^k$ tal como lo expresa la ecuación, su costo computacional sería prohibitivo para entrenar redes extensas. Para poder hacerlo de forma eficiente, el algoritmo de *Backpropagation* explota la "naturaleza" de esta ecuación optimizando así el uso de la memoria de la computadora. Esta forma se compone de dos "fases": una fase "hacia adelante" y otra fase "hacia atrás" (esta última es la que da nombre al algoritmo).

Para explicar esta estrategia de dos fases, primero debemos explicar a qué nos referimos con la "naturaleza" de la ecuación. Una observación directa de la ecuación B.44 nos muestra que cada derivada $\partial\mathcal{L}_d/\partial w_{ip}^k$ implica un factor que depende de los errores en las capas posteriores, $k+1$, y otro factor que involucra las salidas de las capas anteriores $k-1$. Nos preguntamos, ¿cada vez que queremos realizar una derivada debemos calcular todos

las derivadas desde la OL hacia la IL, así como también las salidas de cada nodo? La respuesta es no. Aquí entra en juego la estrategia de las fases. Para hacerlo, en cada iteración del entrenamiento y a medida que el dato se propaga desde la IL hacia la OL para finalmente calcular la función $\mathcal{E}$, se guardan en memoria los valores de $s_p^k \ \forall p, k$. Esta operación constituye la fase "hacia adelante". Ahora resta propagar la regla de la cadena desde el final hasta el comienzo. Esta sería la fase "hacia atrás". Si nos fijamos con atención, la mayoría de lo que necesitamos para el cálculo de estos factores fue almacenado en memoria en la fase hacia adelante. Así, todo lo que necesitamos es calcular el error en la última capa y luego, a medida que avanzamos, el error en cada capa k se calcula como el promedio pesado entre los errores calculados para la capa superior $k+1$ (en este punto, ya calculado y guardado en memoria durante la anterior iteración de la fase hacia atrás) y los pesos w_{ip}^{k+1} y finalmente escalados por un factor $\frac{\partial \phi^k(z_p^k)}{\partial z_p^k} s_i^{k-1}$ (cuyos elementos fueron guardados en memoria durante la fase hacia adelante).

Overfitting y *underfitting*

Como se mencionó en la sección anterior, una red entrenada adecuadamente debe tener la capacidad de "generalizar". Esto es, la capacidad de tener un buen desempeño sobre un dato "nuevo", que no ha visto anteriormente (no utilizado para entrenar). Para saber si una red efectivamente puede generalizar sobre un dato nuevo, es necesario evaluarla y medir su respuesta con un dato de esta naturaleza. Esta medida nos dará una medida del error de generalización, llamado también error de testeo. En líneas generales, se entrena una red basándose en el error de entrenamiento, pero su desempeño se mide en función del error de testeo, siendo esto último lo que finalmente más importa. Al dato destinado para realizar la evaluación final se lo denomina dato de "testeo" y generalmente es una pequeña porción del dato original que se apartó desde el comienzo con este propósito. En el pasado, el tamaño de este conjunto respetaba la regla del 70/30, es decir, se utilizaba un 70 % del total del dato original para el entrenamiento y el restante 30 % se separaba para realizar el testeo final, que nos diría qué tan bien puede generalizar nuestra red. En el presente, y debido a que el volumen de dato disponible para resolver una cierta tarea suele ser mucho mayor a la que se disponía tiempo atrás, la tendencia es utilizar una porción más grande para el entrenamiento y solo apartar un $2-10$ % para el testeo. Es importante mencionar que desde el comienzo de un entrenamiento, se está suponiendo que el conjunto de testeo tiene la misma distribución que el dato de entrenamiento (Goodfellow et al., 2016). Es de esperar que si una red es entrenada con un dato perteneciente a una distribución y testeada en un dato con otra diferente, el desempeño de esta red no sea bueno.

Dejando de lado los aspectos técnicos del dato de testeo, una vez que la red fue entrenada para minimizar el error de entrenamiento, se debe verificar que el error de testeo sea también pequeño y sobre todo, cercano al primero. Si luego de un entrenamiento la diferencia entre

los errores es grande, estamos en presencia del fenómeno denominado *overfitting*. Esto nos estaría indicando que el modelo es capaz de representar o entender muy bien el dato de entrenamiento pero no puede realizar inferencias o predicciones correctas sobre el dato de testeo. En otras palabras, es incapaz de generalizar. Existen diversas razones por las cuales puede ocurrir este fenómeno y también varias estrategias para subsanarlo. Las causas más comunes para obtener *overfitting* están relacionadas con un modelo muy complejo, un dato de entrenamiento insuficiente (o inadecuado) o un entrenamiento que utilizó demasiadas iteraciones (y debió cortarse en alguna iteración previa).

En principio, si nuestro modelo es demasiado complejo (muy profundo o muy ancho, por ejemplo) puede ocurrir que este sea también demasiado "elástico". En esta situación, el modelo utilizó toda su capacidad para aprender (o memorizar) hasta los patrones más sutiles del dato de entrenamiento en particular, y a la hora de ser evaluado en el dato de testeo, su desempeño no fue adecuado. Toda su elasticidad fue destinada a aprender estos patrones y se volvió rígido para desempeñarse correctamente con un dato nuevo. Aun siendo un caso diferente, puede pensarse en un problema de interpolación sencillo donde se necesitan interpolar unos pocos puntos. Si para este problema se plantea un polinomio de grado muy alto, se conseguirá un modelo que pase por todos nuestros puntos (tal como se le requiere a una interpolación), pero que probablemente no represente el comportamiento general de las mediciones. En otras palabras, el modelo será muy "complejo". Si el *overfitting* que observamos se debe a una elección de modelo demasiado compleja, las estrategias para atacar este problema serían simplificar el modelo (por ejemplo, achicarlo), agrandar el conjunto de entrenamiento, o alguna estrategia de regularización (Géron, 2019) (*damping least squares*, *dropout*, etc). Por otro lado, si el modelo es suficientemente simple, pero el dato de entrenamiento es pequeño, entonces es posible que nuestro modelo haya logrado aprender todos los patrones o reglas de este dato con facilidad ajustándose "en exceso" a este conjunto. Lógicamente, en este caso la estrategia adecuada sería aumentar el volumen de datos. Si por el contrario, el dato de entrenamiento es inadecuado (tiene otra distribución o es muy ruidoso) es lógico que ambos errores sean diferentes. En este caso tenemos que considerar la posibilidad de filtrar el dato, o directamente conseguir uno nuevo. Finalmente, puede ocurrir que la diferencia de errores sea grande aun teniendo un modelo de dimensiones razonables y un dato limpio y extenso. En este caso, pudo ocurrir que entrenamos demasiado, permitiendo que nuestro modelo sobre-ajuste al dato de entrenamiento perdiendo su elasticidad para desempeñarse en el conjunto de testeo. Una posible solución a este inconveniente es entrenar una menor cantidad de épocas, que se conoce típicamente como *early stopping*.

Por otro lado, cuando un modelo es demasiado simple para poder aprender los patrones de un dato se habla de *underfitting*. Opuesto al caso anterior, se debe buscar un modelo más complejo (más elástico), que sea capaz de memorizar más información durante el entre-

namiento. También, de haber utilizado algún método de regularización, se puede trabajar sobre el mismo para reducir su impacto. Otra posible forma de mejorar este problema suele ser entrenar una mayor cantidad de épocas o cambiar la entrada por un dato que sea más representativo del problema de estudio.