

Cálculo de los tiempos de viaje de primeros arribos partícipes en el proceso de tomografía sísmica de onda convertida PS sobre un modelo espacial usando redes neuronales

Maria Alejandra Meza Montañez, Fernando Javier Madero Jimenez y Cristhian Alberto Sandoval Leon

Trabajo de Grado para optar al título de Ingeniero Electrónico

Director

Carlos Andres Niño Niño
Magister en Ingeniería Electrónica

Codirector

César Antonio Duarte Gualdrón
PhD in Electrical and Computer Engineering

Universidad Industrial de Santander
Facultad de Ingenierías Fisicomecánicas
Escuela de Ingenierías Eléctrica, Electrónica y de Telecomunicaciones
Ingeniería Electronica
Bucaramanga
2023

Agradecimientos

Quiero expresar mi más profundo agradecimiento a mi querida madre. Su amor incondicional, aliento constante y confianza en mí han sido fundamentales en mi vida. Tu apoyo inquebrantable ha sido mi mayor fortaleza y motivación. Te agradezco de corazón por estar siempre a mi lado. Cristhian Sandoval

Deseo agradecer a mis padres, quienes siempre han estado ahí para mí. En especial, quiero destacar el apoyo emocional inestimable de mi madre. A mis hermanos y pareja, por su apoyo constante. A mis amigos, por colaborar en este proyecto, y a aquellos que, de una u otra forma, brindaron su apoyo. No puedo olvidar a mi hermosa hija Alison, quien nos motivó a concluir este proyecto. Gracias a todos por ser parte de mi vida. Fernando Madero

Agradezco a mi familia por su amor incondicional y por ser mi fuente constante de motivación, sus sacrificios y apoyo moral fueron esenciales para alcanzar este logro académico, también agradezco a mi pareja, amigos y compañeros que, de alguna manera, contribuyeron en mi formación, su colaboración y apoyo fueron fundamentales en este camino. Alejandra Meza

Índice general

Introducción	12
1 Objetivos	14
1.1 Objetivo general	14
1.2 Objetivos específicos	14
2 Marco de Referencia.	15
2.1 Contexto geofísico	15
2.1.1 <i>Metodología actual para calcular tiempos de viaje de primeros arribos</i>	17
2.1.2 <i>Tomografía de onda convertida PS</i>	19
2.2 Inteligencia Artificial.	21
2.2.1 <i>Redes neuronales</i>	22
2.2.2 <i>Entrenamiento</i>	24
3 Metodología	30
3.1 Construcción base de datos	30
3.1.1 <i>Base de datos tiempos P</i>	32
3.1.2 <i>Base de datos tiempos PP y tiempos PS</i>	33
3.2 Red neuronal	34
3.2.1 <i>Arquitectura</i>	36
3.2.2 <i>Entrenamiento</i>	38
3.3 Tiempo de procesamiento	39
4 Resultados experimentales.	42
4.1 Modelo Senoidal:	44
4.1.1 <i>Cercano a la superficie</i>	44
4.1.2 <i>Mitad del modelo espacial</i>	46
4.1.3 <i>Lejano a la superficie.</i>	48
4.2 Modelo V:	50
4.2.1 <i>Cercano a la superficie</i>	50
4.2.2 <i>Mitad del modelo espacial</i>	52
4.2.3 <i>Lejano a la superficie.</i>	54
4.3 Modelo Cóncavo:	56
4.3.1 <i>Cercano a la superficie</i>	56
4.3.2 <i>Mitad del modelo espacial</i>	58
4.3.3 <i>Lejano a la superficie.</i>	60
4.4 Modelo Cóncavo inverso:	62
4.4.1 <i>Cercano a la superficie</i>	62
4.4.2 <i>Mitad del modelo espacial</i>	64
4.4.3 <i>Lejano a la superficie.</i>	66
4.5 Análisis:	68
5 Discusión	73
6 Conclusiones.	76
7 Recomendaciones.	79
Referencias Bibliográficas.	80

Apendices 84

Índice de figuras

2.1. Correspondencia entre las ondas adquiridas por los sensores multicomponentes en la superficie Γ con la conversión PS y la reflexión PP que ocurren en el reflector geológico. (a) Tiempos de viaje de las primeras llegadas de las ondas PS (en rojo) y PP (en azul). (b) Rayos producidos por la activación del disparo en la estrella roja, en líneas punteadas negras las ondas de transmisión P, en azul las ondas P reflejadas y en rojo los rayos PS convertidos. Códigos de color: Rojo: conversiones PS. Azul: reflexiones PP. Fuente: Tomado de (C.A.Niño et al., 2022).	18
2.2. Representación del modelo espacial de tomografía de onda convertida PS y reflexión PP utilizando la metodología Level Set que mediante el nivel $\phi(x, z) = 0$ modela una interfaz R que divide el espacio Ω en dos subespacios Ω_1 y Ω_2 con propiedades físicas diferentes de lentitud, tanto para ondas cortantes S_s como para compresionales S_p Fuente: Tomado de (C.A.Niño et al., 2022).	19
2.3. Esquema de la metodología de tomografía sísmica PS regido por la función Level set (a) Modelado de rayos sobre level set y $\phi(x, z) = 0$ como líneas punteadas, en púrpura para la ondas PS y en verde para PP que deberían converger en el verdadero sistema (físico), rojo para PS y azul para PP que corresponden a la parte inferior de la figura. Tomado de : (C.A.Niño et al., 2022)	20
2.4. Modelo simple de una red neuronal artificial. Tomado de: (Mantilla Parra y Vargas Blanco, 2023)	23
3.1. Ejemplo de tiempos de viaje P, PP y PS en el modelo espacial; en un modelo senoidal con velocidad 0.5 [km/s]	31
3.2. Descripción general funcionamiento EIKO4P-S. (1) Se introducen los datos a ser predichos en el espacio de ω_1 , primero alimentan a la red P que posteriormente estima el vector de tiempos P, de allí se extraen los tiempos P sobre la posición del reflector y se concatenan a la matriz de datos para formar la entrada de las redes PP y PS que estiman los vectores de tiempos correspondientes, (2) explica la reconstrucción de la imagen a partir de los datos predichos.	35
3.3. Arquitectura red neuronal propuesta	36
3.4. Comparación rendimiento para 3 capas de normalización.	37
4.1. Geometrías usadas en el apartado de resultados	42
4.2. Mapa de contorno de modelo senoidal con velocidad de 1km/s, columnas de tiempos P, PP y PS respectivamente, filas 1,2,3 posición del Shot ₁ [50 ,40] m, Shot ₂₅ [50 ,1000] m, Shot ₄₉ [50 ,1960] m, respectivamente	44
4.3. Comparativa de tiempos de viaje sobre el reflector Γ entre Fast marching y EIKO-4PS , columnas 1,2 tiempos PP y PS respectivamente, filas 1,2,3 posición del Shot ₁ [50 ,40] m, Shot ₂₅ [50 ,1000] m, Shot ₄₉ [50 ,1960] mm, respectivamente	45

- 4.4. Errores absolutos del modelo senoidal con velocidad de 1km/s, columnas de tiempos P, PP y PS respectivamente, filas 1,2,3 posición del Shot₁ [50 ,40] m, Shot₂₄ [50 ,1000] m, Shot₄₉ [50 ,1960] m, respectivamente 45
- 4.5. Mapa de contorno de modelo senoidal seccion media con velocidad 1km/s, columnas de tiempos P, PP y PS respectivamente, filas 1,2,3 posición del Shot₁ [50 ,40] m, Shot₂₅ [50 ,1000] m, Shot₄₉ [50 ,1960] m, respectivamente 46
- 4.6. Comparativa de tiempos de viaje sobre el reflector Γ entre Fast marching y EIKO-4PS , columnas 1,2 tiempos PP y PS respectivamente,filas 1,2,3 posición del Shot₁ [50 ,40] m, Shot₂₅ [50 ,1000] m, Shot₄₉ [50 ,1960] m, respectivamente 47
- 4.7. Errores absolutos del modelo senoidal con velocidad de 1km/s, columnas de tiempos P, PP y PS respectivamente, filas 1,2,3 posición del Shot₁ [50 ,40] m, Shot₂₅ [50 ,1000] m, Shot₄₉ [50 ,1960] m, respectivamente 47
- 4.8. Mapa de contorno de modelo senoidal de mayor profundidad con velocidad 1km/s, columnas de tiempos P, PP y PS respectivamente, filas 1,2,3 posición del Shot₁ [50 ,40] m, Shot₂₅ [50 ,1000] m, Shot₄₉ [50 ,1960] m, respectivamente 48
- 4.9. Comparativa de tiempos de viaje sobre el reflector Γ entre Fast marching y EIKO-4PS , columnas 1,2 tiempos PP y PS respectivamente,filas 1,2,3 posición del Shot₁ [50 ,40] m, Shot₂₅ [50 ,1000] m, Shot₄₉ [50 ,1960] m, respectivamente 49
- 4.10. Errores absolutos del modelo senoidal con velocidad de 1km/s, columnas de tiempos P, PP y PS respectivamente, filas 1,2,3 posición del Shot₁ [50 ,40] m, Shot₂₅ [50 ,1000] m, Shot₄₉ [50 ,1960] m, respectivamente 49
- 4.11. Mapa de contorno de modelo V con velocidad 5 km/s, columnas de tiempos P, PP y PS respectivamente, filas 1,2,3 posición del Shot₁ [50 ,40] m, Shot₂₅ [50 ,1000] m, Shot₄₉ [50 ,1960] m, respectivamente 50
- 4.12. Comparativa de tiempos de viaje sobre el reflector Γ entre Fast marching y EIKO-4PS , columnas 1,2 tiempos PP y PS respectivamente,filas 1,2,3 posición del Shot₁ [50 ,40] m, Shot₂₅ [50 ,1000] m, Shot₄₉ [50 ,1960] m, respectivamente 51
- 4.13. Errores absolutos del modelo V con velocidad de 5 km/s, columnas de tiempos P, PP y PS respectivamente, filas 1,2,3 posición del Shot₁ [50 ,40] m, Shot₂₅ [50 ,1000] m, Shot₄₉ [50 ,1960] m, respectivamente 51
- 4.14. Mapa de contorno de modelo V sección media con velocidad 5km/s, columnas de tiempos P, PP y PS respectivamente, filas 1,2,3 posición del Shot₁ [50 ,40] m, Shot₂₅ [50 ,1000] m, Shot₄₉ [50 ,1960] m, respectivamente 52
- 4.15. Comparativa de tiempos de viaje sobre el reflector Γ entre Fast marching y EIKO-4PS , columnas 1,2 tiempos PP y PS respectivamente,filas 1,2,3 posición del Shot₁ [50 ,40] m, Shot₂₅ [50 ,1000] m, Shot₄₉ [50 ,1960] m, respectivamente 53
- 4.16. Errores absolutos del modelo V con velocidad de 5km/s, columnas de tiempos P, PP y PS respectivamente, filas 1,2,3 posición del Shot₁ [50 ,40] m, Shot₂₅ [50 ,1000] m, Shot₄₉ [50 ,1960] m, respectivamente 53
- 4.17. Mapa de contorno de modelo V de mayor profundidad con velocidad 5km/s, columnas de tiempos P, PP y PS respectivamente, filas 1,2,3 posición del Shot₁ [50 ,40] m, Shot₂₅ [50 ,1000] m, Shot₄₉ [50 ,1960] m, respectivamente . . . 54
- 4.18. Comparativa de tiempos de viaje sobre el reflector Γ entre Fast marching y EIKO-4PS , columnas 1,2 tiempos PP y PS respectivamente,filas 1,2,3 posición del Shot₁ [50 ,40] m, Shot₂₅ [50 ,1000] m, Shot₄₉ [50 ,1960] m, respectivamente 55

4.19. Errores absolutos del modelo V con velocidad de 5 km/s, columnas de tiempos P, PP y PS respectivamente, filas 1,2,3 posición del Shot ₁ [50 ,40] m, Shot ₂₅ [50 ,1000] m, Shot ₄₉ [50 ,1960] m, respectivamente	55
4.20. Mapa de contorno de modelo concavo con velocidad 3.75 km/s, columnas de tiempos P, PP y PS respectivamente, filas 1,2,3 posición del Shot ₁ [50 ,40] m, Shot ₂₅ [50 ,1000] m, Shot ₄₉ [50 ,1960] m, respectivamente	56
4.21. Comparativa de tiempos de viaje sobre el reflector Γ entre Fast marching y EIKO-4PS , columnas 1,2 tiempos PP y PS respectivamente,filas 1,2,3 posición del Shot ₁ [50 ,40] m, Shot ₂₅ [50 ,1000] m, Shot ₄₉ [50 ,1960] m, respectivamente	57
4.22. Errores absolutos del modelo cóncavo con velocidad de 3.75 km/s, columnas de tiempos P, PP y PS respectivamente, filas 1,2,3 posición del Shot ₁ [50 ,40] m, Shot ₂₅ [50 ,1000] m, Shot ₄₉ [50 ,1960] m, respectivamente	57
4.23. Mapa de contorno de modelo cóncavo sección media con velocidad 3.75 km/s, columnas de tiempos P, PP y PS respectivamente, filas 1,2,3 posición del Shot ₁ [50 ,40] m, Shot ₂₅ [50 ,1000] m, Shot ₄₉ [50 ,1960] m, respectivamente	58
4.24. Comparativa de tiempos de viaje sobre el reflector Γ entre Fast marching y EIKO-4PS , columnas 1,2 tiempos PP y PS respectivamente,filas 1,2,3 posición del Shot ₁ [50 ,40] m, Shot ₂₅ [50 ,1000] m, Shot ₄₉ [50 ,1960] m, respectivamente	59
4.25. Errores absolutos del modelo cóncavo inverso con velocidad de 2.5 km/s, columnas de tiempos P, PP y PS respectivamente, filas 1,2,3 posición del Shot ₁ [50 ,40] m, Shot ₂₅ [50 ,1000] m, Shot ₄₉ [50 ,1960] m, respectivamente	59
4.26. Mapa de contorno de modelo cóncavo de mayor profundidad con velocidad 3.75 km/s, columnas de tiempos P, PP y PS respectivamente, filas 1,2,3 posición del Shot ₁ [50 ,40] m, Shot ₂₅ [50 ,1000] m, Shot ₄₉ [50 ,1960] m, respectivamente	60
4.27. Comparativa de tiempos de viaje sobre el reflector Γ entre Fast marching y EIKO-4PS , columnas 1,2 tiempos PP y PS respectivamente,filas 1,2,3 posición del Shot ₁ [50 ,40] m, Shot ₂₅ [50 ,1000] m, Shot ₄₉ [50 ,1960] m, respectivamente	61
4.28. Errores absolutos del modelo cóncavo con velocidad de 3.75 km/s, columnas de tiempos P, PP y PS respectivamente, filas 1,2,3 posición del Shot ₁ [50 ,40] m, Shot ₂₅ [50 ,1000] m, Shot ₄₉ [50 ,1960] m, respectivamente	61
4.29. Mapa de contorno de modelo cóncavo inverso con velocidad 2.5 km/s, columnas de tiempos P, PP y PS respectivamente, filas 1,2,3 posición del Shot ₁ [50 ,40] m, Shot ₂₅ [50 ,1000] m, Shot ₄₉ [50 ,1960] m, respectivamente	62
4.30. Comparativa de tiempos de viaje sobre el reflector Γ entre Fast marching y EIKO-4PS , columnas 1,2 tiempos PP y PS respectivamente, filas 1,2,3 posición del Shot ₁ [50 ,40] m, Shot ₂₅ [50 ,1000] m, Shot ₄₉ [50 ,1960] m, respectivamente	63
4.31. Errores absolutos del modelo cóncavo inverso con velocidad de 2.5 km/s, columnas de tiempos P, PP y PS respectivamente, filas 1,2,3 posición del Shot ₁ [50 ,40] m, Shot ₂₅ [50 ,1000] m, Shot ₄₉ [50 ,1960] m, respectivamente	63
4.32. Mapa de contorno de modelo cóncavo inverso sección media con velocidad 2.5 km/s, columnas de tiempos P, PP y PS respectivamente, filas 1,2,3 posición del Shot ₁ [50 ,40] m, Shot ₂₅ [50 ,1000] m, Shot ₄₉ [50 ,1960] m, respectivamente	64

4.33. Comparativa de tiempos de viaje sobre el reflector Γ entre Fast marching y EIKO-4PS , columnas 1,2 tiempos PP y PS respectivamente, filas 1,2,3 posición del Shot ₁ [50 ,40] m, Shot ₂₅ [50 ,1000] m, Shot ₄₉ [50 ,1960] m, respectivamente	65
4.34. Errores absolutos del modelo cóncavo inverso con velocidad de 2.5 km/s, columnas de tiempos P, PP y PS respectivamente, filas 1,2,3 posición del Shot ₁ [50 ,40] m, Shot ₂₅ [50 ,1000] m, Shot ₄₉ [50 ,1960] m, respectivamente	65
4.35. Mapa de contorno de modelo cóncavo inverso de mayor profundidad con velocidad 2.5 km/s, columnas de tiempos P, PP y PS respectivamente, filas 1,2,3 posición del Shot ₁ [50 ,40] m, Shot ₂₅ [50 ,1000] m, Shot ₄₉ [50 ,1960] m, respectivamente	66
4.36. Comparativa de tiempos de viaje sobre el reflector Γ entre Fast marching y EIKO-4PS , columnas 1,2 tiempos PP y PS respectivamente, filas 1,2,3 posición del Shot ₁ [50 ,40] m, Shot ₂₅ [50 ,1000] m, Shot ₄₉ [50 ,1960] m, respectivamente	67
4.37. Errores absolutos del modelo cóncavo inverso con velocidad de 2.5 km/s, columnas de tiempos P, PP y PS respectivamente, filas 1,2,3 posición del Shot ₁ [50 ,40] m, Shot ₂₅ [50 ,1000] m, Shot ₄₉ [50 ,1960] m, respectivamente	67
4.38. Resumen de resultados error absoluto para geometría Senoidal.	70
4.39. Resumen de resultados error absoluto para geometría V.	71
4.40. Resumen de resultados error absoluto para geometría Cóncavo.	71
4.41. Resumen de resultados error absoluto para geometría Cóncavo Inverso.	72

Índice de tablas

2.1. Clases de redes neuronales (Hassoun, 1995)	23
2.2. Definición de conjuntos de prueba, entrenamiento y validación (Genç y Tunc, 2019).	24
2.3. Hiperparámetros de ajuste(Hassoun, 1995)	25
2.4. Funciones de activación más comunes (Rodríguez P. y Buitrago R., 2022) . .	26
2.5. Funciones de costo (Janiesch et al., 2021)	27
2.6. Algoritmos de optimización más comunes.	27
2.7. Terminología encargada de evaluar el desempeño de una red neuronal profunda (Bashir et al., 2020).	28
3.1. Recopilación de datos para la base de datos P	32
3.2. Composición de la base de datos para los tiempos P	33
3.3. Recopilación de datos para la base de datos PP y PS	33
3.4. Composición de la base de datos para los tiempos PP y PS	34
3.5. Resumen parámetros de entrenamiento para las redes de EIKO4P-S.	39
3.6. Comparación tiempos de procesamiento teniendo los parámetros de reflector en el calculo de tiempos P, PP y PS entre Fast Marching y EIKO4P-S. . . .	40
4.1. Organización de mapas de contorno de los modelos con geometrías de forma V, senoidal, cóncavo y cóncavo inverso	43

Resumen

Título: Cálculo de los tiempos de viaje de primeros arribos partícipes en el proceso de tomografía sísmica de onda convertida ps sobre un modelo espacial usando redes neuronales*

Autor: Maria Alejandra Meza Montañez, Fernando Javier Madero Jimenez, Cristhian Alberto Sandoval Leon**

Palabras clave: Ecuación Eikonal, reflector, tiempos de ondas PP-PS, inteligencia artificial, redes neuronales

Descripción: en el proceso de tomografía sísmica, aprovechamos el reciente auge en el campo de la inteligencia artificial y sus capacidades para resolver problemas complejos utilizando derivadas parciales, especialmente en el contexto geofísico para simulaciones basadas en física. Proponemos Eiko 4P-S, un enfoque que permite calcular los tiempos de viaje de los primeros arribos de las ondas P, PP y PS sobre modelos espaciales con diferentes formas de reflector y estructuras de velocidades homogéneas. Para lograrlo, se interconectaron tres redes neuronales, que calcula los tiempos desde una fuente de onda P, hacia cualquier punto sobre o por encima del reflector, posteriormente de un postprocesamiento de los datos predichos se reconstruyó la matriz de tamaño 79×79 píxeles que representa un espacio de 2 km de ancho por 2 km de profundidad. Se aplicó un procedimiento similar para las redes de tiempos PP y PS, pero esta vez, la fuente está representada como los tiempos P en la posición del reflector y los receptores son todos los puntos sobre el reflector hasta la superficie gamma (Γ).

Las solución propuesta para estimar los tiempos de viaje violan los principios de la ecuación diferencial (J. D. Smith et al., 2021). Cuando una ecuación diferencial es resuelta, se buscan soluciones que satisfagan exactamente la ecuación. Sin embargo, en el aprendizaje profundo, se permite que las soluciones propuestas no cumplan exactamente con la ecuación, convirtiéndolo en un problema de optimización con el objetivo de encontrar parámetros que minimicen el grado en el que esta ecuación se ve afectada. Se aprovechó la capacidad de diferenciación de la red para encontrar gradientes, los cuales son utilizados para ajustar los pesos y sesgos con el fin de minimizar la función de pérdida, evitando el uso de métodos de diferenciación finita.

Tanto el entrenamiento como la inferencia se llevaron a cabo mediante cómputo altamente paralelizable, utilizando GPU. Además, se sometió Eiko 4P-S a rigurosas pruebas variando los modelos de velocidad y reflector, demostrando su versatilidad y robustez.

Eiko 4P-S elimina la necesidad de crear modelos de reflector utilizando el enfoque de level set con $\phi = 0$ después del entrenamiento. Esto conlleva una notable reducción en el tiempo total para calcular los tiempos, al evitar el ajuste de contorno y su correspondiente tiempo de procesamiento. No obstante, en términos de procesamiento, una vez que el modelo analítico tiene los parámetros geométricos necesarios para calcular los tiempos, se ha observado que la solución analítica (C.A.Niño et al., 2022) supera a Eiko 4P-S en tiempos de cálculo durante esta etapa, siendo de 3 a 6 veces más rápida. Este enfoque representa un avance significativo en el problema directo de la tomografía sísmica, abriendo nuevas posibilidades para su implementación, mejora y abordaje. Se propone que el algoritmo podría mejorar aún más mediante la adopción de un enfoque de muestreo en espacios de igual dimensión pero con mayor resolución en las distancias entre píxeles.

*Trabajo de Grado

**Facultad de Ingeniería Fisicomecánicas. Escuela de Ingenierías Eléctrica, Electrónica y Telecomunicaciones. Director: Carlos Andres Niño Niño. Codirector: César Antonio Duarte Gualdrón

Abstract

Title: Calculation of the travel times of first arrivals participating in the PS converted wave seismic tomography process on a space model using neural networks^{***}

Author: Maria Alejandra Meza Montañez, Fernando Javier Madero Jimenez, Cristhian Alberto Sandoval Leon^{****}

Key Words: Eikonal equation, reflector, PP-PS wave times, artificial intelligence, neural networks.

Description: in the process of seismic tomography, we leverage the recent boom in the field of artificial intelligence and its capabilities to solve complex problems using partial derivatives, especially in the geophysical context for physics-based simulations. We propose Eiko 4P-S, an approach that allows calculating the travel times of the first arrivals of P, PP, and PS waves over spatial models with different reflector shapes and homogeneous velocity structures. To achieve this, three neural networks were interconnected, calculating the times from a P-wave source to any point above or on the reflector. After post-processing the predicted data, the matrix of size 79×79 pixels representing a 2×2 space was reconstructed, depicting depth. A similar procedure was applied to the PP and PS time networks, but this time, the source is represented as P times at the reflector's position, and the receivers are all points on the reflector up to the gamma (Γ) surface.

The proposed solution to estimate travel times violates the principles of the differential equation (J. D. Smith et al., 2021). When a differential equation is solved, solutions that exactly satisfy the equation are sought. However, in deep learning, it is allowed for proposed solutions not to exactly fulfill the equation, turning it into an optimization problem aimed at finding parameters that minimize the degree to which this equation is affected. The network's differentiation capability was leveraged to find gradients, which are used to adjust weights and biases to minimize the loss function, avoiding the use of finite difference methods.

Both training and inference were conducted through highly parallelizable computing, utilizing GPU. Furthermore, Eiko 4P-S underwent rigorous testing by varying velocity and reflector models, demonstrating its versatility and robustness.

Eiko 4P-S eliminates the need to create reflector models using the level set approach with $\phi = 0$ after training. This results in a significant reduction in the total time for calculating travel times by avoiding contour adjustments and their corresponding processing time. However, in terms of processing, once the analytical model has the necessary geometric parameters to calculate travel times, it has been observed that the analytical solution (C.A.Niño et al., 2022) outperforms EIKO4P-S in computation times during this stage, being 3 to 6 times faster. This approach represents a significant advancement in the direct problem of seismic tomography, opening new possibilities for its implementation, improvement, and approach. It is suggested that the algorithm could be further enhanced by adopting a sampling approach in spaces of equal dimension but with higher resolution in pixel distances.

^{***} Degree work

^{****} Faculty of Physicomechanics. School of Electrical, Electronic and Telecommunications Engineering. Electronics Engineering. Director: Carlos Andres Niño Niño. Codirector: César Antonio Duarte Gualdrón.

Introducción

En la actualidad existen varias técnicas de tomografía sísmica, una de ellas es de onda convertida PS en un modelo físico espacial (modelo bidimensional)(C.A.Niño et al., 2022). En primera instancia, esta tomografía sísmica es originada por una fuente artificial de naturaleza P, la cual es transmitida hacia el primer reflector descrito por la función Level Set(Zeneli et al., 2021), generando ondas de conversión PS y reflexión PP que se propagan hasta la superficie; dicho lo anterior, la duración de este recorrido es lo que se denomina tiempos de viaje de primeros arribos que se calculan por medio de la ecuación Eikonal. Por consiguiente, estos tiempos permiten ajustar parámetros que actualizan los modelos de lentitud y posición del reflector para cada iteración; de tal manera que, en la iteración final el reflector modelado sea similar al modelo físico real del subsuelo. En definitiva, la obtención de los tiempos de viaje de primeros arribos son una parte importante para la realización de tomografía sísmica de onda convertida PS.

En la búsqueda de una forma eficiente de implementar la ecuación eikonal en el contexto de simulaciones geofísicas, se han propuesto diferentes métodos numéricos como Fast Marching Methods(FMM) y Fast Sweeping Methods (FMS), que permiten calcular distancias y tiempos de llegada de ondas en dominios discretizados (grilla en un modelo espacial), siendo Ω el dominio bidimensional donde se determina la solución (Capozzoli et al., 2013). Aportes más recientes en la resolución de la ecuación EIKONAL proponen el uso de la inteligencia artificial. Existen alternativas como PINN (Physics-Informed Neural Networks) (Bin Waheed et al., 2020)(J. Smith, 2021), que resuelve la ecuación eikonal por medio de aprendizaje no supervisado. Mientras que otro enfoque, como EIKONET(J. D. Smith et al., 2021), propone un entrenamiento de la red mediante aprendizaje supervisado. Ambas utilizan deep neural networks (DNN) para el cálculo de tiempos.

Este trabajo se enfoca en el cálculo de tiempos de viaje de primeros arribos mediante el uso de redes neuronales. En primera instancia, se generó una base de datos a partir del

algoritmo propuesto por (C.A.Niño et al., 2022), esta base de datos se construyó sobre un modelo espacial con dimensiones de 2 km de ancho por 2 km de largo, la cual se representó en una imagen de 79×79 píxeles. Cada píxel en esta imagen tiene una representación implícita que incluye información sobre el tiempo de viaje y la velocidad de una onda P, PP o PS en un punto específico de Ω_1 . Definida la base de datos, se propuso un conjunto de tres redes interconectadas entre sí llamado EIKO4P-S, la primera red se encargó de calcular los tiempos de viaje de onda convertida P, que corresponde a los tiempos desde la fuente hasta todos los puntos a lo largo del modelo de velocidad Ω_1 hasta el reflector; estos tiempos se utilizaron como entrada para las dos redes siguientes, una para calcular los tiempos PP y la otra para los tiempos PS, dichas redes calculan los tiempos de viaje desde el reflector hasta la superficie Γ , de tal manera que, con este modelo propuesto se logró calcular todo el recorrido de los tiempos de viaje de primeros arribos, a través de un postprocesamiento que permite reconstruir la imagen a partir de las predicciones de cada uno de los tiempos, todos estos datos fueron entrenados aprovechando la capacidad de cálculo paralelizable que ofrecen las GPU, por lo cual EIKO4P-S se realizó en el entorno de trabajo gratuito de Google Colab, que también delimitó la capacidad de datos posibles para el entrenamiento, los resultados en el cálculo de los tiempos de viaje de primeros arribos en este trabajo revelan la capacidad de generalización de la red para ajustarse a las diferentes formas de reflector así como también a los diferentes modelos de velocidad y sus diferentes shots, lo que permitió identificar posibles mejoras en el entrenamiento y en los resultados. Además nos dejó ver las ventajas y desventajas con respecto al modelo analítico de Fast Marching propuesto en (C.A.Niño et al., 2022).

1. Objetivos

1.1. Objetivo general

Calcular los tiempos de viaje de primeros arribos sobre un modelo espacial que incluye el primer reflector descrito con la función Level Set, en el proceso de tomografía sísmica de onda convertida PS mediante la implementación de un sistema basado en redes neuronales.

1.2. Objetivos específicos

- Crear un conjunto de datos de tiempos de viaje de primeros arribos de ondas convertidas y reflejadas PS-PP, a partir del uso de la ecuación Eikonal en diferentes modelos de lentitudes, localización y forma del reflector para entrenar un sistema de redes neuronales.
- Implementar un sistema basado en redes neuronales a partir de los conjuntos de datos previamente construidos que calcule los tiempos de viaje de primeros arribos como una alternativa al uso de la ecuación Eikonal.
- Comparar los resultados obtenidos por medio del uso de la ecuación Eikonal con los resultados conseguidos mediante el sistema desarrollado de redes neuronales.

2. Marco de Referencia

En la siguiente investigación se consideran una serie de aportes teóricos importantes con el propósito de validar la red EIKO4P-S, como una nueva alternativa en la solución de los tiempos de viaje de primeros arribos de ondas convertidas *PS* y ondas reflejadas *PP* en el contexto de la sismología, mediante técnicas de inteligencia artificial.

2.1. Contexto geofísico

La geofísica es una de las ramas principales de la ciencia que se dedica al estudio de la estructura física de la Tierra, en esta se aplican diferentes técnicas y métodos de investigación para el conocimiento de las propiedades físicas del subsuelo, con el fin de comprender aspectos como la composición interna del planeta, determinar la ubicación adecuada de obras civiles, prevenir desastres naturales, recolectar información de eventos sismológicos y optimizar el proceso de exploración y extracción de minerales, agua y energía (Reynolds, 2011)

Los eventos sismológicos son movimientos inesperados al interior de la tierra, debido a la propagación de los diferentes tipos de ondas sísmicas a través de la corteza terrestre, estas ondas son perturbaciones y se propagan con una rapidez determinada por las propiedades físicas de dicho medio (Mescua et al., 2021). Los tipos de ondas sísmicas son catalogadas como la onda primaria P que es una componente longitudinal o compresional lo que significa que el medio se expande y se contrae en la dirección de la propagación y la onda secundaria S que es una onda transversal o de corte lo que significa que el medio oscila perpendicularmente a la dirección del movimiento; A partir de estas ondas, surgen variantes como las ondas PP que son ondas tipo P y ondas PS que son ondas tipo P que llegan a una discontinuidad interna de la tierra y parte de su energía se convierte en ondas tipo S.

Existen técnicas que construyen modelos tridimensionales para representar la estructura interna de la tierra, a partir de ondas propagadas por fuentes artificiales, dichos modelos suelen presentarse en forma de imágenes tomográficas por lo que recibe el nombre de tomografía sísmica (Sepúlveda-Jaimes y Cabrera-Zambrano, 2018), ya que se observan las diferentes capas y discontinuidades subsuperficiales, de igual manera permiten estimar las velocidades y tiempos de viaje de las ondas P, PP y PS que llegan a las estaciones de monitoreo en la superficie. Estos datos almacenados en las estaciones sismográficas junto con los tiempos de primeros arribos proporcionan información sobre la velocidad de las ondas sísmicas en diferentes direcciones y profundidades, lo que permite inferir la composición y la distribución de las capas geológicas.

Los tiempos de primeros arribos en la tomografía sísmica hace referencia a los momentos en que las ondas sísmicas llegan por primera vez a los receptores o estaciones de monitoreo ubicadas en la superficie de la tierra (Zuluaga Herrera y Muñoz Cuartas, 2020); en particular, el tiempo de viaje de primeros arribos para las ondas PP comprende el recorrido de ondas de naturaleza tipo P que se propagan desde una fuente superficial hasta los receptores y el tiempo de viaje de primeros arribos para las ondas PS, es el recorrido de ondas tipos P que se propagan de la fuente superficial hasta una discontinuidad interna de la tierra y de ahí hasta las estaciones de monitoreo.

Ahora bien, la manera en la que se registran y almacenan los datos sísmicos en los receptores se puede realizar en trazas sismológicas, cada traza corresponde a una estación de monitoreo individual, por lo tanto, recopila información en puntos específicos de la superficie terrestre (Mantilla Parra y Vargas Blanco, 2023). La adquisición de estos datos sísmicos de manera digital permite crear mediante técnicas matemáticas y computacionales una representación tridimensional de la estructura de la tierra, esta representación recibe el nombre de modelo espacial (C.A.Niño et al., 2022).

Una herramienta matemática utilizada para modelar la propagación de las ondas sísmicas y calcular los tiempos de viaje de llegada de estas incluyendo las PS, es la ecuación Eikonal (2.1),

$$|\nabla T|^2 = \frac{1}{V^2} \quad (2.1)$$

Donde:

$|\nabla T|$ es el módulo del gradiente del tiempo de viaje T ,

V es la velocidad de fase de la onda sísmica en el medio.

En resumen, la ecuación eikonal describe cómo los frentes de onda avanzan a través del medio, siguiendo las trayectorias más cortas en términos de tiempo.

2.1.1. Metodología actual para calcular tiempos de viaje de primeros arribos

El desarrollo de técnicas de tomografía sísmica confiables y que simultáneamente estima la velocidad de onda compresional en un campo P : $V_p(x, z)$ y la onda cortante en el campo S : $V_s(x, z)$ es un gran reto el procesamiento de la onda convertida (C.A.Niño et al., 2022).

El modelo analítico que utiliza fast marching representa un modelo espacial de tomografía sísmica de onda convertida PS y reflejada PP, el cual adopta la función level set como modelo de geometría de la interfaz donde se encuentran los puntos comunes y se producen las ondas convertidas PS y reflejadas PP, además para incluir una restricción de profundidad para un campo P utilizando $\phi(x, z) = 0$ como reflector R que desplaza las reflexiones observadas en los datos sísmicos a ubicaciones correctas en el espacio de profundidad (x, z) , el cual mapea una interfaz que define con precisión un cambio de impedancia eléctrica, ade-

más, divide el espacio Ω en dos subespacios, Ω_1 y Ω_2 cada uno con diferentes propiedades físicas, para las ondas de corte una lentitud S_s , mientras que para las ondas de compresion una lentitud S_p como muestra la figura (2.2) se observa el vector normal que sirve como referencia para el cálculo del ángulo de incidencia y reflexión de onda P y la conversión PS del modelo del reflector (C.A.Niño et al., 2022).

La metodología de fast-marching propuesta por (Rawlinson y Sambridge, 2005), para calcular los tiempos modelados PS Y PP resuelven la ecuación eikonal con el modelo geológico del reflector $\phi(x, z) = 0$ utilizando level set

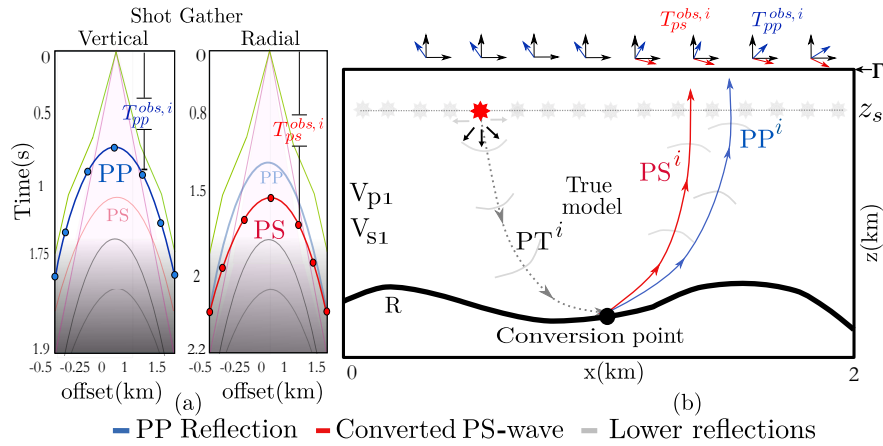


Figura 2.1: Correspondencia entre las ondas adquiridas por los sensores multicomponentes en la superficie Γ con la conversión PS y la reflexión PP que ocurren en el reflector geológico. (a) Tiempos de viaje de las primeras llegadas de las ondas PS (en rojo) y PP (en azul). (b) Rayos producidos por la activación del disparo en la estrella roja, en líneas punteadas negras las ondas de transmisión P, en azul las ondas P reflejadas y en rojo los rayos PS convertidos. Códigos de color: Rojo: conversiones PS. Azul: reflexiones PP. Fuente: Tomado de (C.A.Niño et al., 2022).

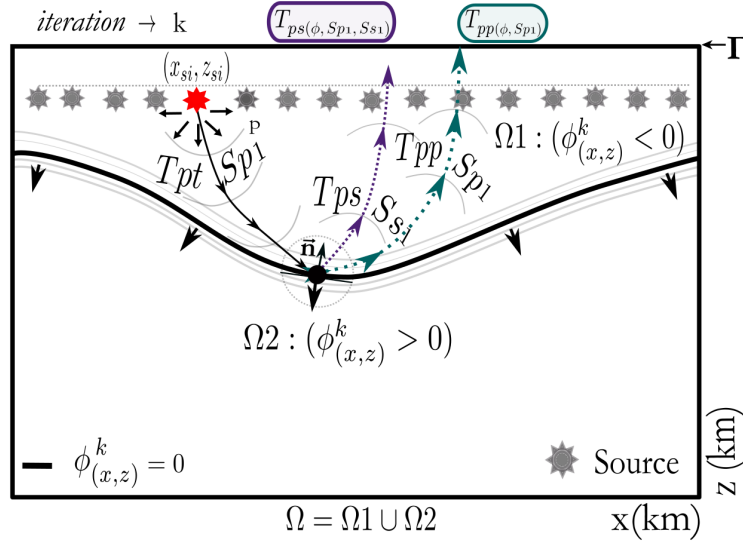


Figura 2.2: Representación del modelo espacial de tomografía de onda convertida PS y reflexión PP utilizando la metodología Level Set que mediante el nivel $\phi(x, z) = 0$ modela una interfaz R que divide el espacio Ω en dos subespacios $\Omega 1$ y $\Omega 2$ con propiedades físicas diferentes de lentitud, tanto para ondas cortantes S_s como para compresionales S_p . Fuente: Tomado de (C.A.Niño et al., 2022).

2.1.2. Tomografía de onda convertida PS

La tomografía de onda PS se obtiene a partir de la adquisición de datos sísmicos de multicomponente permitiendo distinguir las ondas PS que generalmente tienen mayor parte de la energía en la componente radial, como lo muestra en la figura (2.3), la onda fuente-reflector P que viaja con un tiempo T_{pt} se representa con un rayo que se transmite de la fuente hasta el punto de conversión-reflexión en el reflector geológico, que a su vez genera un rayo PP y PS que viaja hacia los receptores de la superficie Γ , mostrando el comportamiento físico de las ondas PP y PS .

En la metodología propone que la función level set se define como una función de distancias positivas o negativas dependiendo de la región que defina el reflector $\phi(x, z) = 0$ en donde la energía se divide en ondas PS y PP siguiendo la ley de Snell (ver 2.2), que divide el espacio Ω en dos subespacios, $\Omega 1$ ($\phi(x, z) < 0$) y $\Omega 2$ ($\phi(x, z) > 0$) para el caso de $\Omega 1$ ($\phi(x, z) < 0$) velocidades V_{p1} y V_{s1} , que matemáticamente se realiza con la función escalón Heaviside $H(\phi(x, z))$, (Prieto y Dorn, 2016), (Zheglova y Farquharson, 2012), (Li et al., 2014)

y (Aghasi et al., 2011).

$$V_{(x,z)} = V_{1(x,z)}(1 - H(\phi)), (x, z) \in \Omega_1 \quad (2.2)$$

Lo que permite realizar el proceso de tomografía sísmica estimando las velocidades por subregiones, la ecuación (2.3) que (C.A.Niño et al., 2022) utiliza en su algoritmo solo pertenecen a la parte superior Ω_1 ya que los procesos de reflexión y conversión ocurren en la subregion superior .

En el caso de las ondas PP se determinan numéricamente resolviendo la ecuación eikonal (2.1) en cada $shot_i | i = 1, 2, \dots, N.$

$$V_{(x,z)} = V_{1(x,z)}(1 - H(\phi)), (x, z) \in \Omega_1 \quad (2.3)$$

Por último, se propone el ajuste de contorno de nivel de $\phi^k(x, z) = 0 | k = 1, 2, \dots, end,$ donde tiende a converger en el reflector geológico(real) R, en donde también se ajusta V_p^k y V_s^k , dicha convergencia esta restringida por la forma del reflector matemáticamente complejo de calcular, pero se aproxima bien para formas poco comunes como lo muestra la figura (2.3).

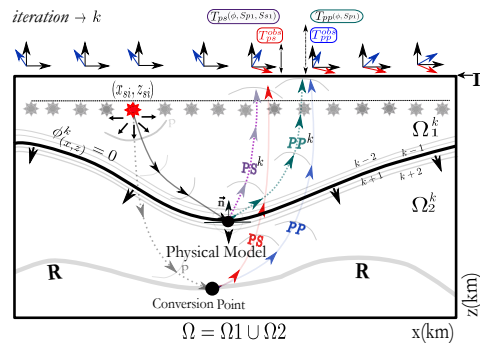


Figura 2.3: Esquema de la metodología de tomografía sísmica PS regido por la función Level set (a) Modelado de rayos sobre level set y $\phi(x, z) = 0$ como líneas punteadas, en púrpura para la ondas PS y en verde para PP que deberían converger en el verdadero sistema (físico), rojo para PS y azul para PP que corresponden a la parte inferior de la figura. Tomado de : (C.A.Niño et al., 2022)

En la figura (2.3) se ilustra la secuencia de activación de N fuentes de onda P ($Shot_i$),

with $i = 1, 2, \dots, N$) que se encuentra a una profundidad (z_s) de la superficie Γ (espacio libre) y para cada $Shot_i$ se calculan numéricamente los tiempos $T_{ps}^{k,i}(x, z)$ y $T_{pp}^{k,i}(x, z)$ del cual un tiempo residual con respecto a los tiempos $T_{ps}^{obs,i}(x, z)$ y $T_{pp}^{obs,i}(x, z)$ obteniendo la actualización de los gradientes $S_{s1}(x, z), S_{p1}(x, z)$ y $\phi(x, z = 0)$ utilizando el operador adjunto designado a cada $Shot_i$.

2.2. Inteligencia Artificial

En cuanto a la inteligencia artificial (IA) se refiere a la creación y aplicación de algoritmos y sistemas computacionales que imitan la inteligencia humana para realizar tareas que normalmente requieren habilidades cognitivas, como el aprendizaje, el razonamiento, la resolución de problemas y la percepción (Bellman, 1978). Estos sistemas pueden aprender patrones a partir de datos, adaptarse a nuevas situaciones y mejorar su rendimiento con el tiempo. Para la resolución de problemas de IA más complejos, fluctuantes y con una gran cantidad de datos, surgieron términos como el machine learning (ML) y el Deep learning (DL) (Janiesch et al., 2021).

El aprendizaje automático (Machine Learning) es una técnica dentro de la IA y, consiste en darle a un sistema o computador la habilidad de aprender características de un conjunto de datos, sin la necesidad de generar una lista de reglas explícitas. Se busca mejorar el rendimiento de una tarea a partir de la experiencia al desarrollar esta tarea (Geron, 2017). Existen tres clases de ML, el aprendizaje supervisado, no supervisado y por refuerzo.

En el aprendizaje supervisado se dispone de un conjunto de datos y se sabe para esos datos cuál es la respuesta correcta, en este tipo de aprendizaje se distinguen dos tipos de tareas: clasificación y regresión. El objetivo de la clasificación se centra en predecir resultados que son discretos, por ejemplo: el estado del clima, determinar si es frío o caliente. Por otra parte, la regresión es predecir un resultado que varía en un rango continuo a partir de la información proporcionada por una serie de variables predictoras (Rodríguez-Tapia y Camacho-Cañamón, 2018), por ejemplo: el valor de la temperatura en un día soleado. Entre

los algoritmos de aprendizaje supervisados más usados se encuentra la regresión lineal, la regresión logística, los árboles de decisión, las redes neuronales y las máquinas de soporte vectorial.

Así pues, en el aprendizaje no supervisado, se busca identificar estructuras en los datos, puesto que no se tiene la respuesta conocida en cada caso, por lo que es necesario que el algoritmo busque las relaciones entre las variables involucradas (Quituisaca-Samaniego y Álvarez, 2017), por ejemplo: dividir documentos en grupos basados en similitudes en su contenido, sin que se le diga específicamente qué temas o categorías. Los tipos de algoritmos de este aprendizaje son por detección de anomalías, clustering y asociación de reglas.

En cuanto al aprendizaje por refuerzo, es una técnica que busca que agentes inteligentes tomen acciones en un entorno determinado, y así la máquina aprenda a resolver el problema mediante prueba y error, de tal manera que, se le proporcione una recompensa al agente en función de la calidad del movimiento realizado (Mnih et al., 2013). Por ejemplo: El juego de ajedrez. Entre los algoritmos más usados están el Q-Learning y por fuerza bruta.

En efecto, es crucial destacar que este proyecto de grado se centra en un problema de regresión, abordado a través de algoritmos de aprendizaje supervisado, en particular, mediante el uso de redes neuronales.

2.2.1. *Redes neuronales*

El funcionamiento de las redes neuronales artificiales (o Artificial Neuronal Networks, ANN por sus siglas en inglés) consiste en una abstracción simple del sistema nervioso central, donde las unidades más básicas de su funcionamiento son denominadas neuronas y son organizadas en distintas capas. Una red neuronal suele tener principalmente tres partes: una capa de entrada, encargada de recibir y presentar los datos; una o múltiples capas ocultas, en su mayoría encargadas de la extracción y detección de características; y una capa de salida encargada de presentar el resultado obtenido por esta red (Mantilla Parra y Vargas Blanco, 2023).

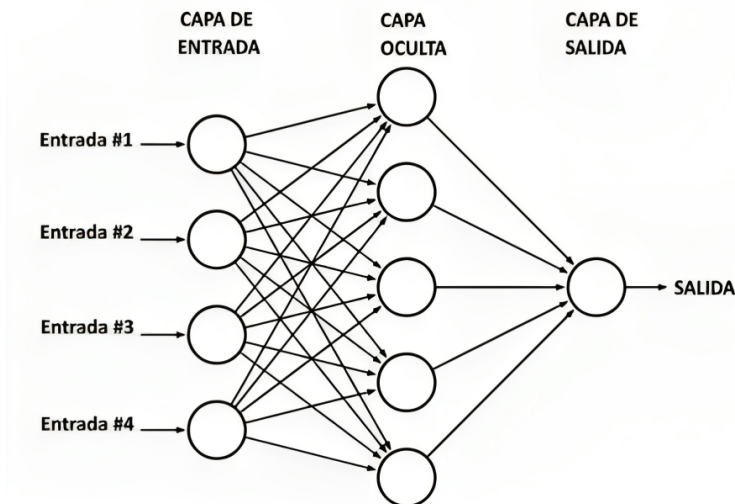


Figura 2.4: Modelo simple de una red neuronal artificial. Tomado de: (Mantilla Parra y Vargas Blanco, 2023)

Existen varios tipos de redes neuronales, cada una diseñada para abordar diferentes tipos de problemas y estructuras de datos (Ver tabla 2.1).

Arquitecturas	Definición
Feed-forward Neural Networks	Cada neurona en una capa está conectada a todas las neuronas de la capa siguiente, y no hay ciclos ni retroalimentación en la red.
Convolutional Neural Networks (CNN)	Están diseñadas específicamente para el procesamiento de datos estructurados en cuadrículas, como imágenes.
Recurrent Neural Networks (RNN)	Son adecuadas para trabajar con datos secuenciales o temporales. Están construidas con conexiones cíclicas que les permiten mantener y utilizar información anterior.
Residual Neural Networks o ResNets	Incorporan conexiones residuales o permiten "saltar conexiones", facilitando el entrenamiento de redes neuronales muy profundas al mitigar el problema de desvanecimiento del gradiente.

Tabla 2.1: Clases de redes neuronales (Hassoun, 1995)

2.2.2. *Entrenamiento*

El proceso de entrenamiento de una red neuronal profunda se desglosa en una serie de pasos fundamentales (Sanchez y Velaquez, 2011). A continuación, se describen los pasos esenciales de este entrenamiento, así como los conceptos claves que intervienen en dicho procedimiento.

En primera instancia, se realiza la recopilación y preparación de los datos, ya que son necesarios para el entrenamiento, la validación y las pruebas (ver tabla 2.2).

Conjunto de datos	Definición
Training data set	Usados para ajustar o entrenar el modelo, el modelo ve y aprende de estos.
Validation data set	Utilizados para proporcionar una evaluación imparcial del ajuste de un modelo en el conjunto de datos de entrenamiento, ajusta los hiperparámetros del modelo, el modelo ve estos datos pero no aprende de ellos
Test dataset	Conjunto de datos separado que se utiliza para evaluar el rendimiento final del conjunto de datos de entrenamiento

Tabla 2.2: Definición de conjuntos de prueba, entrenamiento y validación (Genç y Tunc, 2019).

Asimismo, es importante seleccionar hiperparámetros cruciales para controlar el proceso de aprendizaje, estos hiperparámetros son configuraciones que no se aprenden durante el entrenamiento, sino que se establecen antes del proceso de entrenamiento (ver tabla 2.3).

Realizado lo anterior, se preparan los datos, llevando a cabo un preprocesamiento y normalización para asegurar la calidad y consistencia de los datos. Posteriormente, se procede a la definición del modelo, lo cual implica la creación de la arquitectura de la red neuronal. En esta fase, se determina el número y tipo de capas que conformarán la red, así como la elección de la función de activación que se aplicará a cada neurona en las capas correspondientes. Los tipos de capas son: densas, activación, convolución, pooling, recurrente y de salida.

Hiperparámetros	Definición
Learning rate (tasa de aprendizaje)	Controla el tamaño de los pasos que se dan durante la optimización. Un valor muy alto puede llevar a oscilaciones o que no converja el modelo, mientras que un valor muy bajo puede hacer que el entrenamiento sea lento o que el modelo se quede atascado en mínimos locales.
Batch size (tamaño del lote)	Determina cuántas muestras se utilizan en cada iteración de entrenamiento antes de actualizar los pesos.
Número de épocas	cantidad de veces que el modelo pasa por todo el conjunto de entrenamiento, una época es una iteración.

Tabla 2.3: Hiperparámetros de ajuste(Hassoun, 1995)

Las funciones de activación en una capa son utilizadas para introducir no linealidad en una red neuronal y permitir que la red aprenda representaciones más complejas de los datos (ver tabla 2.4).

Definido el modelo, se procede con la inicialización de pesos y sesgos, esto es importante para evitar problemas como el estancamiento del entrenamiento, lentitud en la convergencia y simetría no deseada ya que, si no se inicializa los pesos o se inicializa de la misma manera, las unidades en una capa aprenderán patrones idénticos y no aportara diversidad al aprendizaje. Entre las técnicas de inicialización de pesos más comunes se encuentra:

- RandomNormal: genera tensores con distribución normal
- RandomUniform: genera tensores con distribución uniforme
- TruncatedNormal: genera una distribución normal truncada.
- HeNormal: inicializa los pesos de forma aleatoria utilizando una distribución gaussiana
- HeUniform: inicializa los pesos de forma aleatoria utilizando una distribución uniforme.

Función	Definición	Expresión
ReLU	Devuelve cero para valores negativos y el valor mismo para valores positivos.	$f(x) = \max(0, x)$
Sigmoide	Transforma los valores de entrada en el rango de 0 a 1.	$f(x) = \frac{1}{1+e^{-x}}$
Leaky ReLU	Es una variante de ReLU que permite una pequeña pendiente negativa α para valores negativos, lo que ayuda a prevenir el problema de "neuronas muertas". ^{en} ReLU.	$f(x) = \begin{cases} x, & \text{si } x > 0 \\ \alpha x, & \text{si } x \leq 0 \end{cases}$
ELU	La pendiente de esta función disminuye exponencialmente a medida que x se acerca a cero, lo cual ayuda a mitigar problemas asociados con ReLU y Leaky ReLU, como "neuronas muertas"z activaciones saturadas.	$f(x) = \begin{cases} x, & \text{si } x > 0 \\ \alpha \cdot (e^x - 1), & \text{si } x \leq 0 \end{cases}$

Tabla 2.4: Funciones de activación más comunes (Rodríguez P. y Buitrago R., 2022)

Asimismo, se define la función de costo que también es conocida como función de pérdida y esta es la que cuantifica el error entre el valor estimado y el valor real (Ver tabla(2.5)). .

En esta etapa, es importante establecer el algoritmo de optimización, este algoritmo se utiliza para ajustar los pesos de la red con el objetivo de minimizar la función de costo (Ver tabla 2.6).

Función	Expresión
Error cuadrático medio	$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$
Error logarítmico cuadrático medio	$\text{MSLE} = \frac{1}{N} \sum_{i=1}^N (\log(y_i+1) - \log(\hat{y}_i+1))^2$
Pérdida de entropía cruzada	$-\frac{1}{N} \sum_{i=1}^N [y_i \cdot \log(\hat{y}_i) + (1 - y_i) \cdot \log(1 - \hat{y}_i)]$

Tabla 2.5: Funciones de costo (Janiesch et al., 2021) .

Algoritmo	Definición
SGD	Actualiza los pesos de la red utilizando el gradiente de la función de costo con respecto a los parámetros, multiplicado por una tasa de aprendizaje fija.
RMSprop	Adapta la tasa de aprendizaje para cada parámetro individual en función de la magnitud de los gradientes pasados.
Adam	Calcula estimaciones del primer momento (media) y del segundo momento (varianza no centrada) de los gradientes para adaptar la tasa de aprendizaje de manera individual para cada parámetro.
Adamax	Es una variante del Adam y la diferencia radica en cómo se calcula el momento de segundo orden, este utiliza el máximo absoluto acumulado y ayuda a controlar mejor las tasas de aprendizaje en presencia de gradientes muy grandes

Tabla 2.6: Algoritmos de optimización más comunes.

Una vez ajustados los parámetros anteriores, se procede a evaluar el desempeño del modelo y cabe resaltar términos como: la tasa de error, el bias y la variance; también es importante mencionar problemas asociados con el rendimiento del modelo durante el entrenamiento, tales como el underfitting y el overfitting. La realidad es que cualquier modelo, por más preciso que sea en su ajuste no está exentos de ellos.

Término	Definición
Tasa de error	Cuantifica la proporción de predicciones incorrectas realizadas por un modelo con respecto al total de predicciones
Bias	Es la diferencia entre el valor esperado o predicho y el valor real. (tasa de error de los datos de entrenamiento)
Variance	Es cuánto varía la predicción según los datos que se utilizan para el entrenamiento. (tasa de error de los datos de prueba)
Underfitting (Subajuste)	El modelo tiene una tasa de error alto tanto en los datos de entrenamiento como en los datos de validación, es decir, el modelo es demasiado simple para aprender o ajustarse a los datos, este problema puede surgir al entrenar el modelo con pocas épocas o con un conjunto de datos muy pequeños.
Overfitting (Sobreajuste)	El modelo tiene una tasa de error baja en los datos de entrenamiento y alta en los datos de validación, lo que significa que el modelo se vuelve demasiado específico para los datos de entrenamiento y, por lo tanto, tiene un rendimiento deficiente en nuevos datos que no ha visto antes, como el conjunto de prueba.

Tabla 2.7: Terminología encargada de evaluar el desempeño de una red neuronal profunda (Bashir et al., 2020).

Para abordar el Overfitting, es común utilizar técnicas de regularización (Goodfellow et al., 2016), entre las técnicas más comunes:

- Dropout: consiste en desactivar aleatoriamente un porcentaje de neuronas o unidades ocultas durante el entrenamiento de una red neuronal. Esto significa que, en cada iteración o época del entrenamiento, algunas neuronas se seleccionan al azar y se omiten

junto con sus conexiones.

- L1 (Regularización de Lasso): agrega un término de penalización a la función de costo del modelo que es proporcional a la suma de los valores absolutos de los coeficientes o pesos del modelo.
- L2 (Regularización de Ridge): agrega un término de penalización a la función de costo del modelo que es proporcional a la suma de los cuadrados de los valores de los coeficientes o pesos del modelo.

Por último, para monitorear y controlar el entrenamiento de la red, normalmente se agregan Callbacks, como:

- Early stopping (interrupción anticipada): detiene el entrenamiento tan pronto como el error de validación alcance un mínimo, significa que, si el rendimiento en el conjunto de validación deja de mejorar o comienza a empeorar, se detiene el entrenamiento antes de que se alcance el punto de sobreajuste.
- ReduceLROnPlateau: se utiliza para ajustar automáticamente la tasa de aprendizaje durante el entrenamiento del modelo. Su objetivo principal es mejorar la convergencia del modelo ajustando dinámicamente la tasa de aprendizaje.

En resumen, la elección cuidadosa de funciones de pérdida y optimizadores, junto con la validación y ajuste de hiperparámetros, demuestra ser esencial en la búsqueda de un modelo de rendimiento óptimo.

3. Metodología

En esta sección, se detalla la creación de las red neuronales, abordando su arquitectura y proceso de entrenamiento. Se describe el desarrollo de la base de datos utilizada para entrenar la redes neuronales. Asimismo, se analizan los tiempos de procesamiento de la red en comparación con el método analítico, proporcionando una evaluación de la eficiencia de EIKO4P-S respecto a Fast Marching.

3.1. Construcción base de datos

Para el cálculo de los tiempos se crearon tres bases de datos a partir del algoritmo de (C.A.Niño et al., 2022) mencionado en la sección 2.1.1, el cual proporcionó los tiempos de viaje de ondas P, PP y PS, dicho programa se modificó para la creación de 100 diferentes modelos aleatorios de reflectores entre formas V y senoidales, además de los tiempos asociados a cada geometría, con 20 modelos de velocidad distintos y 49 $Shot_i$ s en posiciones fijas, donde cada $Shot_i$ es analizado de forma independiente para cada modelo.

Los tiempos P, PP y PS son representados como se observa la figura (3.1), estos tiempos están sobre un modelo espacial con una velocidad implícita, dichos modelos tienen una dimensión espacial de $2 \times 2 \text{ km}$ representados por imágenes de mapa de bits de 79×79 pixeles donde cada píxel corresponde a una ubicación específica del modelo espacial y representa el tiempo ya sea P, PP, o PS en esa ubicación.

Para caracterizar la base de datos se tuvieron en cuenta los siguientes parámetros extraídos del modelo obtenido de Fast Marching:

- Velocidad del modelo
- Forma del reflector
- Localización del $Shot_i$
- Tiempo asociado en un punto del modelo espacial

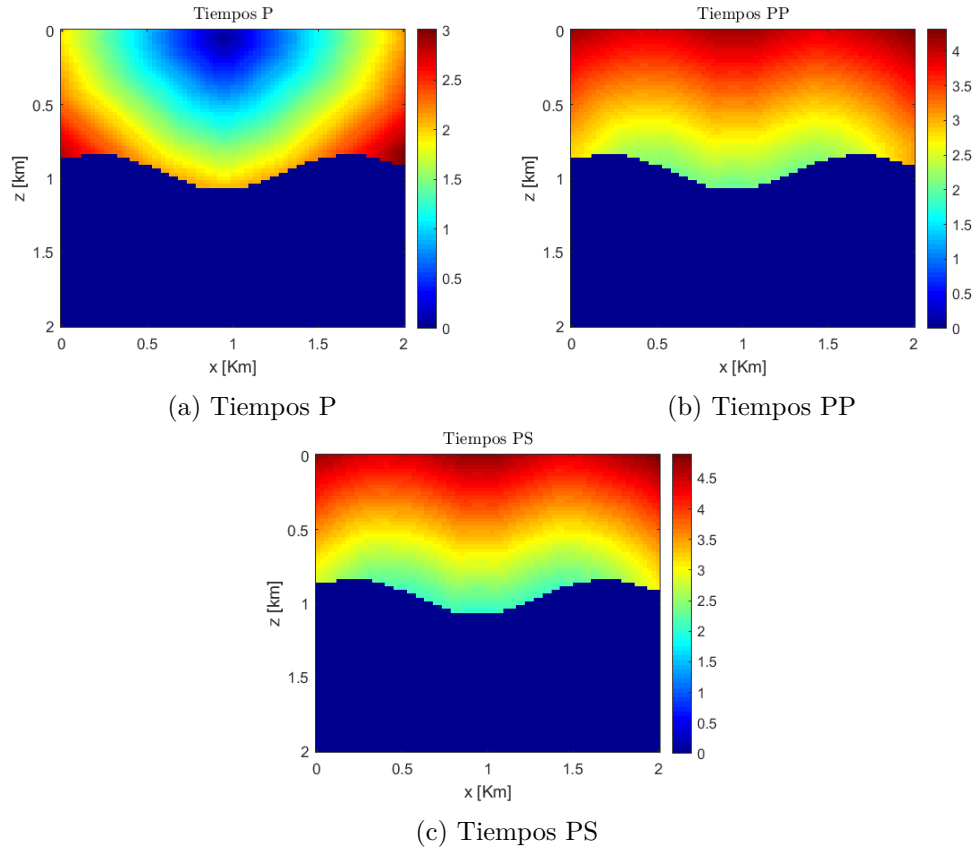


Figura 3.1: Ejemplo de tiempos de viaje P, PP y PS en el modelo espacial; en un modelo senoidal con velocidad 0.5 [km/s]

El primer enfoque se basó en resolver el problema por medio de reconocimiento de imágenes, sin embargo, esta idea se descartó porque se observó que para poder extraer las características mencionadas anteriormente por medio de imágenes se deben manejar los datos como tensores, lo que aumentaba la cantidad de hardware requerido para realizar esta prueba.

Rápidamente se tomó un enfoque distinto propuesto por EIKONET (J. D. Smith et al., 2021) en un espacio 3D, en donde se acopló la idea a un espacio 2D, que consiste en dos conceptos principales, el de fuente y otro de receptor, como un par de puntos asociados, mientras que EIKONET tomó una fuente y un receptor aleatorios en todo el espacio 3D; en el enfoque de EIKO4P-S se desarrolló a partir de fuentes fijas equiespaciadas y definidas por el algoritmo de (C.A.Niño et al., 2022) en donde se tomó como par de puntos fuente ($Shot_i$)

y un valor aleatorio dentro del espacio delimitado por el reflector modelado por la función level set $\phi(x, z) = 0$ en el espacio de Ω_1 , que se denomina receptor. Esto permitió tomar unos cuantos puntos aleatorios en el modelo espacial. De tal forma que al entrenar la red, esta sea capaz de predecir los tiempos en cada uno de los puntos del modelo espacial que están en Ω_1 . para posteriormente realizar un post procesamiento de estos datos y reconstruir la imagen de tiempos.

Como generalidad en la construcción de las bases de datos para tiempos P, PP y PS en cada una la cantidad corresponde a un total de 4.900.000. Estos datos son extraídos, de 50 receptores en puntos aleatorios tomados de Ω_1 para cada uno de los 49 shots en 20 modelos de velocidad diferentes y para 100 geometrías distintas. También se tomaron un 80% de dichos datos para el entrenamiento de la red neuronal con un total de 3.9200.000 datos de entrenamiento y el 20% restante 980.000 se utilizaron como datos de validación.

3.1.1. Base de datos tiempos P

En la base de datos de tiempos P se recopilieron un total de 84 características, que se dividen en cinco categorías principales:

Categoría	Definición
Profundidad del Reflector	79 puntos que representan la profundidad del reflector. Estos puntos se denotan como $z_{r1}, z_{r2}, z_{r3}, \dots, z_{r79}$.
Posición de la Fuente	Posición espacial de la fuente, que se especifica en términos de sus coordenadas x_s y z_s .
Posición del Receptor	Posición espacial del receptor, que se especifica en términos de sus coordenadas x_r y z_r .
Velocidad	Valor de la velocidad correspondiente a cada punto en el estudio.
Etiqueta de Tiempo	Para cada punto en la base de datos, se asoció un valor de tiempo correspondiente.

Tabla 3.1: Recopilación de datos para la base de datos P

En la tabla (3.2) se presenta como queda compuesta la base de datos para los tiempos P .

Coordenadas z del reflector	Coordenadas del shot		Coordenadas punto aleatorio de Ω		Velocidad	Tiempos
Dirz [km] (79 puntos)	Coordenada eje z [km]	Coordenada eje x [km]	Coordenada eje z [km]	Cordenada eje z [Km]	Velocidad del modelo [v]	Tiempos P asociado
84 Características						Etiqueta

Tabla 3.2: Composición de la base de datos para los tiempos P

3.1.2. Base de datos tiempos PP y tiempos PS

La base de datos de tiempos PP y PS consta de un total de 163 características, junto con sus respectivas etiquetas (tiempo PP o PS). Las características se dividen en varias categorías principales:

Categoría	Definición
Tiempos P (PP) en el reflector	79 tiempos P sobre el reflector, que se denotan como $T_{p1}, T_{p2}, T_{p3}, \dots, T_{p79}$
Profundidad del Reflector	79 puntos que representan la profundidad z del reflector, identificados como $z_{r1}, z_{r2}, z_{r3}, \dots, z_{r79}$
Velocidad del Modelo	Valores de velocidad del modelo $\frac{\text{km}}{\text{s}}$
Posición de la Fuente	Posición espacial de la fuente, especificada por las coordenadas x_s y z_s
Posición del Receptor	Posición espacial del receptor, con coordenadas x_r y z_r
Tiempo PP o PS	Cada conjunto de características mencionado se asoció con un valor de tiempo PP o PS correspondiente

Tabla 3.3: Recopilación de datos para la base de datos PP y PS

La tabla (3.4) resumen de la composición de la base de datos para los tiempos PP y PS.

Tiempos P sobre el reflector [s]	Coordenadas z del reflector [km]	Coordenadas del shot [km]	Coordenadas punto aleatorio de Ω [km]		Velocidad $\frac{km}{s}$	Tiempos [s]	
Tiempos P (79 tiempos)	Dirz (79 puntos)	Coordenada eje z	Coordenada eje x	Coordenada eje z	Cordenada eje z	Velocidad del modelo	Tiempos PP/PS asociado
163 Características							Etiqueta

Tabla 3.4: Composición de la base de datos para los tiempos PP y PS

Es importante resaltar que las bases de datos de tiempos PP y tiempos PS son dos bases de datos diferentes ya que la velocidad de ondas PP y PS no son iguales.

3.2. Red neuronal

El enfoque empleado para resolver la ecuación EIKONAL en el cálculo de los tiempos de viaje de primeros arribos, se llevo a cabo mediante un conjunto de redes neuronales profundas con una estructura y forma idénticas, denominada Eiko4P-S. Este conjunto consiste en tres redes neuronales independientes interconectadas entre sí, las cuales se encargan de calcular los tiempos P, PP y PS en un modelo de velocidad homogéneo y predefinido, sobre un reflector. Como primera predicción de las redes, se calcula los tiempos P que posteriormente tienen la función de servir como uno de los parámetros de entrada en el cálculo de los tiempos PP y tiempos PS, gráficamente se puede observar en la figura (3.2):

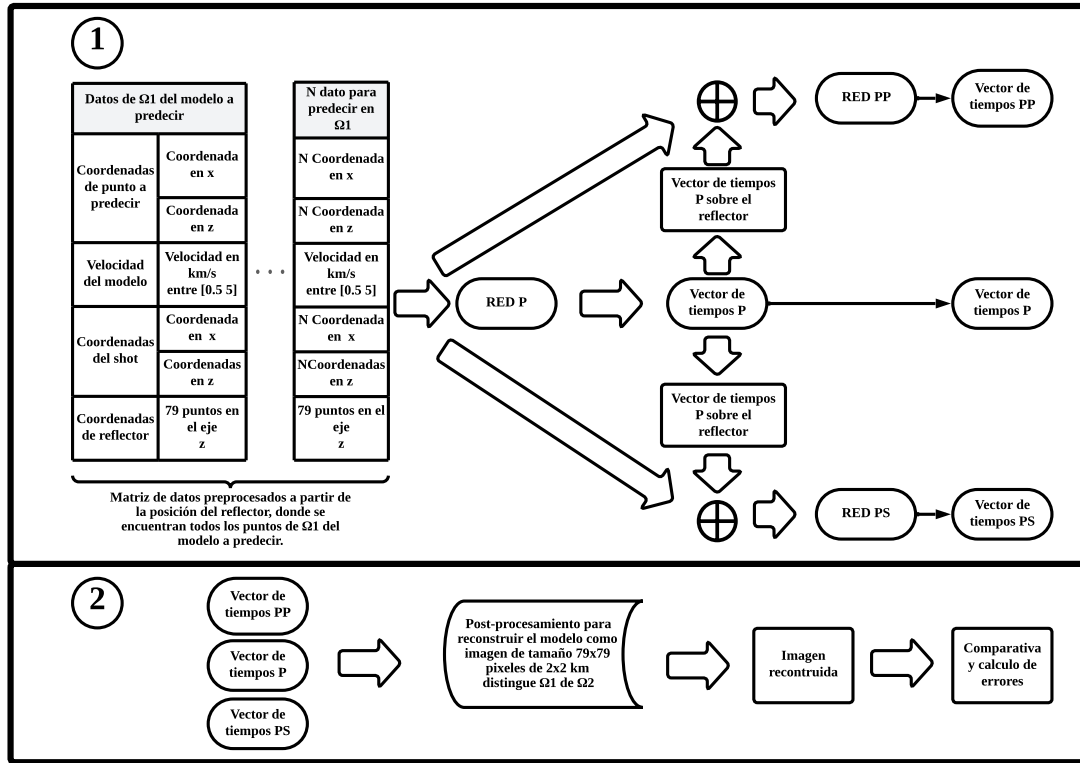


Figura 3.2: Descripción general funcionamiento EIKO4P-S. (1) Se introducen los datos a ser predichos en el espacio de ω_1 , primero alimentan a la red P que posteriormente estima el vector de tiempos P, de allí se extraen los tiempos P sobre la posición del reflector y se concatenan a la matriz de datos para formar la entrada de las redes PP y PS que estiman los vectores de tiempos correspondientes, (2) explica la reconstrucción de la imagen a partir de los datos predichos.

Además, para el entrenamiento de las redes el cambio principal entre ellas es la base de datos ingresada, ya que en el enfoque de fuente reflector en una primera instancia con los tiempos P se tiene un par de puntos fuente-receptor sobre un modelo de velocidad definido, mientras que para los tiempos PP y PS se tomó como fuente todos los tiempos sobre el reflector hasta un punto aleatorio en el espacio predefinido.

3.2.1. Arquitectura

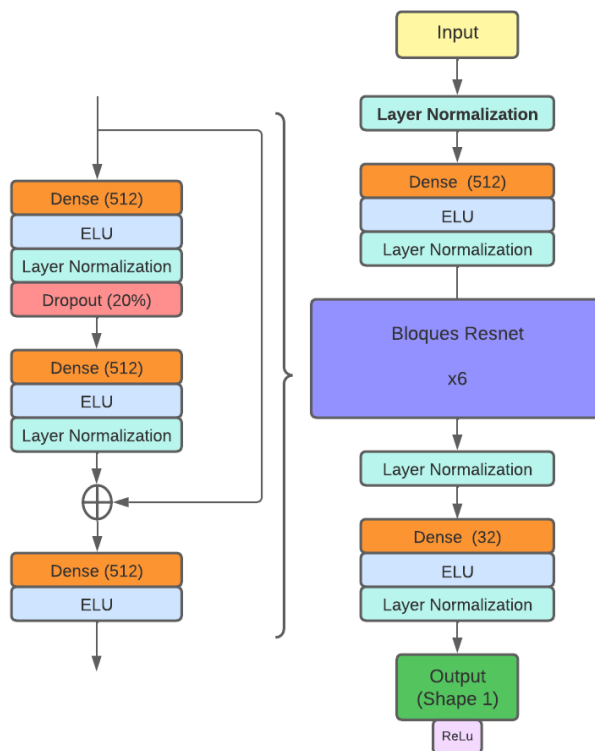


Figura 3.3: Arquitectura red neuronal propuesta

El diseño de la arquitectura de la red EIKO4P-S, tiene como referencia la estructura de las redes neuronales residuales (ResNet). Esta arquitectura se compone de una red neuronal feed-forward que consta de bloques residuales conectados con capas densas (fully connected layers). La elección de utilizar una arquitectura ResNet se basa en las múltiples ventajas que esta ofrece como lo son: su capacidad para modelar relaciones no lineales complejas en los datos (Chen et al., 2020) lo cual es el caso de este proyecto, también ayudan en la mitigación del desvanecimiento del gradiente lo cual permite que las redes neuronales puedan ser muy profundas (He et al., 2016), por último esta arquitectura ofrece una mayor generalización y robustez que facilita que las redes neuronales capten patrones y relaciones sutiles en los datos, lo que contribuye a prevenir el overfitting.

Respecto a la función de activación para las capas densas se usó la Exponential Linear Unit (ELU) ya que como se menciona (He et al., 2015) esta función de activación tiene un

mejor rendimiento para redes profundas como lo muestra en el anexo. Sin embargo, para la capa de salida se utilizó la rectified linear unit (ReLU), esto con el fin de prevenir que la red presenta valores negativos en su predicción. Por otra parte para la inicialización de los pesos en las capas se utilizó HeUniform (Clevert et al., 2016).

Para la elección de la capa de normalización se realizó un entrenamiento del modelo con la arquitectura propuesta pero variando las capas de normalización y se comparó cual obtuvo una menor pérdida en el entrenamiento y en los datos de validación. Se realizaron pruebas con 3 tipos de capas de normalización: Layer normalization, Batch Normalization y sin normalización.

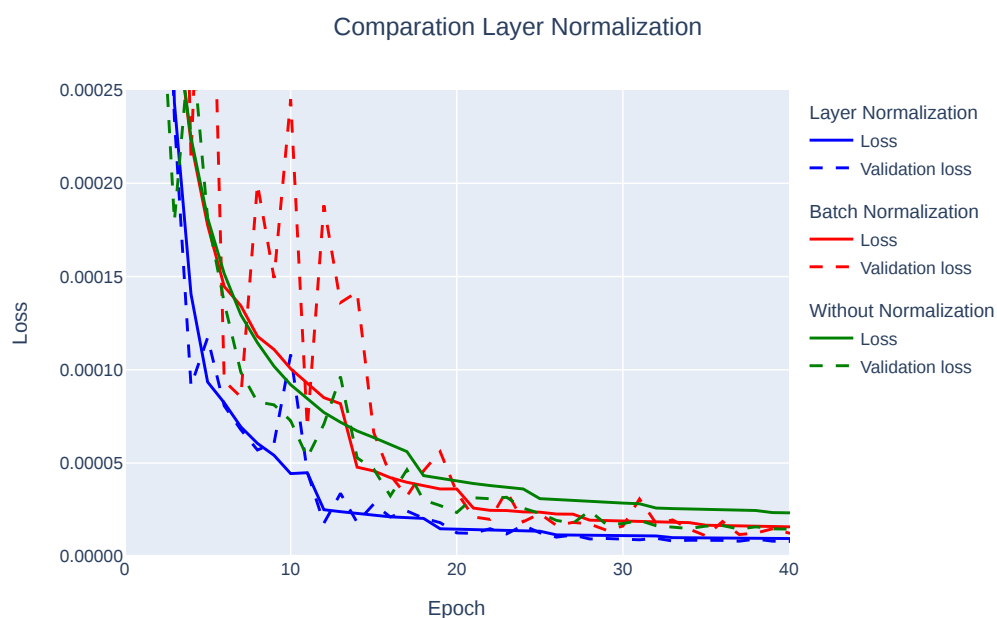


Figura 3.4: Comparación rendimiento para 3 capas de normalización.

Durante el proceso de entrenamiento se observó que era fundamental la normalización de los datos de entrada, ya que sin este paso la red no convergia a menos que se cambiara el learning rate a valores muy bajos .

Como se observa en la figura (3.4) la red tuvo un mejor rendimiento al usar Layer normalization como capa de normalización por lo tanto esta capa fue la seleccionada para la arquitectura propuesta de la red.

Como regularizador se utilizan capas de dropout del 20% en los bloques residuales, esto con el fin de que la red generalice mejor los datos y se evite el overfitting. La elección de donde se ubican estas capas se basa en (Kim et al., 2023), donde menciona que la ubicación elegida que es dentro de los bloques resnet y después de la primera capa densa, fue donde se obtuvo un mejor rendimiento para redes Resnet como es el caso de la arquitectura propuesta. Por último el total de parámetros entrenables en la red son de 4,793,921.

3.2.2. *Entrenamiento*

Para el entrenamiento de la red se trató de tomar un enfoque parecido al que se tomó en eikonet (J. D. Smith et al., 2021) en donde los mejores resultados se obtuvieron usando tamaños de batch pequeños y además el entrenamiento lo realizan con 200 épocas. En el caso de EIKO4P-S se realizaron algunas modificaciones a lo mencionado anteriormente.

Respecto al número de épocas y el tamaño del batch debido a las limitaciones que se contaba con Google Colab se utilizó un tamaño del batch de 1024 ya que un número más bajo implica mayor tiempo de cálculo y no necesariamente resultados mucho mejores con respecto a los que se obtuvieron con el batch de 1024; el número de épocas de entrenamiento fue de 100 esto para limitar el tiempo de entrenamiento de la red que es aproximadamente de 3 horas por sesión cuando se utiliza la GPU, fundamental para el cálculo paralelo requerido en las redes neuronales profundas. Para la función de pérdida se utilizó la Mean Squared Logarithmic Error teniendo en cuenta lo mencionado en (Jadon et al., 2022) que indica que para problemas de regresión donde se quiere penalizar valores más cercanos a cero y no penaliza tanto los valores más alejados, lo cual se ajusta a la problemática que se tenía ya que los errores más grandes se presentaban en las predicciones de los valores de tiempos entre 0 y 1, por lo que se definió como función de pérdida al cumplir con dichas características.

Como optimizador se usó el algoritmo de Adamax con un Learning rate inicial de 0.025, la razón para el uso de Adamax es que es una variación del algoritmo de Adam el cual se usa comúnmente para redes neuronales con grandes datasets y/o gran número de

parámetros como se menciona en (Kingma y Ba, 2017).

Por último, durante el entrenamiento se adicionaron los siguientes Callbacks:

ReduceLRonPlateau: permitió reducir el learning rate del optimizador, para ayudar a la red a converger.

EarlyStopping: para el caso de esta red se decidió tomar 10 épocas como el umbral para detener el entrenamiento.

A continuación, en la tabla (3.5), se encuentra un resumen de los parámetros de entrenamiento que se utilizaron para la red Eiko4P-S:

Número épocas	Función de Pérdida	Tamaño del batch	Callback	Optimizador	Learning rate
100	Mean Squared Logarithmic Error	1024	-ReduceLRonPlateau -Early Stopping	Adamax	0.025

Tabla 3.5: Resumen parámetros de entrenamiento para las redes de EIKO4P-S.

3.3. Tiempo de procesamiento

El tiempo necesario para calcular los tiempos de los viajes P, PP y PS utilizando el método analítico de (C.A.Niño et al., 2022) se presenta en la figura (3.6). Es importante destacar que, para el proceso completo del modelo analítico, es necesario primero esquematizar el modelo de geometría. Luego, se realiza un ajuste del contorno de nivel de $\phi^k(x, z) = 0 | k = 1, 2, \dots, end$, que tiende a converger en el reflector geológico real, denotado como R . Durante este proceso, también se ajustan las velocidades V_p^k y V_s^k . La duración del ajuste depende de la profundidad, la complejidad del modelo y el número de iteraciones definidas manualmente, dicho proceso es la etapa que mayor tiempo requiere, debido a que puede durar desde unos minutos hasta unas cuantas horas.

Una vez que se ha encontrado la geometría, esta se utiliza para calcular los tiempos

de viaje P, PP y PS utilizando el método de Fast Marching.

En contraste, el modelo EIKO4P-S requiere la información espacial de la ubicación del reflector como uno de los parámetros de entrada en el modelo ya entrenado, utilizado para calcular los tiempos P, PP y PS, entre otros. Al eliminar la necesidad de realizar un ajuste de contorno, este enfoque ahorra tiempo en la fase de procesamiento destinada a la determinación de la geometría. Como resultado, el tiempo necesario para llevar a cabo el proceso completo es inferior en comparación con el enfoque del modelo analítico.





COMPARATIVA DE TIEMPOS DE PROCESAMIENTO							
Modelo	Numero de shot	Tiempos P [ms]		Tiempos PP [ms]		Tiempos PS [ms]	
		EIKO 4P-S	Fast Marching	EIKO 4P-S	Fast Marching	EIKO 4P-S	Fast Marching
	1	4-7	40~68	4	129~132	4	120~133
	49	198	3379	198	6360	198	6490
	1	1001	160~170	1001	320 ~330	1001	320~330
	49	49196	8200	49198	16080	49197	16440

Tabla 3.6: Comparación tiempos de procesamiento teniendo los parámetros de reflector en el calculo de tiempos P, PP y PS entre Fast Marching y EIKO4P-S.

La Tabla 3.6, muestra la comparación del tiempo de procesamiento entre el modelo EIKO4P-S y el modelo Fast Marching (C.A.Niño et al., 2022) en la estimación de los tiempos P,PP y PS.

Se observó que con el modelo EIKO4P-S, la duración de los cálculos aumentó a medida que la geometría se encontraba a mayor profundidad. Sin embargo, también se observó que a medida que la geometría se encontraba más superficial, los cálculos se realizaban más

rápidamente. La red neuronal obtuvo un tiempo de cálculo 3 veces mayor para los tiempos P y 6 veces mayor para los tiempos PP y PS en geometrías más profundas. Por otro lado, EIKO4P-S resultó ser 17 veces más rápido que Fast Marching para geometrías superficiales y 32 veces más rápido en el cálculo de los tiempos PP y PS, esto se debió a que la red neuronal calculó menos puntos para geometrías más superficiales, lo que le permitió calcular la solución de (ω_1) mucho más rápido que el modelo analítico.

Por lo tanto para geometrías mas profundas, se sugiere considerar el uso de un hardware de mayor potencia con una GPU más potente para mejorar el rendimiento en el cálculo de tiempos utilizando EIKO4P-S.

4. Resultados experimentales

En el apartado de resultados se pondrá a prueba la red neuronal Eiko4P-S para cuatro geometrías con cuatro velocidades diferentes que no han sido vistas por la red durante el entrenamiento. A continuación, se podrá observar las geometrías junto a su respectiva velocidad:

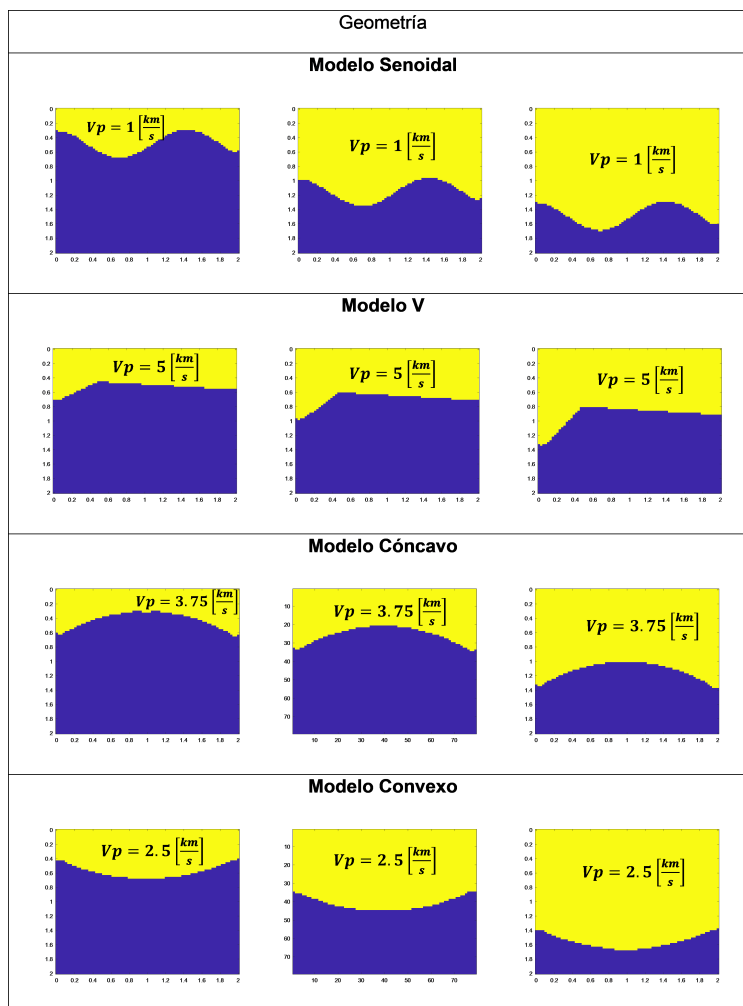


Figura 4.1: Geometrías usadas en el apartado de resultados

Es importante resaltar que para cada geometría se mostrarán resultados para tres $Shot_i$ s diferentes (Localizaciones de fuente diferentes en el modelo espacial).

Para presentar los resultados se utilizaran 3 gráficas para cada uno de los modelos presentados anteriormente:

- Gráfica de contorno
- Gráfica comparación de tiempos de primeros arribos en la superficie de tiempos PP y PS.
- Gráfica del error absoluto.

Cada predicción realizada por EIKO4P-S se compara con los valores obtenidos por Fast marching. Las gráficas de contorno es una ayuda visual que nos permite corroborar el ajuste de los tiempos de viaje de ondas P, reflejadas PP y convertidas PS de los diferentes modelos de velocidad, diferentes $Shot_i$ s y diferentes alturas de una misma geometría, en comparación con los errores generados entre las predicciones impuestas por la red neuronal y el del modelo analítico Fast-Marching.

En cada grafica de contorno se encontraran los resultados para los tiempos P, PP y PS de tres $Shot_i$ s diferentes, por lo tanto habrán 9 mapas de contorno por cada gráfica, los cuales estarán etiquetados con letras (a,b,c,d,e,f,g,h,i), el orden de cada gráfica de contorno será el siguiente:

Orden gráficas de contorno	
Primera columna	mapas de contorno de los tiempos P, estos estarán etiquetados con las letras (a,d,g)
Segunda columna	Mapas de contorno de los tiempos PP, estos estarán etiquetados con las letras (b,e,h)
Tercera columna	mapas de contorno de los tiempos PS, estos estarán etiquetados con las letras (c,f,i)
Cada fila	Representara un $Shot_{ii}$ diferente, para los resultados se utilizaran los $Shot_{ii}$ s 1,25 y 49.

Tabla 4.1: Organización de mapas de contorno de los modelos con geometrías de forma V, senoidal, cóncavo y cóncavo inverso

4.1. Modelo Senoidal:

Las figuras (4.2) hasta la figura (4.10) detallan los resultados de las pruebas sobre un modelo senoidal escalado a diferentes alturas: superficial, mitad de modelo espacial y por último el más alejado de la superficie. En estos se observan los mapas de contornos, los tiempos de viaje sobre la superficie Γ de los tiempos PP y PS, por último los errores asociados a cada uno de los mapas de contorno de cada uno de los modelos.

4.1.1. Cercano a la superficie

En el modelo más cercano a la superficie se puede encontrar primero en la figura (4.2) el mapa de contorno que muestra la comparación de EIKO4 PS con Fast Marching. Para la figura (4.3) se encuentran los tiempos de viaje sobre la superficie Γ de los tiempos PP y PS entre la Red Neuronal y el modelo analítico. Por último, los errores asociados a cada uno de los mapas de contorno en la figura (4.4).

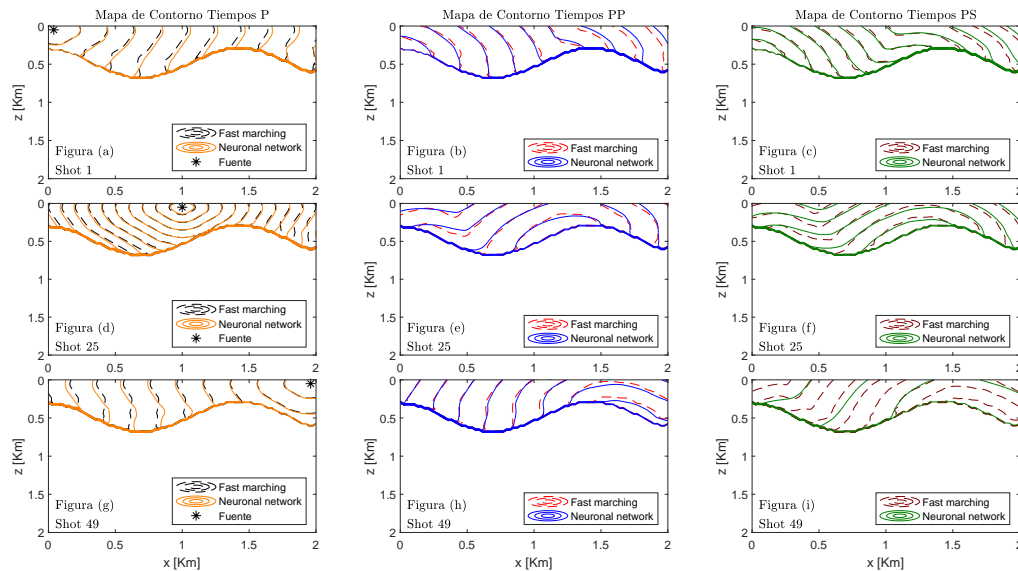


Figura 4.2: Mapa de contorno de modelo senoidal con velocidad de 1km/s, columnas de tiempos P, PP y PS respectivamente, filas 1,2,3 posición del Shot₁ [50 ,40] m, Shot₂₅ [50 ,1000] m, Shot₄₉ [50 ,1960] m, respectivamente

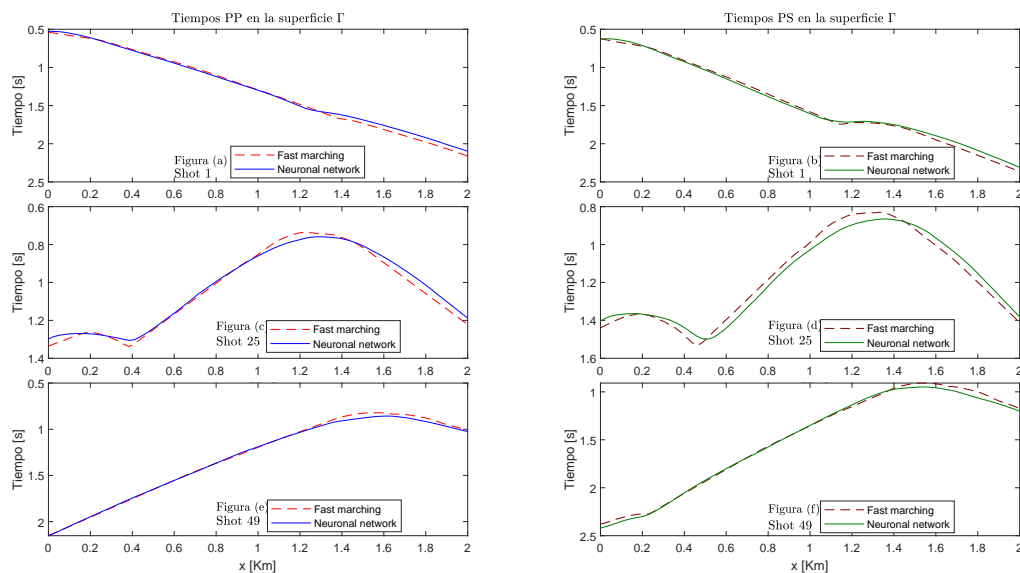


Figura 4.3: Comparativa de tiempos de viaje sobre el reflector Γ entre Fast marching y EIKO-4PS , columnas 1,2 tiempos PP y PS respectivamente, filas 1,2,3 posición del Shot₁ [50 ,40] m, Shot₂₅ [50 ,1000] m, Shot₄₉ [50 ,1960] mm, respectivamente

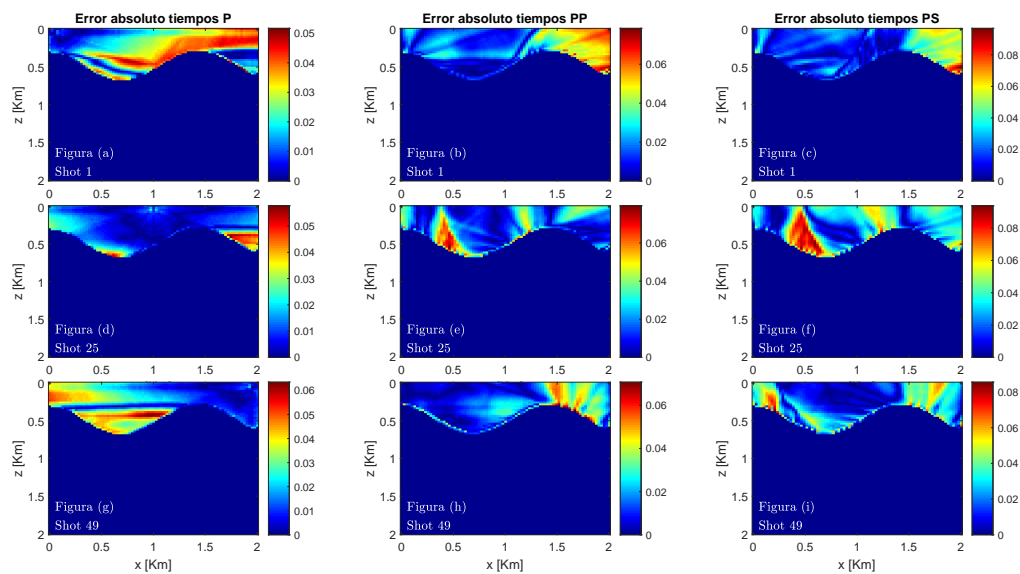


Figura 4.4: Errores absolutos del modelo senoidal con velocidad de 1km/s, columnas de tiempos P, PP y PS respectivamente, filas 1,2,3 posición del Shot₁ [50 ,40] m, Shot₂₄ [50 ,1000] m, Shot₄₉ [50 ,1960] m, respectivamente

4.1.2. Mitad del modelo espacial

En el modelo ubicado en la parte central de espacio predefinido se puede encontrar primero en la figura (4.5) el mapa de contorno que muestra la comparación de EIKO4 PS con Fast Marching. Para la figura (4.6) se encuentran los tiempos de viaje sobre la superficie Γ de los tiempos PP y PS entre la Red Neuronal y el modelo analítico. Por último, los errores asociados a cada uno de los mapas de contorno en la figura (4.7).

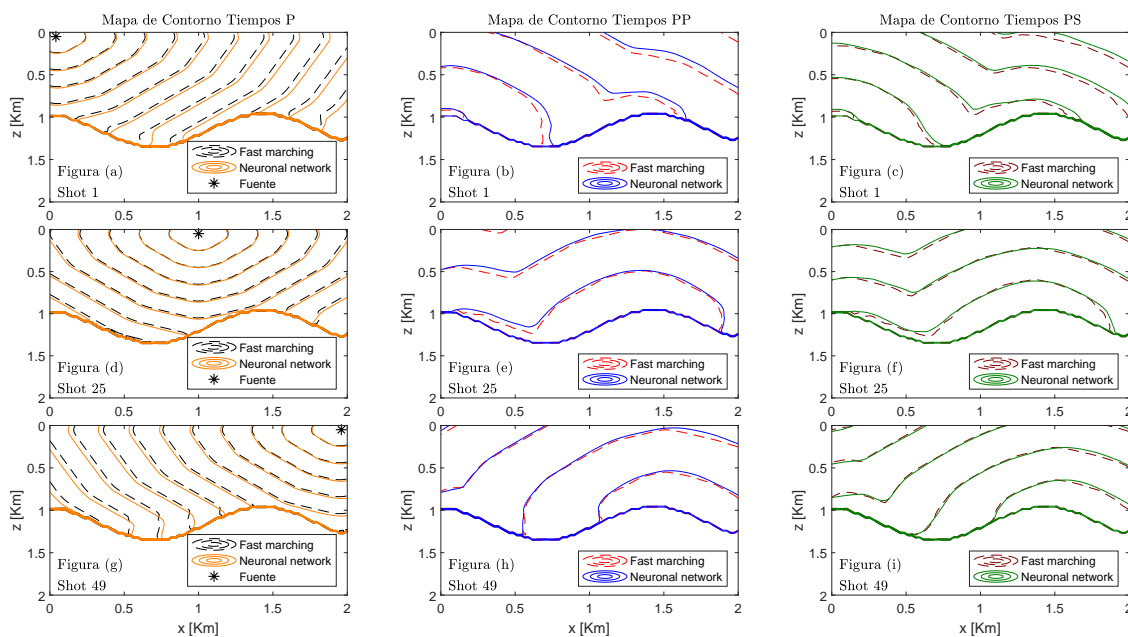


Figura 4.5: Mapa de contorno de modelo senoidal seccion media con velocidad 1km/s, columnas de tiempos P, PP y PS respectivamente, filas 1,2,3 posición del Shot₁ [50 ,40] m, Shot₂₅ [50 ,1000] m, Shot₄₉ [50 ,1960] m, respectivamente

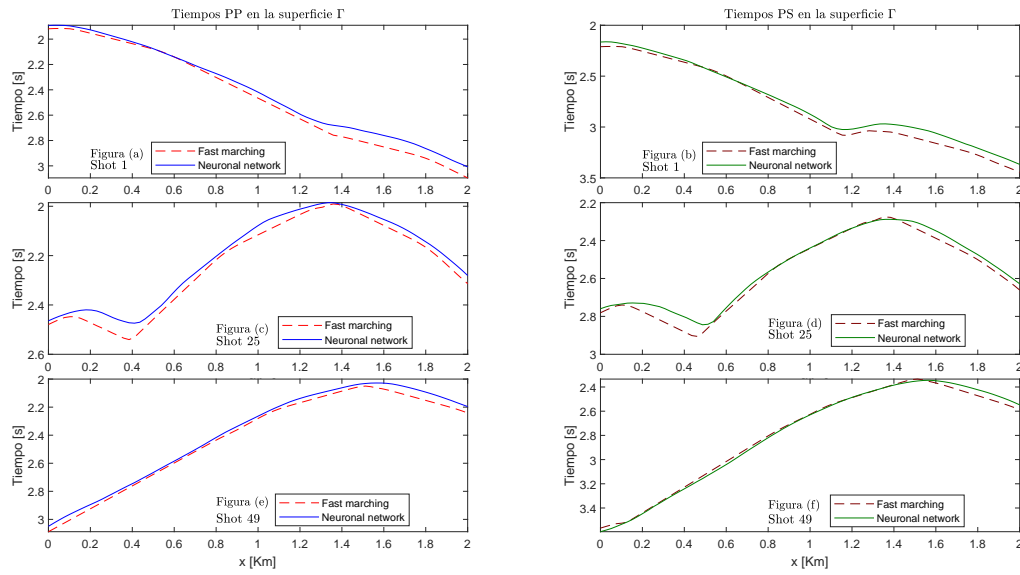


Figura 4.6: Comparativa de tiempos de viaje sobre el reflector Γ entre Fast marching y EIKO-4PS , columnas 1,2 tiempos PP y PS respectivamente,filas 1,2,3 posición del Shot₁ [50 ,40] m, Shot₂₅ [50 ,1000] m, Shot₄₉ [50 ,1960] m, respectivamente

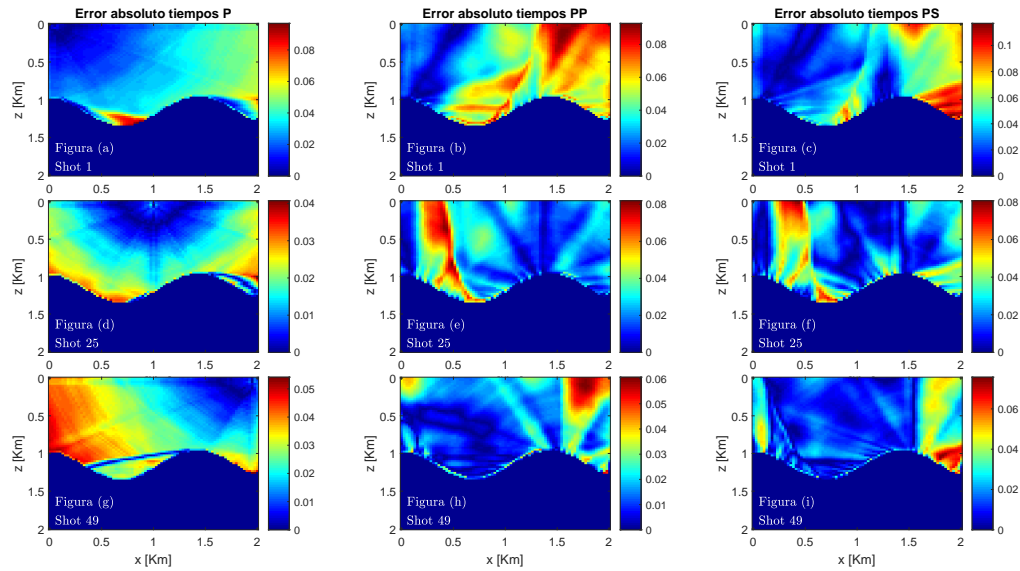


Figura 4.7: Errores absolutos del modelo senoidal con velocidad de 1km/s, columnas de tiempos P, PP y PS respectivamente, filas 1,2,3 posición del Shot₁ [50 ,40] m, Shot₂₅ [50 ,1000] m, Shot₄₉ [50 ,1960] m, respectivamente

4.1.3. Lejano a la superficie

En el modelo mas alejado de la superficie se encuentran en la figura (4.8) el mapa de contorno que muestra la comparación de EIKO4 PS con Fast Marching. Para la figura (4.9) se encontraron los tiempos de viaje sobre la superficie Γ de los tiempos PP y PS entre la Red Neuronal y el modelo analítico. Por último, los errores asociados a cada uno de los mapas de contorno en la figura (4.10).

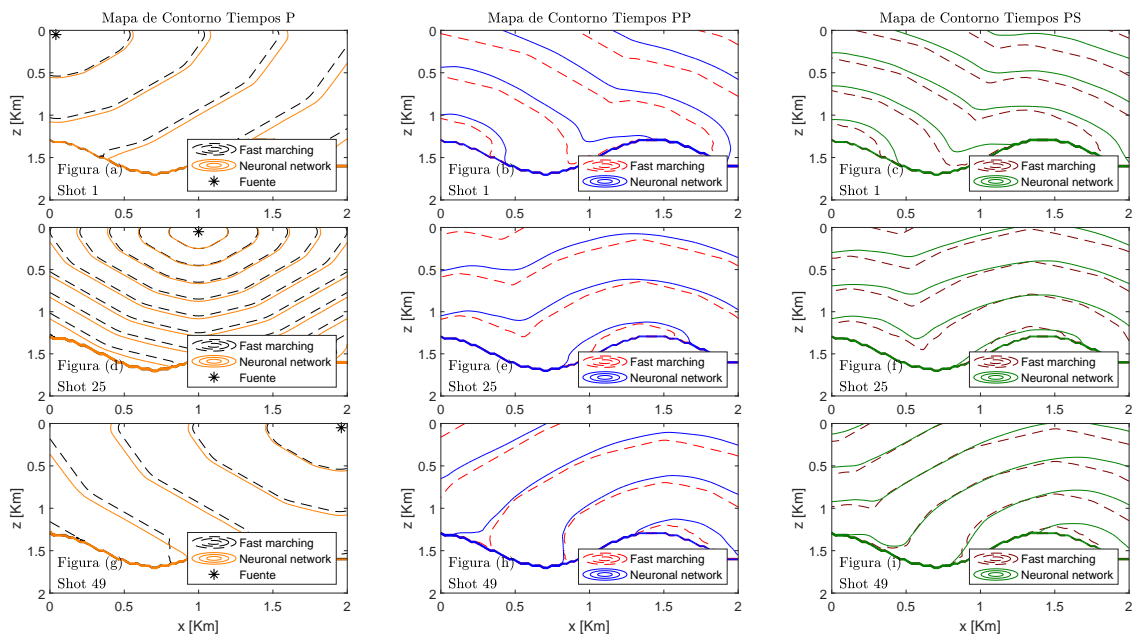


Figura 4.8: Mapa de contorno de modelo senoidal de mayor profundidad con velocidad 1km/s, columnas de tiempos P, PP y PS respectivamente, filas 1,2,3 posición del Shot₁ [50 ,40] m, Shot₂₅ [50 ,1000] m, Shot₄₉ [50 ,1960] m, respectivamente

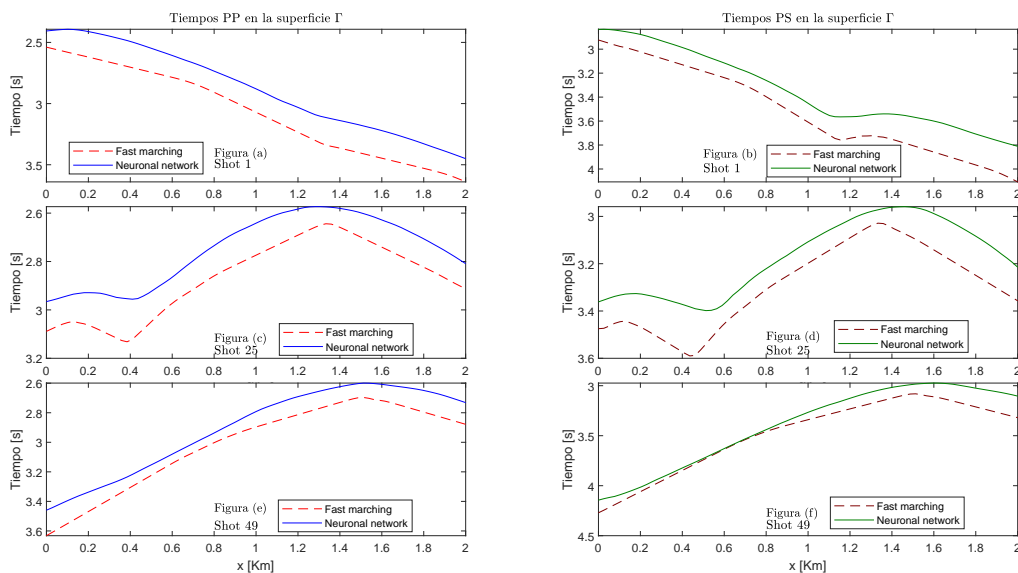


Figura 4.9: Comparativa de tiempos de viaje sobre el reflector Γ entre Fast marching y EIKO-4PS , columnas 1,2 tiempos PP y PS respectivamente,filas 1,2,3 posición del Shot₁ [50 ,40] m, Shot₂₅ [50 ,1000] m, Shot₄₉ [50 ,1960] m, respectivamente

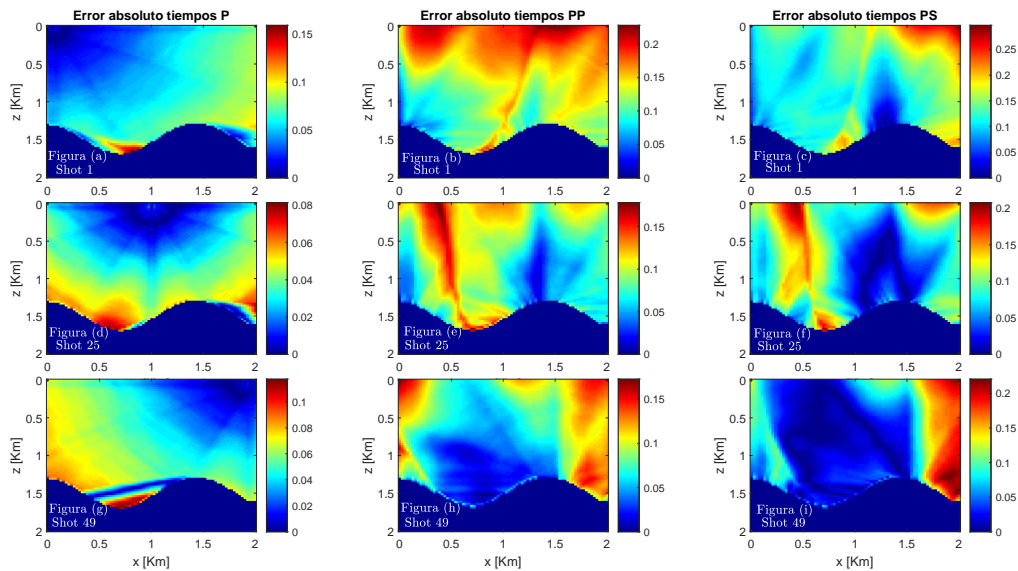


Figura 4.10: Errores absolutos del modelo senoidal con velocidad de 1km/s, columnas de tiempos P, PP y PS respectivamente, filas 1,2,3 posición del Shot₁ [50 ,40] m, Shot₂₅ [50 ,1000] m, Shot₄₉ [50 ,1960] m, respectivamente

4.2. Modelo V:

Las figuras (4.11) hasta la figura (4.19) detallan los resultados de las pruebas sobre un modelo en forma de V invertida escalado a diferentes alturas: superficial, mitad de modelo espacial y por último el más alejado de la superficie. En estos se observan los mapas de contornos, los tiempos de viaje sobre la superficie Γ de los tiempos PP y PS, por último los errores asociados a cada uno de los mapas de contorno de cada uno de los modelos.

4.2.1. Cercano a la superficie

En el modelo más cercano a la superficie se puede encontrar primero en la figura (4.11) el mapa de contorno que muestra la comparación de EIKO4 PS con Fast Marching. Para la figura (4.12) se encuentran los tiempos de viaje sobre la superficie Γ de los tiempos PP y PS entre la Red Neuronal y el modelo analítico. Por último, los errores asociados a cada uno de los mapas de contorno en la figura (4.13).

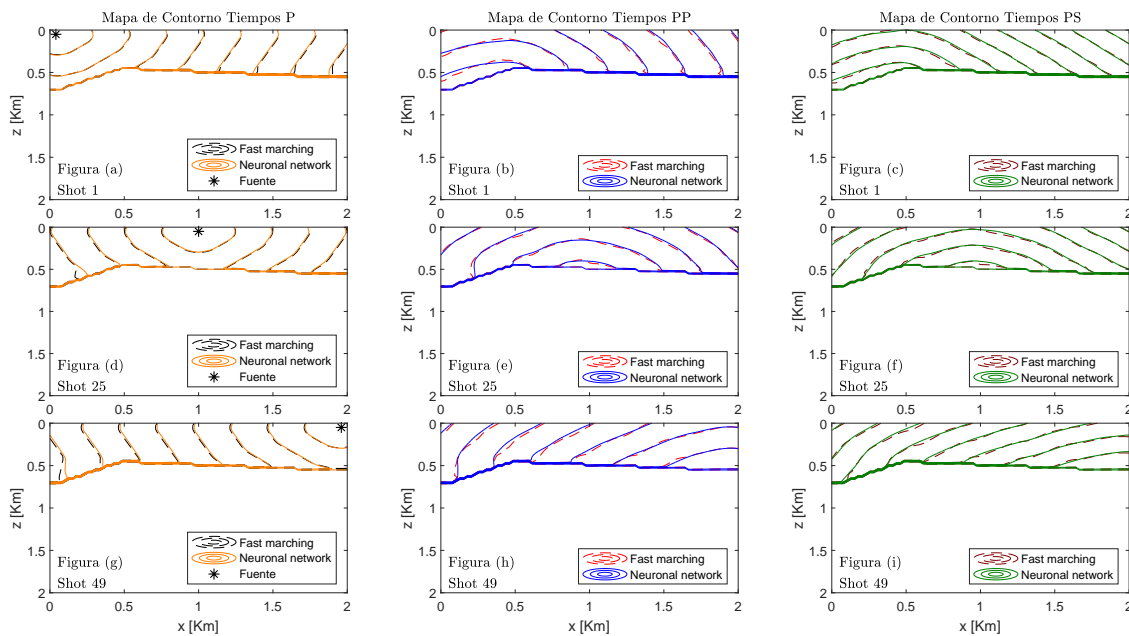


Figura 4.11: Mapa de contorno de modelo V con velocidad 5 km/s, columnas de tiempos P, PP y PS respectivamente, filas 1,2,3 posición del Shot₁ [50 ,40] m, Shot₂₅ [50 ,1000] m, Shot₄₉ [50 ,1960] m, respectivamente

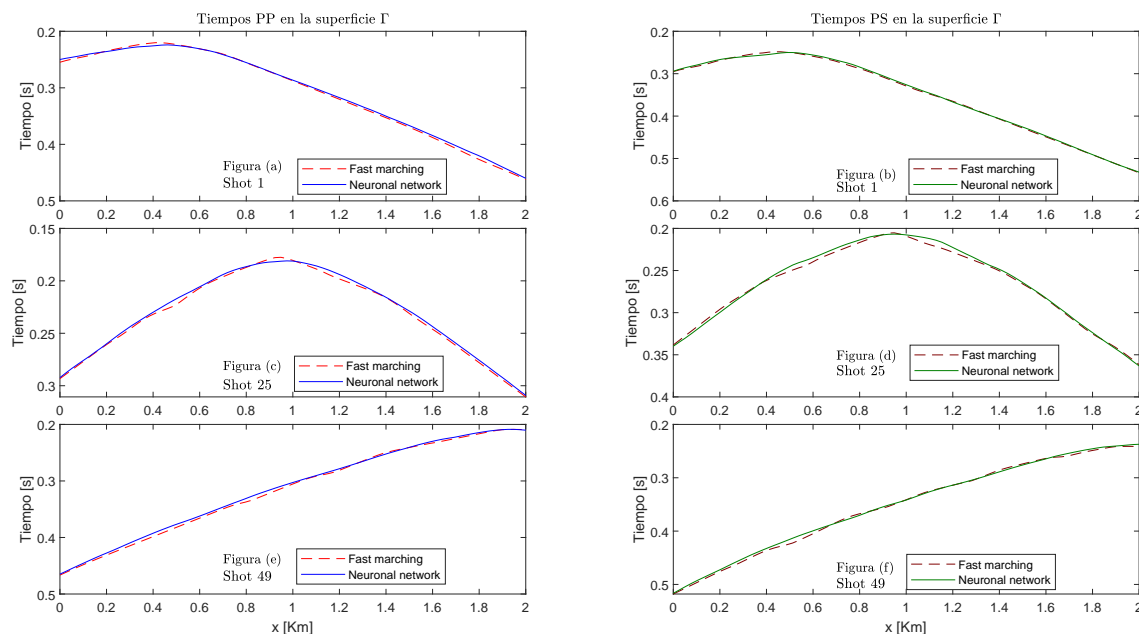


Figura 4.12: Comparativa de tiempos de viaje sobre el reflector Γ entre Fast marching y EIKO-4PS , columnas 1,2 tiempos PP y PS respectivamente,filas 1,2,3 posición del Shot₁ [50 ,40] m, Shot₂₅ [50 ,1000] m, Shot₄₉ [50 ,1960] m, respectivamente

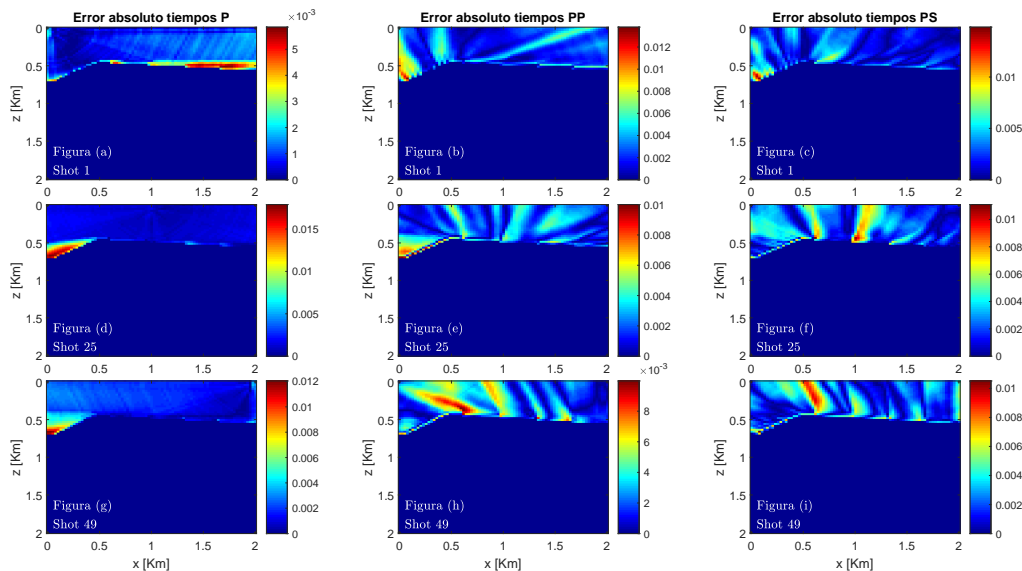


Figura 4.13: Errores absolutos del modelo V con velocidad de 5 km/s, columnas de tiempos P, PP y PS respectivamente, filas 1,2,3 posición del Shot₁ [50 ,40] m, Shot₂₅ [50 ,1000] m, Shot₄₉ [50 ,1960] m, respectivamente

4.2.2. *Mitad del modelo espacial*

En el modelo ubicado en la parte central de espacio predefinido se puede encontrar primero en la figura (4.14) el mapa de contorno que muestra la comparación de EIKO4 PS con Fast Marching. Para la figura (4.15) se encontraron los tiempos de viaje sobre la superficie Γ de los tiempos PP y PS entre la Red Neuronal y el modelo analítico. Por último, los errores asociados a cada uno de los mapas de contorno en la figura (4.16).

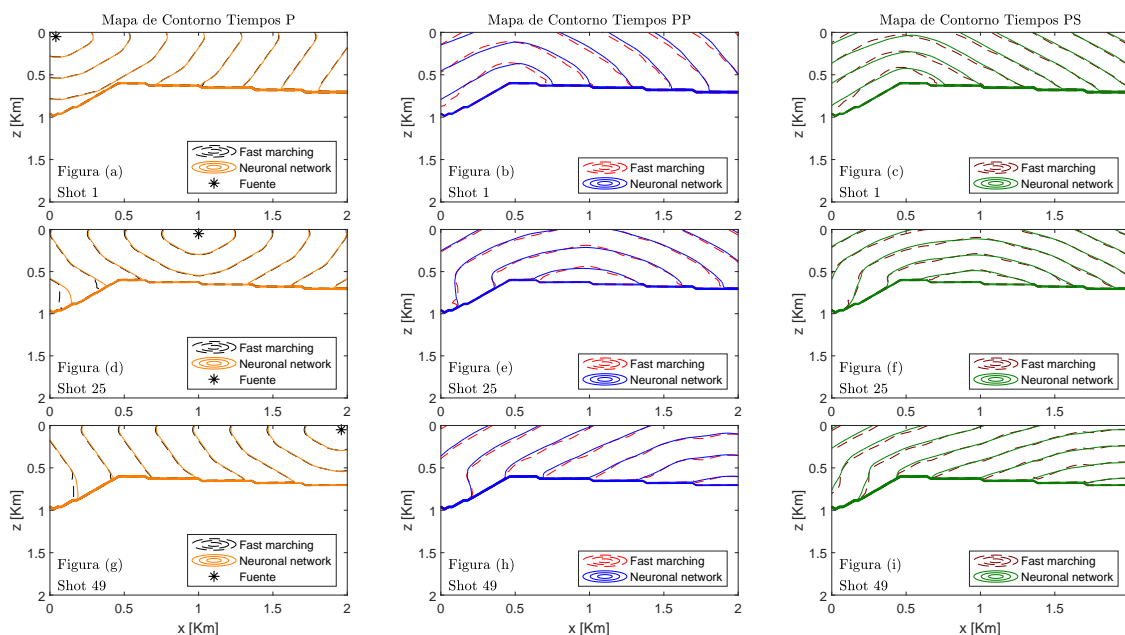


Figura 4.14: Mapa de contorno de modelo V sección media con velocidad 5km/s, columnas de tiempos P, PP y PS respectivamente, filas 1,2,3 posición del Shot₁ [50 ,40] m, Shot₂₅ [50 ,1000] m, Shot₄₉ [50 ,1960] m, respectivamente

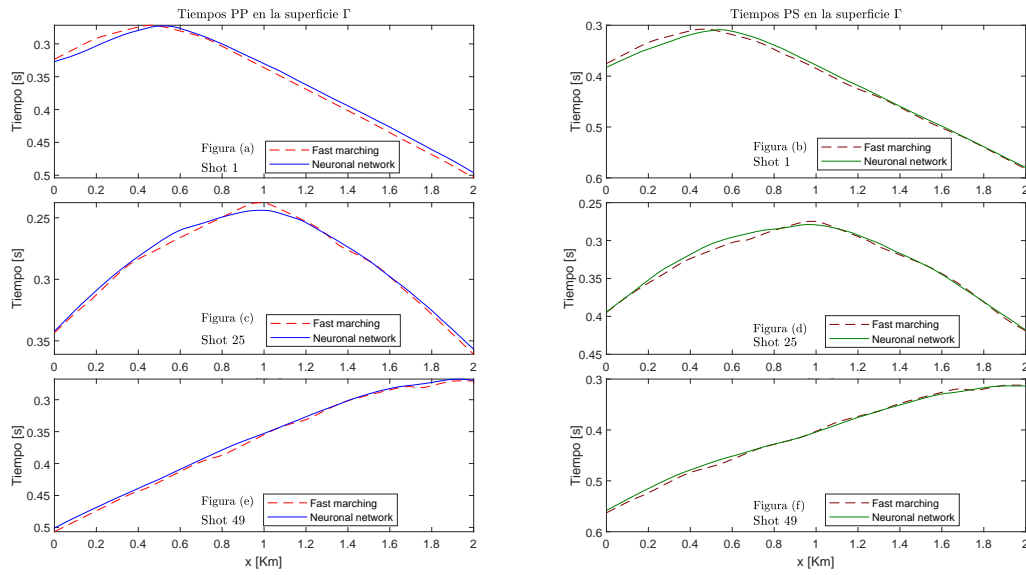


Figura 4.15: Comparativa de tiempos de viaje sobre el reflector Γ entre Fast marching y EIKO-4PS , columnas 1,2 tiempos PP y PS respectivamente,filas 1,2,3 posición del Shot₁ [50 ,40] m, Shot₂₅ [50 ,1000] m, Shot₄₉ [50 ,1960] m, respectivamente

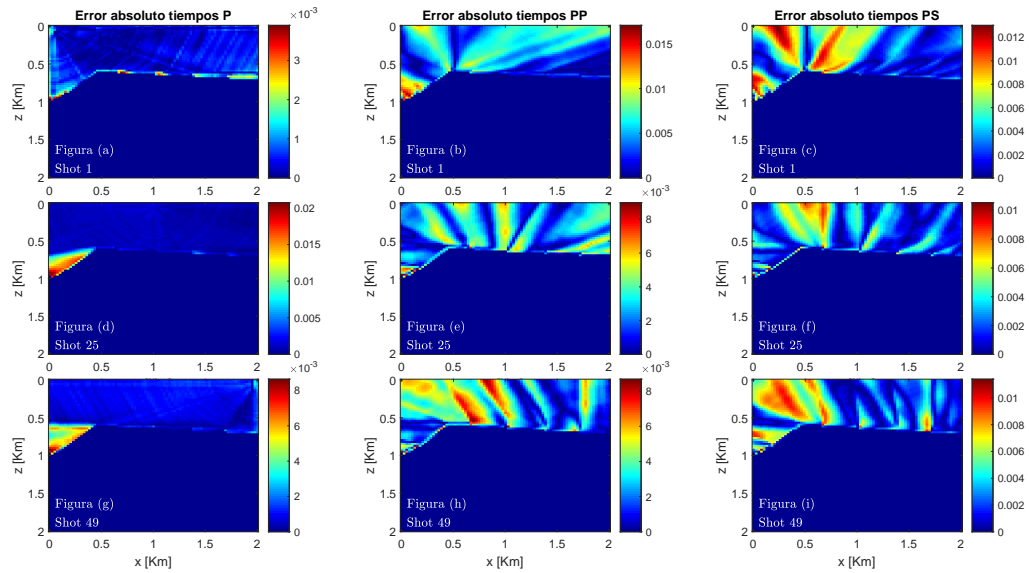


Figura 4.16: Errores absolutos del modelo V con velocidad de 5km/s, columnas de tiempos P, PP y PS respectivamente, filas 1,2,3 posición del Shot₁ [50 ,40] m, Shot₂₅ [50 ,1000] m, Shot₄₉ [50 ,1960] m, respectivamente

4.2.3. Lejano a la superficie

En el modelo mas alejado de la superficie se encuentra primero en la figura (4.17) el mapa de contorno que muestra la comparación de EIKO4 PS con Fast Marching. Para la figura (4.18) se encuentran los tiempos de viaje sobre la superficie Γ de los tiempos PP y PS entre la Red Neuronal y el modelo analítico. Por último, los errores asociados a cada uno de los mapas de contorno en la figura (4.19).

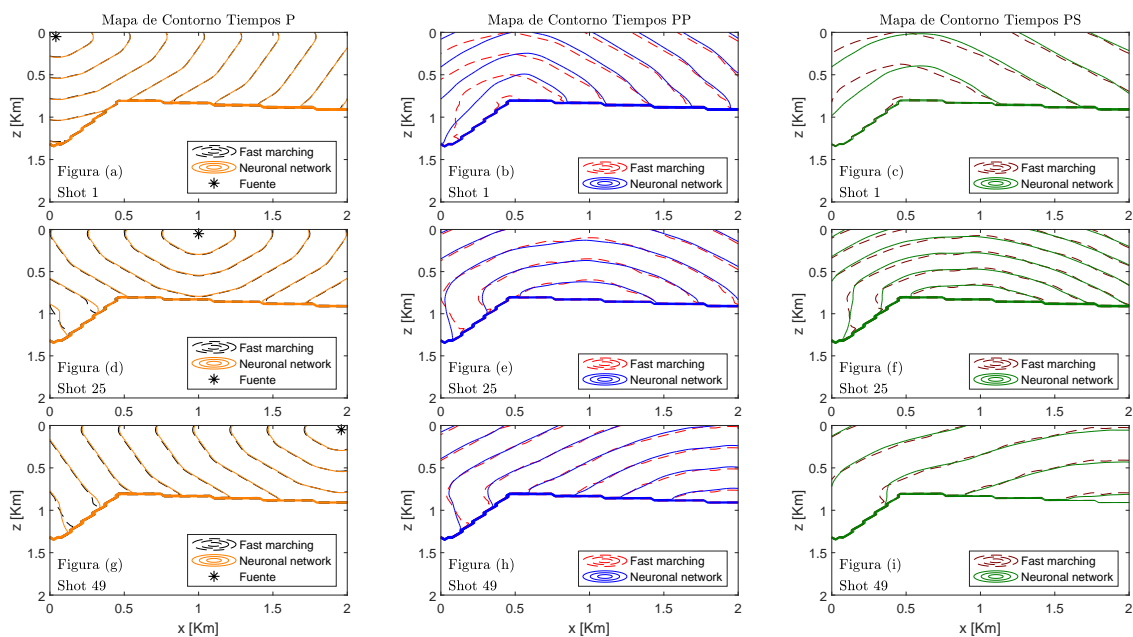


Figura 4.17: Mapa de contorno de modelo V de mayor profundidad con velocidad 5km/s, columnas de tiempos P, PP y PS respectivamente, filas 1,2,3 posición del Shot₁ [50 ,40] m, Shot₂₅ [50 ,1000] m, Shot₄₉ [50 ,1960] m, respectivamente

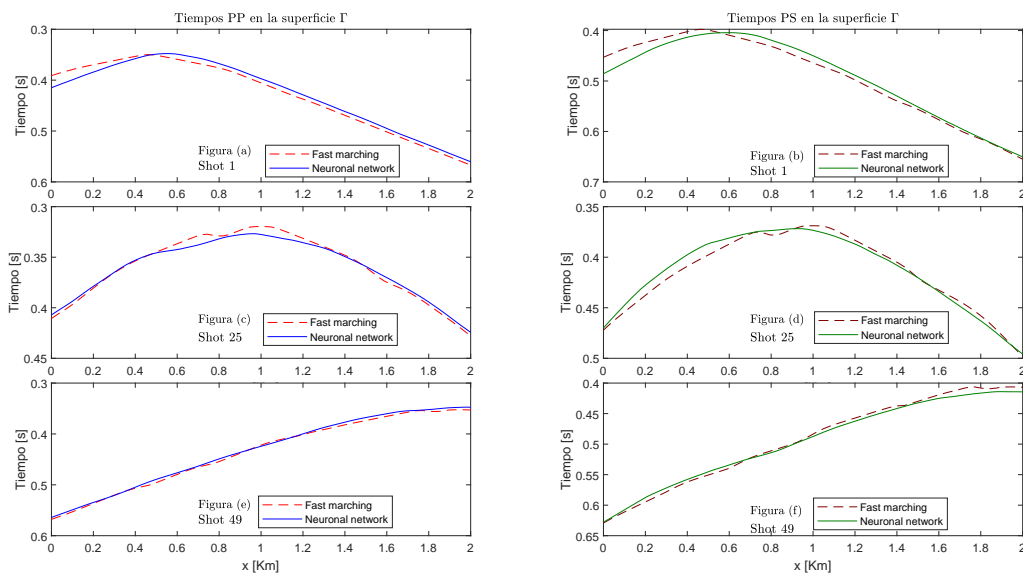


Figura 4.18: Comparativa de tiempos de viaje sobre el reflector Γ entre Fast marching y EIKO-4PS , columnas 1,2 tiempos PP y PS respectivamente,filas 1,2,3 posición del Shot₁ [50 ,40] m, Shot₂₅ [50 ,1000] m, Shot₄₉ [50 ,1960] m, respectivamente

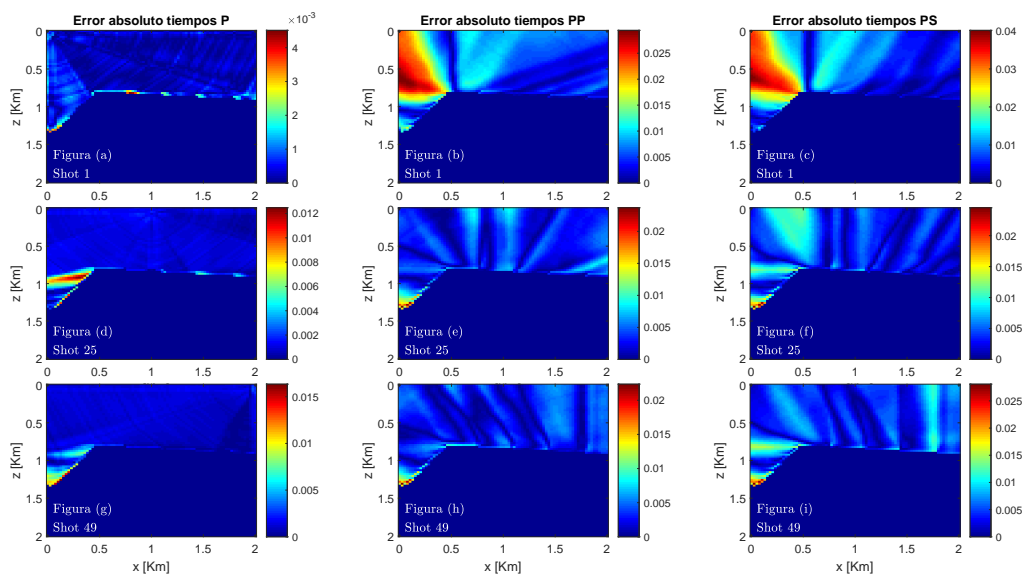


Figura 4.19: |Errores absolutos del modelo V con velocidad de 5 km/s, columnas de tiempos P, PP y PS respectivamente, filas 1,2,3 posición del Shot₁ [50 ,40] m, Shot₂₅ [50 ,1000] m, Shot₄₉ [50 ,1960] m, respectivamente

4.3. Modelo Cóncavo:

Las figuras (4.20) hasta la figura (4.28) detallan los resultados de las pruebas sobre un modelo en concavo invertida escalado a diferentes alturas: superficial, mitad de modelo espacial y por último el más alejado de la superficie. En estos se observan los mapas de contornos, los tiempos de viaje sobre la superficie Γ de los tiempos PP y PS, por último los errores asociados a cada uno de los mapas de contorno de cada uno de los modelos.

4.3.1. Cercano a la superficie

En el modelo más cercano a la superficie se encuentra en la figura (4.20) el mapa de contorno que muestra la comparación de EIKO4 PS con Fast Marching. Para la figura (4.21) se encuentran los tiempos de viaje sobre la superficie Γ de los tiempos PP y PS entre la Red Neuronal y el modelo analítico. Por último, los errores asociados a cada uno de los mapas de contorno en la figura (4.22).

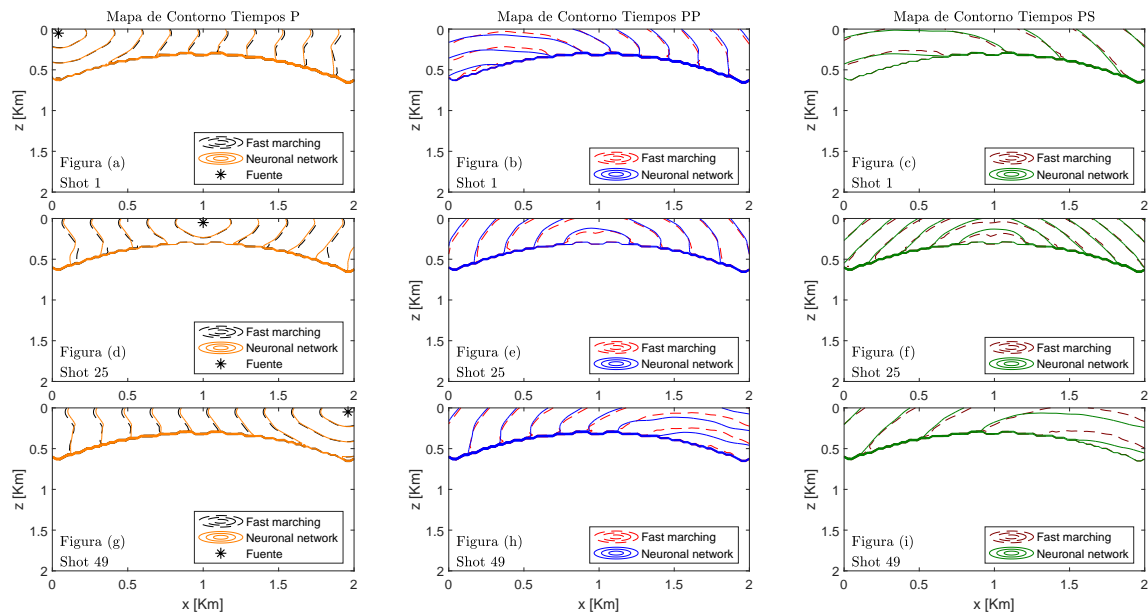


Figura 4.20: Mapa de contorno de modelo concavo con velocidad 3.75 km/s, columnas de tiempos P, PP y PS respectivamente, filas 1,2,3 posición del Shot₁ [50 ,40] m, Shot₂₅ [50 ,1000] m, Shot₄₉ [50 ,1960] m, respectivamente

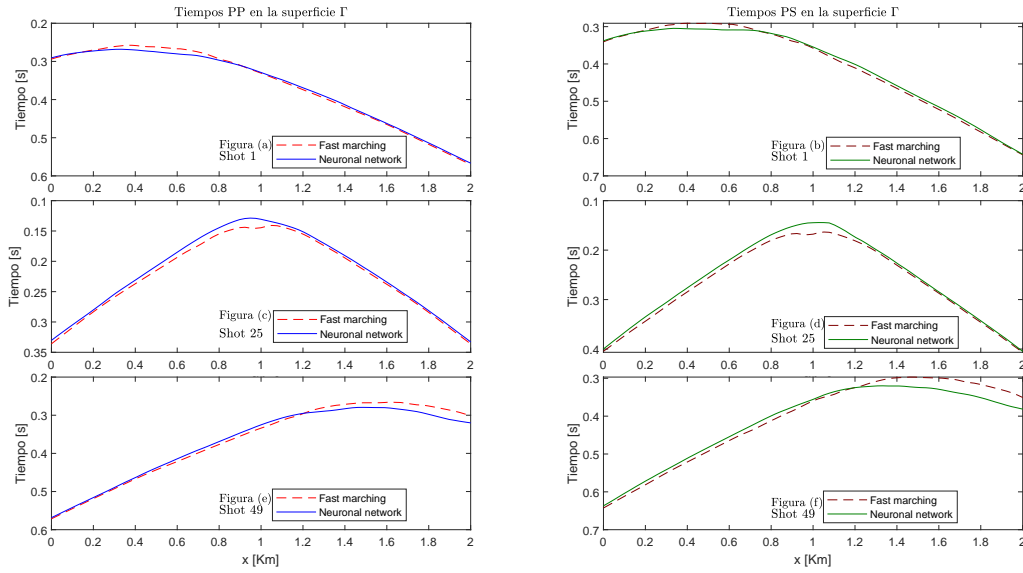


Figura 4.21: Comparativa de tiempos de viaje sobre el reflector Γ entre Fast marching y EIKO-4PS , columnas 1,2 tiempos PP y PS respectivamente,filas 1,2,3 posición del Shot₁ [50 ,40] m, Shot₂₅ [50 ,1000] m, Shot₄₉ [50 ,1960] m, respectivamente

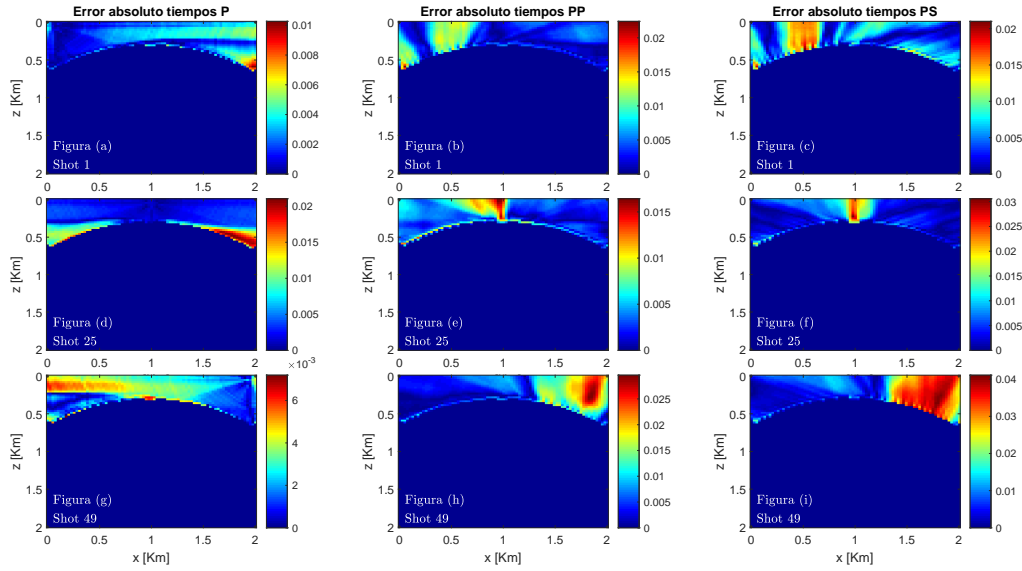


Figura 4.22: Errores absolutos del modelo cóncavo con velocidad de 3.75 km/s, columnas de tiempos P, PP y PS respectivamente, filas 1,2,3 posición del Shot₁ [50 ,40] m, Shot₂₅ [50 ,1000] m, Shot₄₉ [50 ,1960] m, respectivamente

4.3.2. *Mitad del modelo espacial*

En el modelo ubicado en la parte central del espacio predefinido se puede encontrar primero en la figura (4.23) el mapa de contorno que muestra la comparación de EIKO4 PS con Fast Marching. Para la figura (4.24) se encuentran los tiempos de viaje sobre la superficie Γ de los tiempos PP y PS entre la Red Neuronal y el modelo analítico. Por último, los errores asociados a cada uno de los mapas de contorno en la figura (4.25).

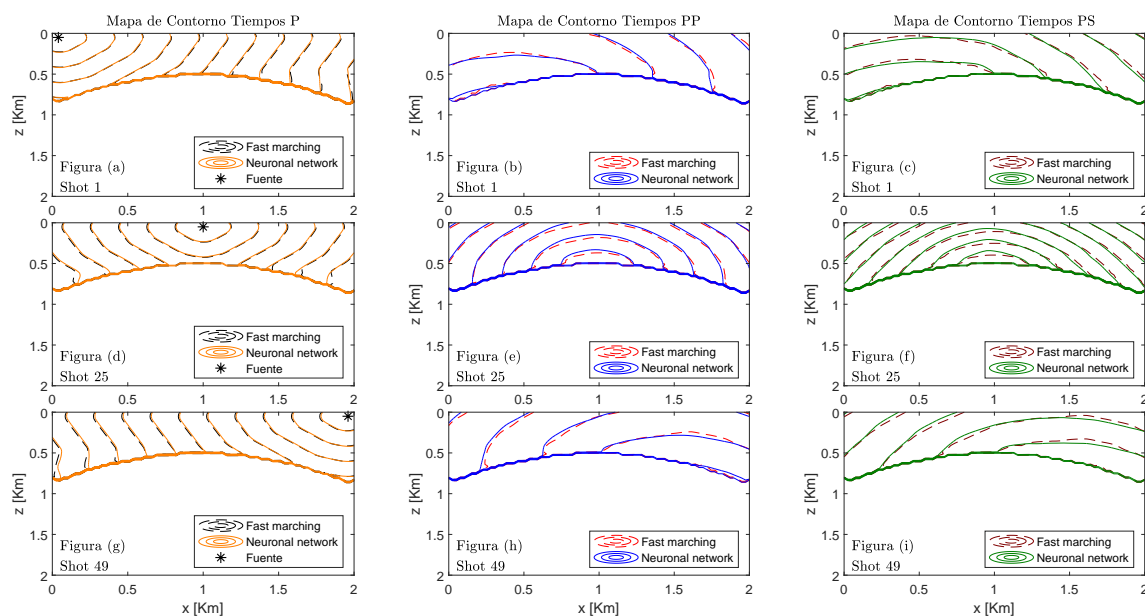


Figura 4.23: Mapa de contorno de modelo cóncavo sección media con velocidad 3.75 km/s, columnas de tiempos P, PP y PS respectivamente, filas 1,2,3 posición del Shot₁ [50 ,40] m, Shot₂₅ [50 ,1000] m, Shot₄₉ [50 ,1960] m, respectivamente

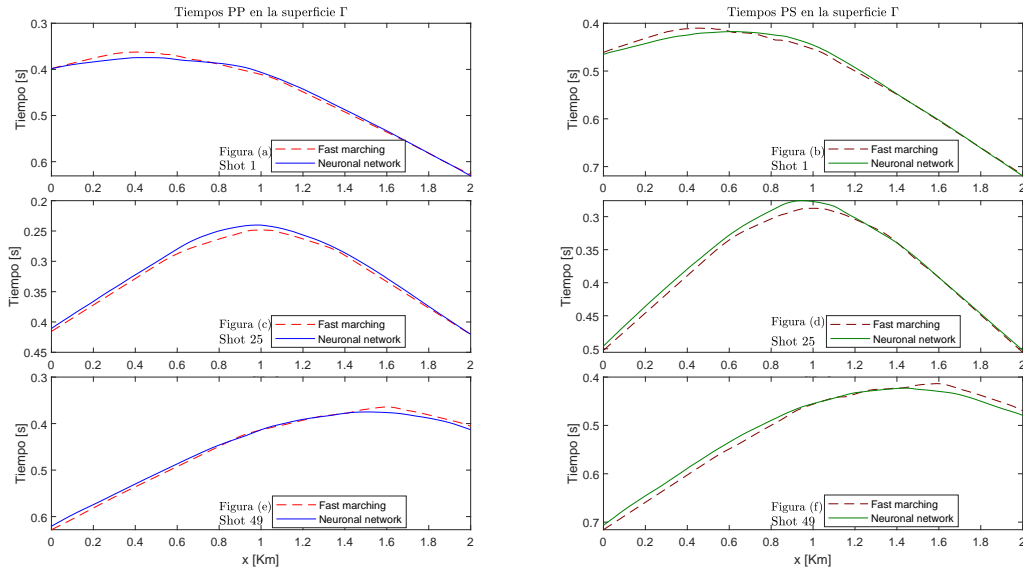


Figura 4.24: Comparativa de tiempos de viaje sobre el reflector Γ entre Fast marching y EIKO-4PS , columnas 1,2 tiempos PP y PS respectivamente,filas 1,2,3 posición del Shot₁ [50 ,40] m, Shot₂₅ [50 ,1000] m, Shot₄₉ [50 ,1960] m, respectivamente

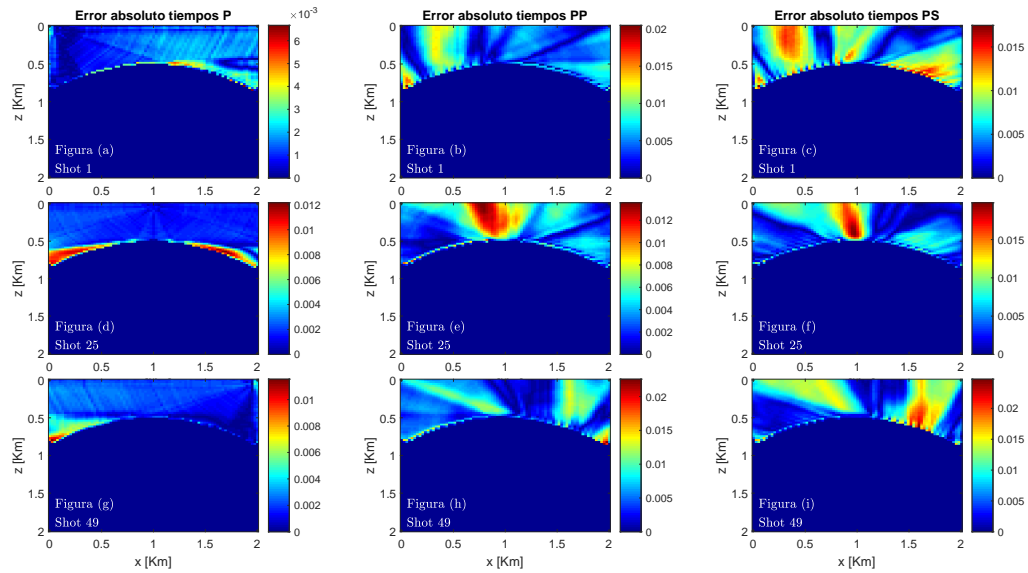


Figura 4.25: Errores absolutos del modelo cóncavo inverso con velocidad de 2.5 km/s, columnas de tiempos P, PP y PS respectivamente, filas 1,2,3 posición del Shot₁ [50 ,40] m, Shot₂₅ [50 ,1000] m, Shot₄₉ [50 ,1960] m, respectivamente

4.3.3. *Lejano a la superficie*

En el modelo mas alejado de la superficie se encuentra primero en la figura (4.26) el mapa de contorno que muestra la comparación de EIKO4 PS con Fast Marching. Para la figura (4.27) se encuentran los tiempos de viaje sobre la superficie Γ de los tiempos PP y PS entre la Red Neuronal y el modelo analítico. Por último, los errores asociados a cada uno de los mapas de contorno en la figura (??).

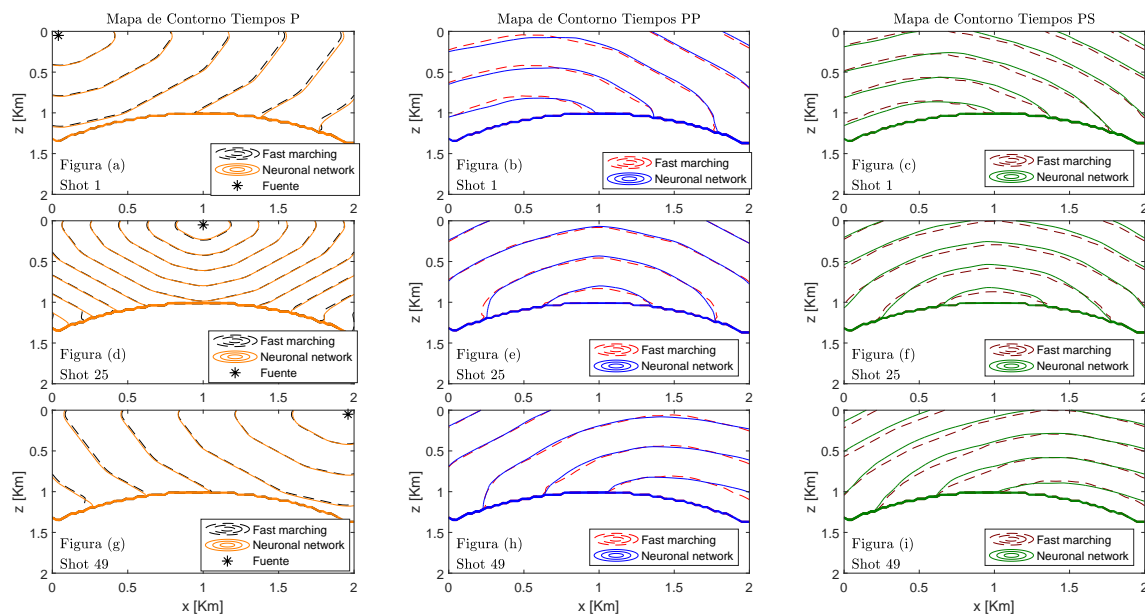


Figura 4.26: Mapa de contorno de modelo cóncavo de mayor profundidad con velocidad 3.75 km/s, columnas de tiempos P, PP y PS respectivamente, filas 1,2,3 posición del Shot₁ [50 ,40] m, Shot₂₅ [50 ,1000] m, Shot₄₉ [50 ,1960] m, respectivamente

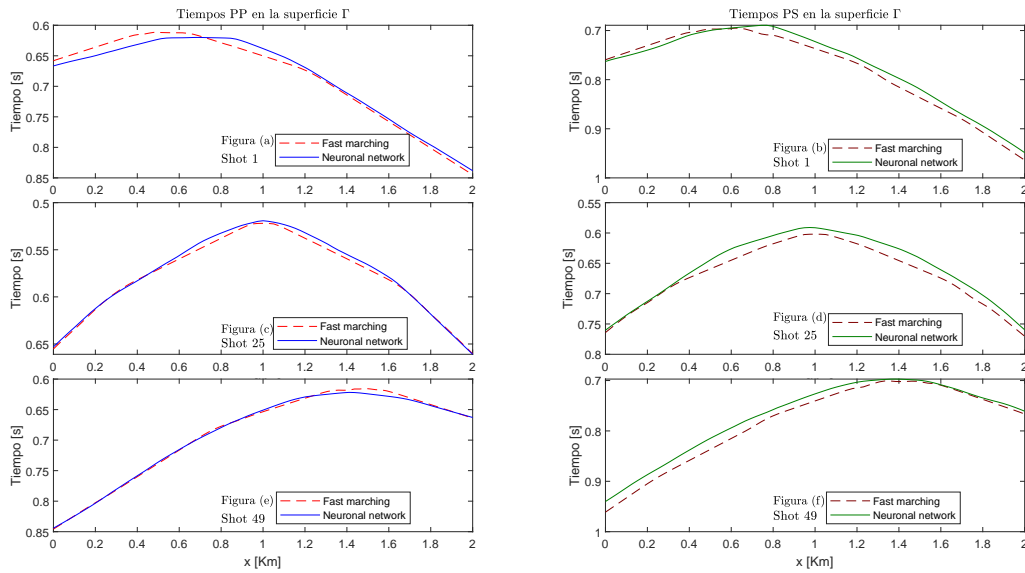


Figura 4.27: Comparativa de tiempos de viaje sobre el reflector Γ entre Fast marching y EIKO-4PS , columnas 1,2 tiempos PP y PS respectivamente,filas 1,2,3 posición del Shot₁ [50 ,40] m, Shot₂₅ [50 ,1000] m, Shot₄₉ [50 ,1960] m, respectivamente

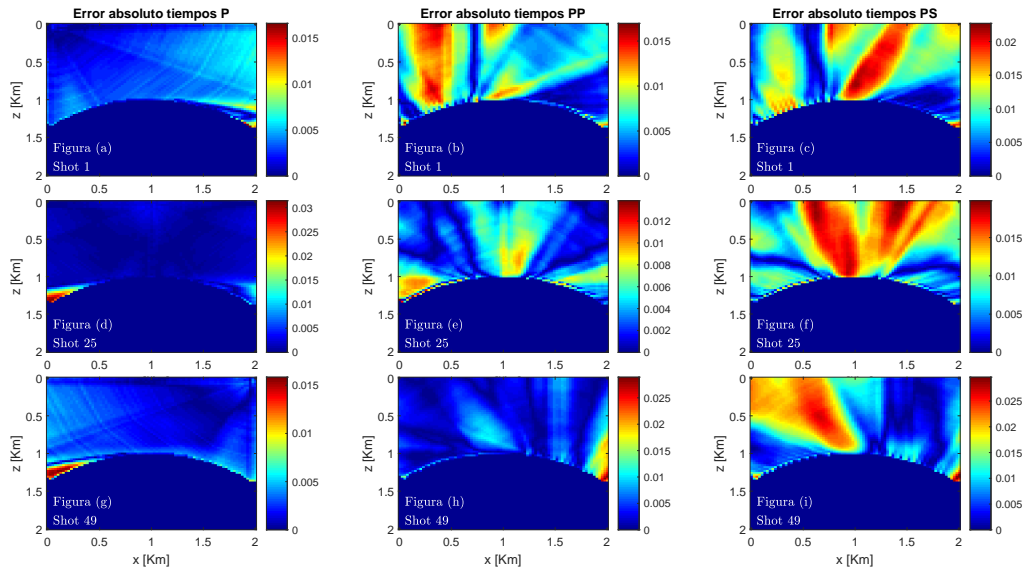


Figura 4.28: Errores absolutos del modelo cóncavo con velocidad de 3.75 km/s, columnas de tiempos P, PP y PS respectivamente, filas 1,2,3 posición del Shot₁ [50 ,40] m, Shot₂₅ [50 ,1000] m, Shot₄₉ [50 ,1960] m, respectivamente

4.4. Modelo Cóncavo inverso:

Las figuras (4.29) hasta la figura (4.37) detallan los resultados de las pruebas sobre un modelo en cóncavo escalado a diferentes alturas: superficial, mitad de modelo espacial y por último el más alejado de la superficie. En estos se observan los mapas de contornos, los tiempos de viaje sobre la superficie Γ de los tiempos PP y PS, por último los errores asociados a cada uno de los mapas de contorno de cada uno de los modelos.

4.4.1. Cercano a la superficie

En el modelo más cercano a la superficie se puede encontrar primero en la figura (4.29) el mapa de contorno que muestra la comparación de EIKO4 PS con Fast Marching. Para la figura (4.30) se encuentra los tiempos de viaje sobre la superficie Γ de los tiempos PP y PS entre la Red Neuronal y el modelo analítico. Por último, los errores asociados a cada uno de los mapas de contorno en la figura (4.31).

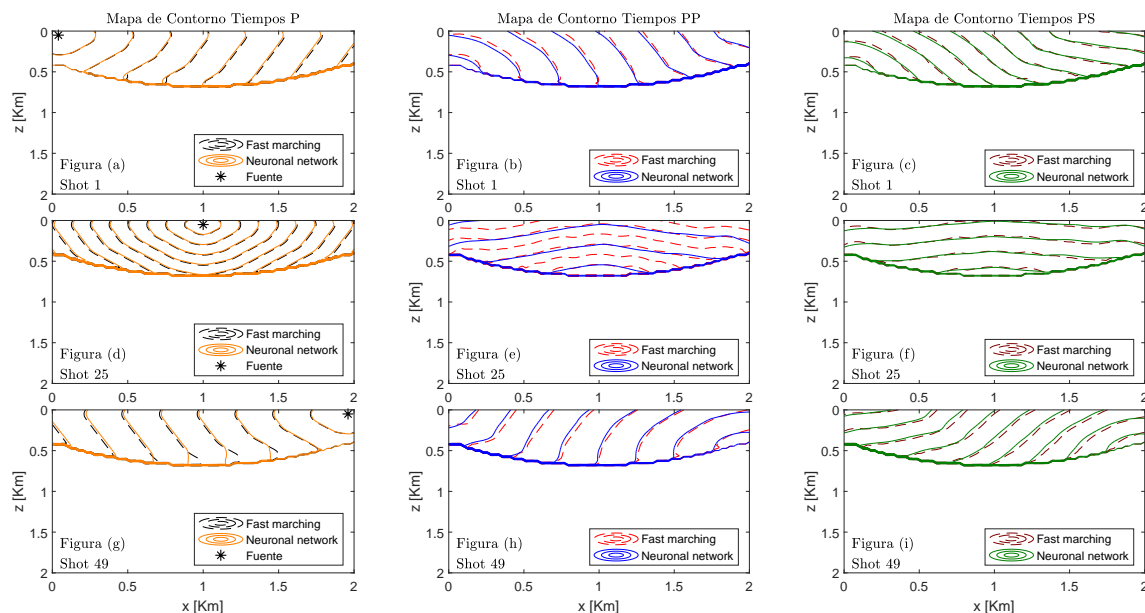


Figura 4.29: Mapa de contorno de modelo cóncavo inverso con velocidad 2.5 km/s, columnas de tiempos P, PP y PS respectivamente, filas 1,2,3 posición del Shot₁ [50 ,40] m, Shot₂₅ [50 ,1000] m, Shot₄₉ [50 ,1960] m, respectivamente

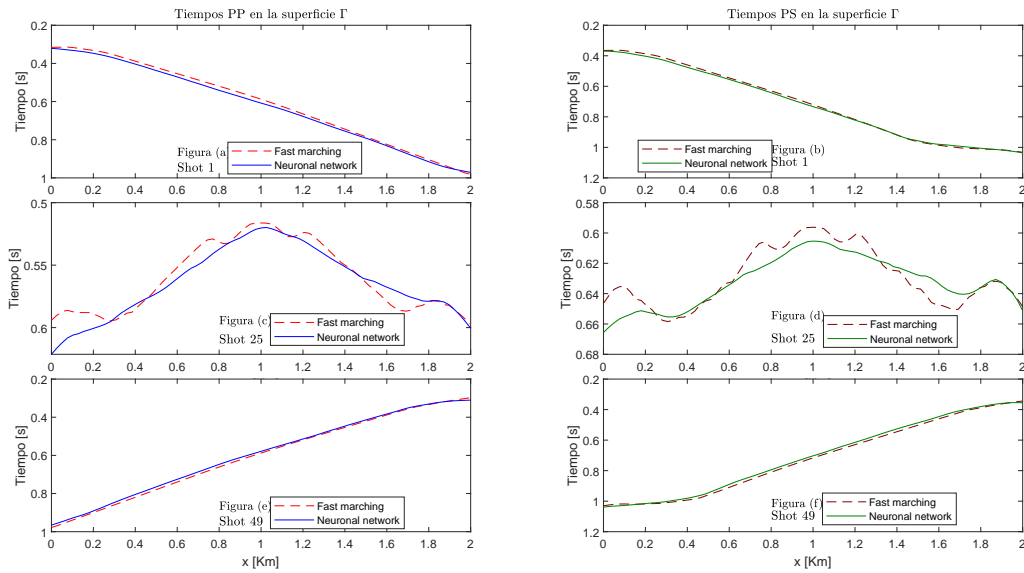


Figura 4.30: Comparativa de tiempos de viaje sobre el reflector Γ entre Fast marching y EIKO-4PS , columnas 1,2 tiempos PP y PS respectivamente, filas 1,2,3 posición del Shot₁ [50 ,40] m, Shot₂₅ [50 ,1000] m, Shot₄₉ [50 ,1960] m, respectivamente

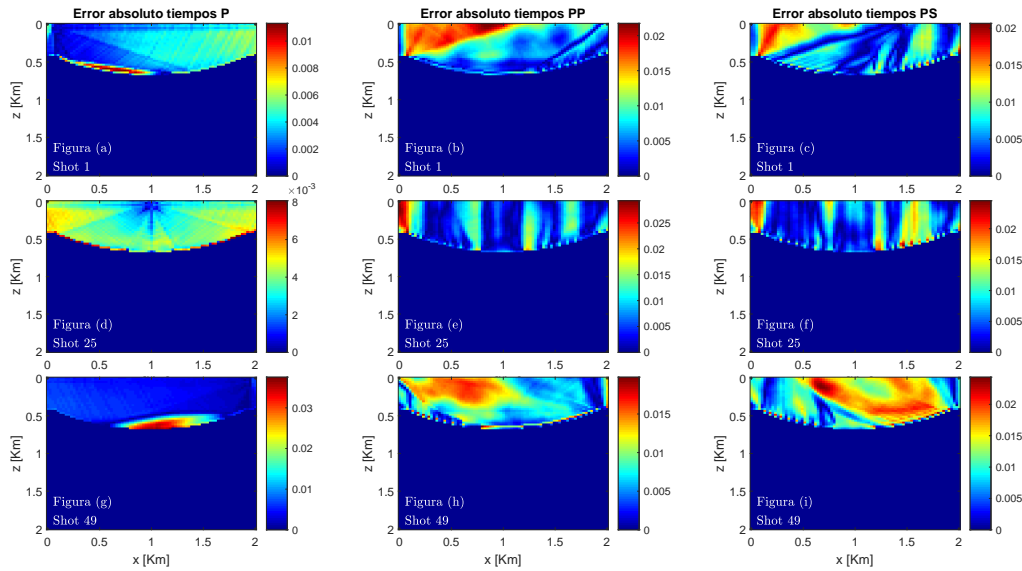


Figura 4.31: Errores absolutos del modelo cóncavo inverso con velocidad de 2.5 km/s, columnas de tiempos P, PP y PS respectivamente, filas 1,2,3 posición del Shot₁ [50 ,40] m, Shot₂₅ [50 ,1000] m, Shot₄₉ [50 ,1960] m, respectivamente

4.4.2. *Mitad del modelo espacial*

En el modelo ubicado en la parte central de espacio predefinido, se encuentra primero en la figura (4.32) el mapa de contorno que muestra la comparación de EIKO4 PS con Fast Marching. Para la figura (4.33) se encuentra los tiempos de viaje sobre la superficie Γ de los tiempos PP y PS entre la Red Neuronal y el modelo analítico. Por último, los errores asociados a cada uno de los mapas de contorno en la figura (4.34).

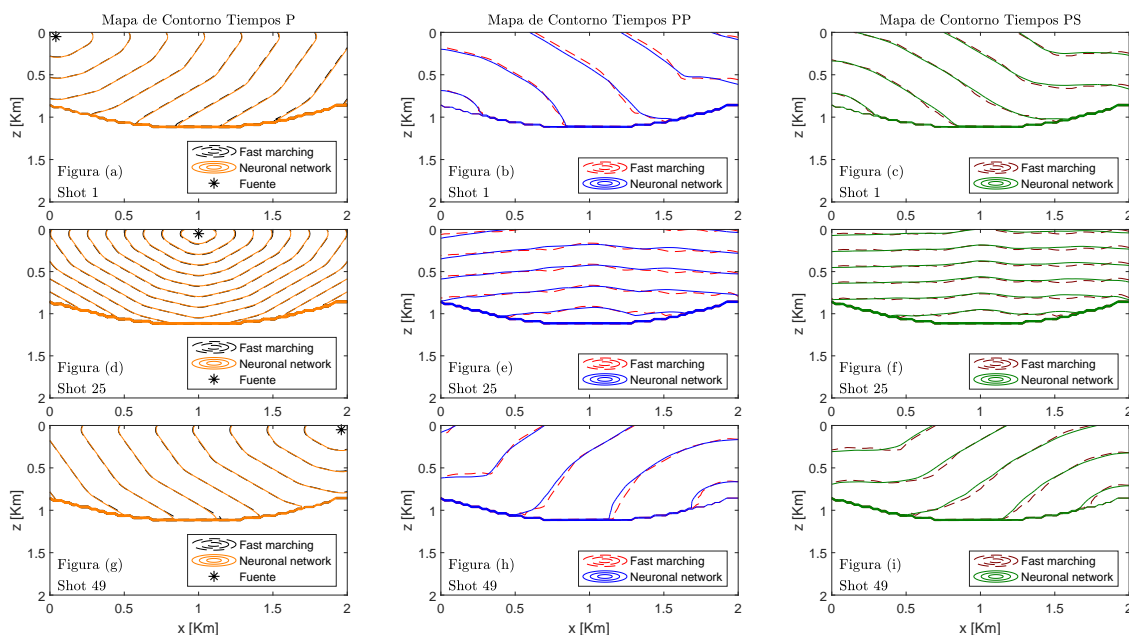


Figura 4.32: Mapa de contorno de modelo cóncavo inverso sección media con velocidad 2.5 km/s, columnas de tiempos P, PP y PS respectivamente, filas 1,2,3 posición del Shot₁ [50 ,40] m, Shot₂₅ [50 ,1000] m, Shot₄₉ [50 ,1960] m, respectivamente

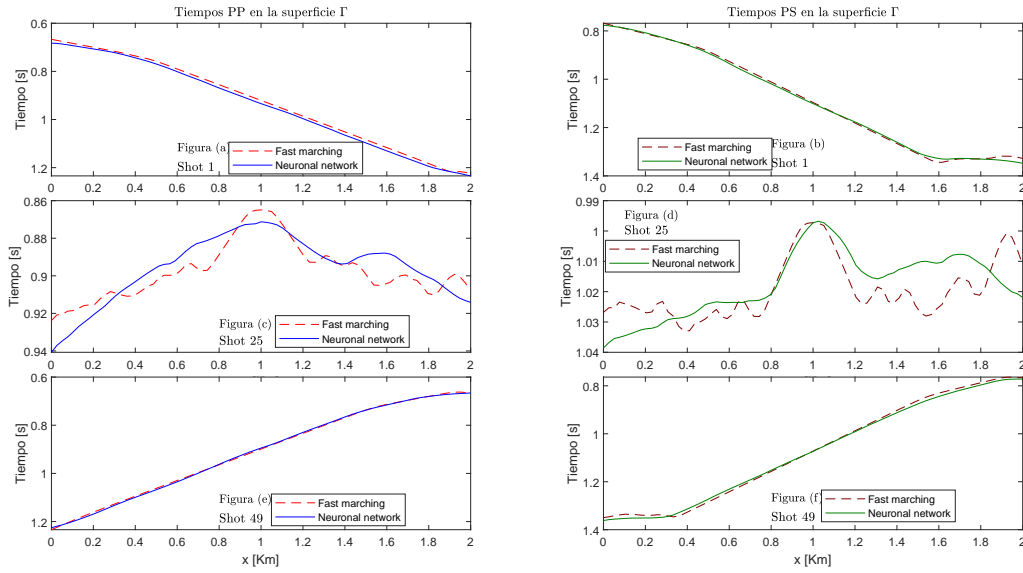


Figura 4.33: Comparativa de tiempos de viaje sobre el reflector Γ entre Fast marching y EIKO-4PS , columnas 1,2 tiempos PP y PS respectivamente, filas 1,2,3 posición del Shot₁ [50 ,40] m, Shot₂₅ [50 ,1000] m, Shot₄₉ [50 ,1960] m, respectivamente

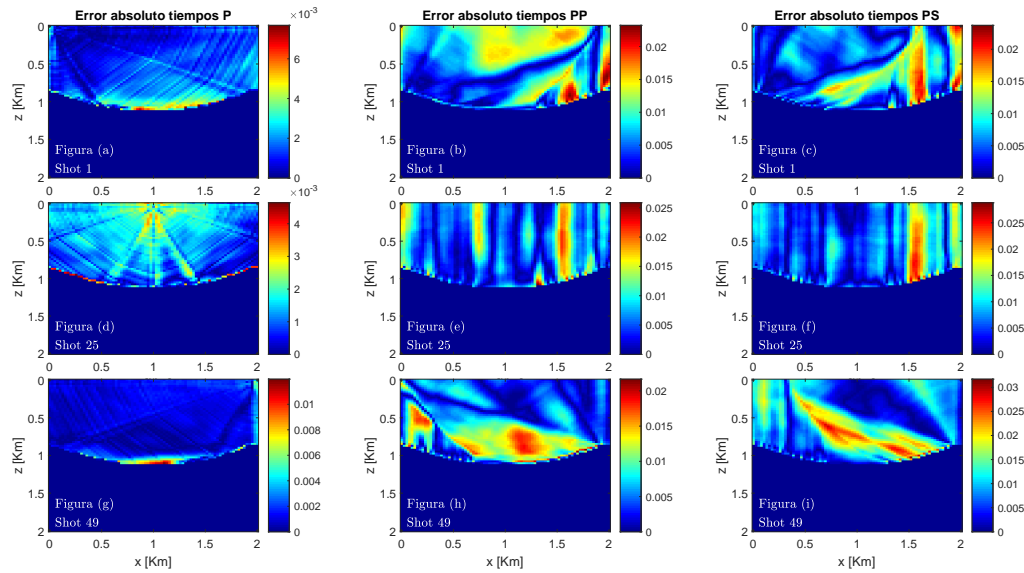


Figura 4.34: Errores absolutos del modelo cóncavo inverso con velocidad de 2.5 km/s, columnas de tiempos P, PP y PS respectivamente, filas 1,2,3 posición del Shot₁ [50 ,40] m, Shot₂₅ [50 ,1000] m, Shot₄₉ [50 ,1960] m, respectivamente

4.4.3. *Lejano a la superficie*

En el modelo mas alejado de la superficie se encuentra primero en la figura (4.35) el mapa de contorno que muestra la comparación de EIKO4 PS con Fast Marching. Para la figura (4.36) se encuentra los tiempos de viaje sobre la superficie Γ de los tiempos PP y PS entre la Red Neuronal y el modelo analítico. Por último, los errores asociados a cada uno de los mapas de contorno en la figura (4.37).

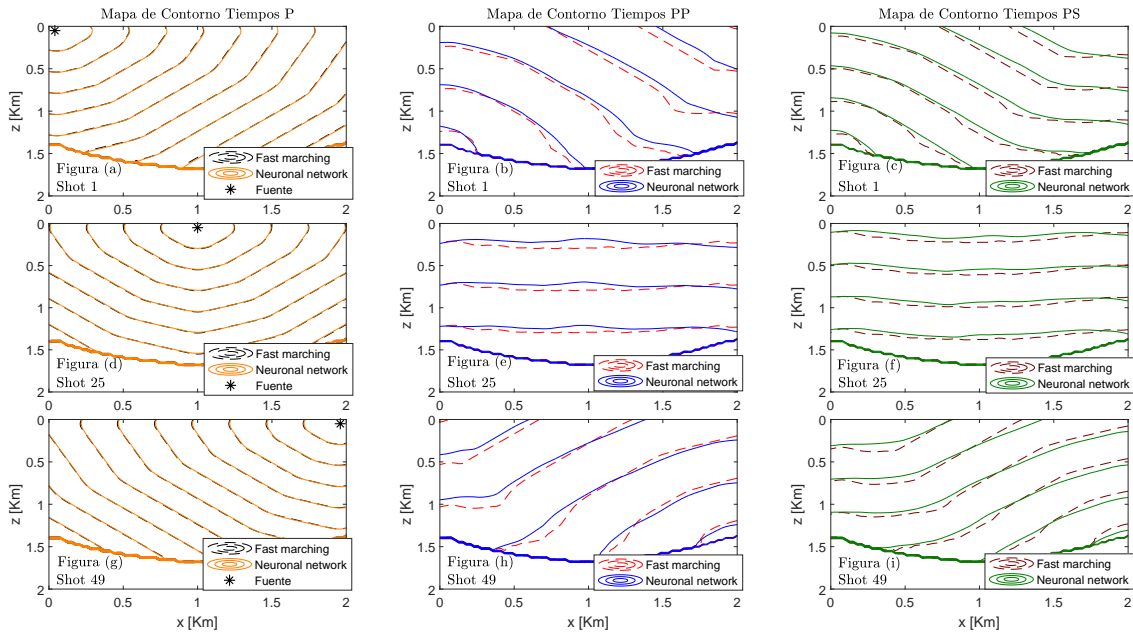


Figura 4.35: Mapa de contorno de modelo cóncavo inverso de mayor profundidad con velocidad 2.5 km/s, columnas de tiempos P, PP y PS respectivamente, filas 1,2,3 posición del Shot₁ [50 ,40] m, Shot₂₅ [50 ,1000] m, Shot₄₉ [50 ,1960] m, respectivamente

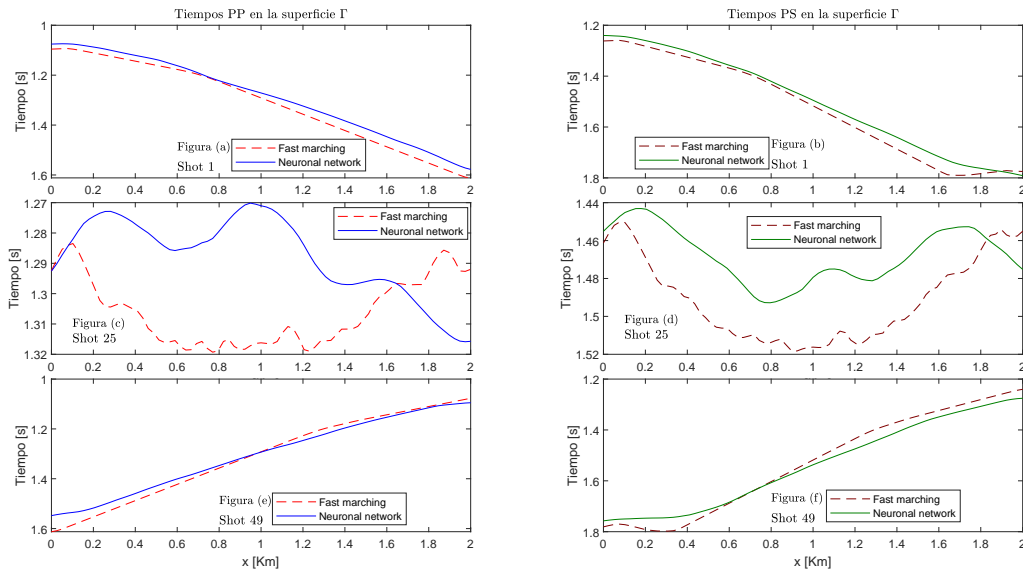


Figura 4.36: Comparativa de tiempos de viaje sobre el reflector Γ entre Fast marching y EIKO-4PS , columnas 1,2 tiempos PP y PS respectivamente, filas 1,2,3 posición del Shot₁ [50 ,40] m, Shot₂₅ [50 ,1000] m, Shot₄₉ [50 ,1960] m, respectivamente

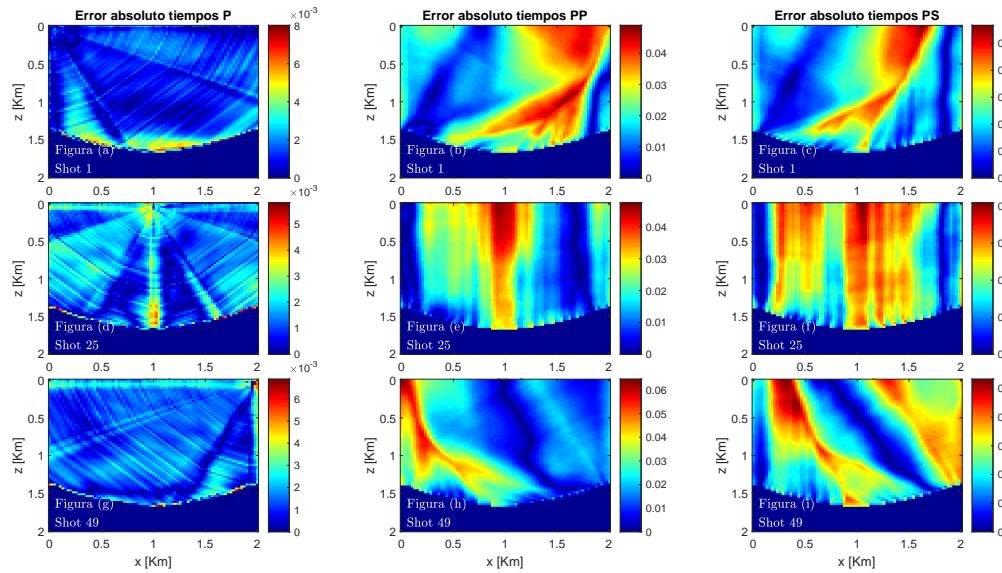


Figura 4.37: Errores absolutos del modelo cóncavo inverso con velocidad de 2.5 km/s, columnas de tiempos P, PP y PS respectivamente, filas 1,2,3 posición del Shot₁ [50 ,40] m, Shot₂₅ [50 ,1000] m, Shot₄₉ [50 ,1960] m, respectivamente

4.5. Análisis:

A partir de los resultados obtenidos con EIKO4PS en el cálculo de tiempos P, PP y PS utilizando redes neuronales, se observaron fenómenos que influyeron en la calidad de las predicciones. A continuación, se resumen los principales hallazgos:

Se encontró que las mejores predicciones corresponden a los tiempos P, independientemente de la geometría en comparación a los tiempo PP y PS, esto porque no presentaron errores inherentes significativos más allá de los provocados por la propia red, en su contra parte, los tiempos PP y PS mostraron errores más notables en comparación con los tiempos P, debido a la necesidad de calcularlos considerando la predicción de los tiempos P y el error inherente a esta predicción. Estos errores acumulativos generaron predicciones más dispares y por lo tanto menos precisas y alejadas del modelo analítico.

Los resultados obtenidos mostraron que los tiempos de ondas PP presentaron una menor desviación de alrededor 0.012 respecto a los datos observados que los tiempos de ondas PS, al compararlos sobre el mismo reflector y el mismo disparo. Esta diferencia se debe a la metodología empleada para el cálculo utilizado por el propio fast marching y la predicción de los tiempos, que busca un ajuste óptimo a los datos registrados, y confirma la mayor complejidad que implica el manejo de las ondas PS.

Se identificó que los errores en los tiempos PP y PS no seguían un patrón específico y variaron según la geometría. Las franjas de mayor error absoluto representada por franjas rojas; eran distintas en la misma geometría en diferentes alturas e incluso dentro de un mismo Shot_i no se repetía el patrón de la franja de errores, indicando una adaptación variable de la red a la complejidad del modelo. Esto se debe a que la red neuronal realiza un proceso de optimización que generaliza una función que predice los tiempos PP y PS, de tal forma que al comparar la predicción con los datos reales se observa como intenta ajustarse a ellos,

obtuvo entonces diferentes patrones dependiendo la geometría.

En el ámbito de las predicciones sísmicas, se observa que a pesar de la presencia de errores, la aplicación de métodos de optimización hace que EIKO4P-S se ajuste de manera eficaz a una variedad de geometrías, muchas de las cuales eran previamente desconocidas para el modelo. Es importante destacar que estas geometrías incluyen escenarios de mayor profundidad, los cuales no estaban representados en el conjunto de datos de entrenamiento. La prevalencia de errores en estas profundidades se debe a la ausencia de geometrías equivalentes en el conjunto de datos de entrenamiento. Esta discrepancia se explica por la falta de datos en el conjunto de entrenamiento que aborden la profundidad en cuestión. En el conjunto de datos utilizado, la profundidad más común de un reflector se encuentra en la sección media del modelo espacial, que abarca desde varios cientos de metros hasta algunos pocos kilómetros en la litosfera.

Este hallazgo tiene implicaciones significativas en aplicaciones prácticas, como la búsqueda de recursos naturales como hidrocarburos y carbón, donde la precisión de las predicciones es esencial.

En consecuencia, surge la cuestión de expandir el conjunto de datos de entrenamiento para incluir una gama más amplia y profunda de geometrías. Esta expansión podría conducir a mejoras sustanciales en la precisión de las predicciones en contextos de alta profundidad, lo que resultaría en avances significativos en la comprensión de los fenómenos geológicos en el subsuelo.

Los resultados ofrecen una visión clara de los desafíos y oportunidades en el campo de las predicciones sísmicas, destacando la importancia de la adaptación de las redes neuronales a condiciones desconocidas para avanzar en la investigación geofísica y la exploración

de recursos naturales.

Se destacó en el cálculo de tiempos PP y PS, que las geometrías ubicadas a profundidades mayores de 1.4 km, nunca vistas por la red neuronal durante el entrenamiento, generaron los mayores errores en las predicciones. Esto nos sugiere que el modelo no estaba familiarizado con estas configuraciones de geometría y, por ende, presenta dificultades para realizar las predicciones de los tiempos PP y PS de manera precisa. Es importante resaltar que en el desarrollo del proyecto estuvo limitado por los recursos disponibles (hardware, tiempo procesamiento) y en consecuencia una cantidad limitada de geometrías disponibles para el entrenamiento. Se considera con gran seguridad que la exposición de la red a una mayor diversidad de geometrías con formas y alturas diferentes durante el entrenamiento, resultaría en un mejor ajuste para geometrías que no ha visto previamente, obteniendo así una red con mayor capacidad de generalización.

Para finalizar se presentara un resumen de los resultados que se obtuvieron para cada uno de los modelos:




Geometría	Shot	Tiempos P				Tiempos PP				Tiempos PS			
		Error min	Error max	EIKO4P-S (en error max) [s]	Fast marching [s]	Error min	Error max	EIKO4P-S (en error max) [s]	Fast marching [s]	Error min	Error max	EIKO4P-S (en error max) [s]	Fast marching [s]
	1	1e-5	0.051	1.876	1.927	8 e-6	0.078	1.977	2.055	3e-6	0.09	1.994	2.0916
	25	1e-6	0.057	0.867	0.809	5 e-5	0.079	0.812	0.891	2e-5	0.09	0.872	0.966
	49	3e-5	0.063	1.123	1.059	2e-6	0.07	0.549	0.478	5e-5	0.08	1.892	1.807
	1	7e-6	0.096	1.546	1.643	2e-5	0.092	2.604	2.697	2e-5	0.11	2.204	2.321
	25	9e-6	0.040	1.545	1.586	2e-6	0.081	1.710	1.792	1e-6	0.080	2.820	2.901
	49	5e-5	0.054	1.170	1.224	8e-6	0.060	1.976	2.036	5e-5	0.075	1.203	1.279
	1	7e-5	0.159	1.779	1.939	0.014	0.227	3.172	3.399	0.003	0.296	3.810	4.107
	25	1e-5	0.081	1.671	1.752	0.01	0.17	2.953	3.131	1e-5	0.207	3.382	3.589
	49	3e-5	0.117	2.224	2.106	4e-5	0.17	3.459	3.632	3e-6	0.22	1.561	1.782

Figura 4.38: Resumen de resultados error absoluto para geometría Senoidal.




Geometría	Shot	Tiempos P				Tiempos PP				Tiempos PS			
		Error min	Error max	EIKO4P-S (en error max) [s]	Fast marching [s]	Error min	Error min	Error max	EIKO4P-S (en error max) [s]	Error min	Error max	Error min	Error max
	1	1e-6	0.005	0.364	0.358	1e-6	0.013	0.123	0.137	1e-6	0.014	0.123	0.138
	25	1e-6	0.017	0.239	0.221	5e-6	0.010	0.237	0.227	8e-7	0.011	0.125	0.136
	49	1e-6	0.012	0.418	0.406	1e-6	0.009	0.300	0.310	2e-7	0.010	0.399	0.410
	1	7e-7	0.003	0.197	0.193	1e-6	0.017	0.171	0.188	2e-7	0.013	0.342	0.329
	25	1e-5	0.020	0.277	0.256	6e-6	0.008	0.242	0.233	2e-6	0.010	0.143	0.153
	49	4e-7	0.008	0.433	0.425	4e-7	0.008	0.302	0.311	8e-8	0.011	0.417	0.429
	1	9e-8	0.004	0.247	0.243	5e-6	0.029	0.317	0.287	1e-5	0.040	0.361	0.321
	25	6e-6	0.012	0.252	0.240	3e-7	0.023	0.309	0.333	1e-6	0.024	0.310	0.334
	49	2e-7	0.016	0.448	0.465	4e-7	0.022	0.468	0.490	3e-6	0.028	0.463	0.491

Figura 4.39: Resumen de resultados error absoluto para geometría V.




Geometría	Shot	Tiempos P				Tiempos PP				Tiempos PS			
		Error min	Error max	EIKO4P-S (en error max) [s]	Fast marching [s]	Error min	Error min	Error max	EIKO4P-S (en error max) [s]	Error min	Error max	Error min	Error max
	1	2e-5	0.011	0.335	0.0324	1e-5	0.021	0.539	0.517	1e-5	0.020	0.853	0.832
	25	0.002	0.008	0.404	0.396	1e-6	0.029	0.572	0.542	7e-6	0.024	0.318	0.343
	49	4e-5	0.037	0.389	0.4267	1e-5	0.019	0.567	0.587	1e-5	0.024	0.823	0.848
	1	2e-6	0.007	0.742	0.749	5e-7	0.023	0.832	0.856	1e-7	0.023	0.809	0.832
	25	9e-6	0.004	0.535	0.530	8e-6	0.025	0.484	0.458	2e-6	0.028	0.490	0.518
	49	1e-6	0.011	0.606	0.618	1e-6	0.021	0.369	0.391	1e-5	0.031	0.451	0.483
	1	5e-8	0.008	0.909	0.917	1e-5	0.049	1.159	1.208	8e-6	0.056	1.730	1.787
	25	2e-6	0.005	0.736	0.730	8e-6	0.047	1.270	1.318	3e-5	0.040	1.462	1.503
	49	7e-7	0.006	0.007	0.014	3e-6	0.064	1.547	1.612	3e-6	0.053	1.592	1.646

Figura 4.40: Resumen de resultados error absoluto para geometría Cóncavo.




Geometría	Shot	Tiempos P				Tiempos PP				Tiempos PS			
		Error min	Error max	EIKO4P-S (en error max) [s]	Fast marching [s]	Error min	Error min	Error max	EIKO4P-S (en error max) [s]	Error min	Error max	Error min	Error max
	1	5e-6	0.010	0.537	0.527	2e-6	0.022	0.139	0.162	5e-6	0.021	0.187	0.166
	25	7e-5	0.021	0.255	0.234	1e-5	0.016	0.128	0.145	8e-7	0.030	0.066	0.097
	49	2e-5	0.007	0.484	0.491	1e-5	0.029	0.243	0.135	6e-5	0.041	0.184	0.143
	1	9e-7	0.006	0.522	0.529	1e-5	0.020	0.196	0.217	7e-7	0.017	0.201	0.219
	25	9e-5	0.012	0.270	0.258	2e-6	0.013	0.243	0.256	1e-6	0.019	0.125	0.145
	49	9e-6	0.011	0.557	0.545	1e-6	0.022	0.194	0.217	4e-5	0.024	0.224	0.200
	1	2e-6	0.016	0.634	0.650	1e-6	0.016	0.336	0.353	7e-6	0.022	0.628	0.650
	25	5e-7	0.031	0.467	0.436	3e-7	0.013	0.463	0.449	1e-5	0.019	0.618	0.638
	49	6e-6	0.015	0.635	0.620	4e-7	0.028	0.339	0.367	3e-6	0.029	0.333	0.362

Figura 4.41: Resumen de resultados error absoluto para geometría Cóncono Inverso.

5. Discusión

En las redes neuronales (NN) aplicadas a la geofísica, se encontraron tres enfoques que destacan por sus propias singularidades y potencialidades: EIKO 4PS, EIKONET y PINNeik. Estas metodologías desempeñan un papel esencial en la resolución de problemas relacionados con la reconstrucción de soluciones en geofísica. A continuación, se analizarán estas tres propuestas, destacando sus diferencias y similitudes.

Los modelos espaciales utilizados varían y son claramente delimitados en escala. EIKO 4PS trabaja sobre un modelo espacial de $2 \times 2 \text{ km}$, mientras que EIKONET maneja un modelo espacial tridimensional de $20 \times 20 \text{ km}$. En el caso de PINNeik, opera en un modelo de $6 \times 6 \text{ km}$.

Tanto EIKONET como PINNeik están sobre un modelo de velocidad heterogéneo durante todas las pruebas, mientras que el enfoque de EIKO4PS está direccionado a modelos de velocidad homogéneos, en donde los tiempos pueden ser calculados sobre un rango de velocidades comprendidas entre $[0.5; 5]$ tanto para las ondas P, PP Y PS.

El enfoque de reflector es una característica que distingue a EIKO 4PS de los otros dos enfoques. EIKO 4PS realiza el cálculo de tiempos desde la fuente a puntos aleatorios dentro del espacio delimitado por el reflector modelado con level set con $\phi(x, y) = 0$, lo que proporciona una visión detallada de las ondas sísmicas y su interacción con las estructuras geológicas. EIKONET y PINN, en cambio, los enfoques de estas dos redes esta en una grilla sin reflector lo cual corresponde al desplazamiento de las ondas sobre el medio de velocidad hasta el final del espacio predefinido, cabe resaltar que en el caso de la PINNeik mediante transfer learning se puede modelar una forma isométrica esto solo se puede para dicha geometría, no varía su forma y hay que reentrenar la red para dicha tarea, lo que significa un modelo que no generaliza dicho enfoque y solo lo mencionan como una posible mejora.

Una de las diferencias fundamentales entre estos enfoques radica en la ubicación de la fuente o disparo que emiten ondas de naturaleza P, cada modelo fue entrenado con múltiples posiciones de fuentes. Tal es el caso para, EIKO 4PS que opta por ubicaciones de una fuente fija a una profundidad de 50 m y espaciadas horizontalmente 40,81m para un total de 49 fuentes, situada en puntos predefinidos del espacio, lo que ofrece un control preciso sobre la ubicación de los datos generados. En contraste, EIKONET las fuentes se ubican en distintos puntos en el espacio aleatoriamente lo que permite una mayor versatilidad del modelo en el cálculo de los tiempos P, por último, PINNeik también emplea puntos fijos para la ubicación de la fuente, con 16 fuentes equiespaciadas horizontalmente y verticalmente 2km cada una en un modelo espacial de 6×6 km. Otro aspecto clave es la disposición de los receptores en el espacio. Tanto EIKO 4PS, EIKONET y PINNeik emplean receptores aleatorios, lo que les permite captar información de manera diversificada en diferentes ubicaciones espaciales. Esto debido a que se optimizan recursos, este método permite que los pares de fuente-receptores tengan una distribución de distancia que sea uniforme en todo el espacio del modelo (EIKONET)

La dimensión espacial es un factor esencial en estos enfoques ya que delimita el alcance de la solución. Tanto EIKO 4PS y PINNeik se especializan en problemas geofísicos en espacios bidimensionales (2D). En contraste, EIKONET opera en espacios tridimensionales (3D), lo que le permite modelar con mayor realismo situaciones físicas complejas.

El cálculo de tiempos es otra área de distinción. EIKO 4PS sobresale al calcular tiempos P, PP y PS, lo que proporciona una caracterización detallada de las ondas sísmicas. EIKONET y PINNeiko se enfocan principalmente en el cálculo de tiempos P, lo que puede ser adecuado en ciertos contextos geofísicos, aunque menos completo que el enfoque de EIKO 4PS.

Las épocas de entrenamiento son un aspecto crucial en el aprendizaje profundo. EIKO 4PS realiza 100 épocas de entrenamiento, mientras que EIKONET se extiende a 200 épocas. PINNeko por su parte supera a ambos, llevando a cabo un extenso entrenamiento de 2000 épocas, lo que puede influir en la convergencia y la calidad del modelo resultante.

El hardware utilizado también difiere. EIKO 4PS utiliza la Tesla T4 GPU Nvidia, mientras que EIKONET aprovecha el rendimiento de una Tesla V100 GPU de Nvidia, conocida por su potencia. En contraposición, PINN se ejecuta en una Tesla P100 GPU de Nvidia, que, aunque efectiva, puede ser ligeramente menos rápida que la V100. Esto fue un factor importante ya que en EIKO4PS el hardware usado limitó la cantidad de pruebas y de geometrías diferentes que se puedan realizar en el enfoque de EIKO4PS, por lo que se vio limitado a tan solo 100 diferentes geometrías y 20 modelos de velocidad homogéneos comprendido por un vector $[0.5,5]$ equiespaciados 0.225 Km, lo que implica que la red tiene capacidad de mejora en su precisión al calcular los tiempos P, PP,y PS para muchas más formas de reflector y un vector más grande de velocidades y así abarcar un espectro del problemas más allá de lo que se pudo abordar.

Para el caso particular del cálculo de los tiempos P la redes EIKONET Y PINNeik tienen errores absolutos muy bajos alrededor de 0.008, en comparación al enfoque de EIKO 4PS donde se presentan errores absolutos de hasta 0.08, esto debido a que el enfoque abordado en este documento está en la generalización del cálculo de los tiempos para diferentes geometrías y diferentes shots lo que que implica una mayor espectro del problema y no tan específico como lo hacen los enfoques de las otras NN, además de que este enfoque corresponde a una etapa exploratoria del problema, por lo cual resulta un reto comparar estos resultados ya que a pesar de que calculan lo mismo no lo hacen bajo las mismas condiciones.

6. Conclusiones

El modelo presentado estima los tiempos de primeros arribos con errores máximos de hasta el 10 % en términos de error absoluto para geometrías con profundidades de hasta 1,2 km o en la zona intermedia del modelo espacial, como se ilustra en el mapa de colores de errores. Sin embargo, en general, más del 80 % de estos errores se mantuvieron por debajo del 5 %, lo que indica que la red logró adaptarse adecuadamente a este tipo de geometrías y demostró su capacidad de generalización.

En contraste, se observó un menor ajuste en geometrías con profundidades mayores a 1.2 km como se muestra en los mapas de colores de errores. Para las geometrías de forma V, cóncavas y cóncavas inversas, el error máximo aumentó en un 2 % con respecto a las geometrías de profundidad cercana e intermedia de la superficie. Los resultados menos favorables se obtuvieron en el caso de la superficie senoidal de mayor profundidad, con errores que alcanzaron hasta un 30 %, debido a la cantidad de cambios en la propia forma sinusoidal, y no es coincidencia que a medida que aumenta la complejidad del modelo, también lo haga el valor de los errores presentes en el modelo. Esto se confirma a través del mapa de colores, donde los errores más significativos cubren entre el 40 % y el 50 % de la superficie del modelo, lo que indica un ajuste menos preciso de la red, aproximadamente un 20 % menos en comparación con los modelos más superficiales y menos complejos.

Los resultados revelan que EIKO4P-S, se muestra menos eficiente en términos de tiempo de procesamiento en el caso particular de geometrías más profundas, llegando a experimentar demoras hasta 3 veces mayores en la obtención de los tiempos PP y PS en comparación con el algoritmo de (C.A.Niño et al., 2022), mientras que para los tiempos P las demoras son hasta 6 veces mayores. Para mejorar esta métrica se sugiere el uso de un mejor hardware con un GPU con mayor poder de procesamiento ya que para el modelo de EIKO4PS es fundamental el uso de GPU para un cálculo más rápido, mientras que el modelo analítico realiza el cálculo por medio de CPU.

Los resultados revelan que EIKO4P-S, para geometrías superficiales o con menos

puntos en el espacio Ω_1 es más eficiente en términos de tiempo de procesamiento, ya que es 17 veces más rápida que fast marching en el cálculo de tiempos P y 32 veces más rápida en el cálculo de los tiempos PP y PS.

Es importante resaltar que el proyecto puede escalarse, ya que, a lo largo de su desarrollo, enfrentó limitaciones de hardware que restringieron tanto el tamaño del conjunto de datos de entrenamiento como el diseño y entrenamiento de la red neuronal. Este hecho se evidenció claramente en las pruebas realizadas, donde el rendimiento y la capacidad de generalización de la red mejora con el aumento de la base de datos. Este proyecto, por lo tanto, sienta las bases para futuras investigaciones y desarrollos que podrían beneficiarse de recursos más robustos, con el potencial de optimizar y perfeccionar aún más la aplicación de redes neuronales en el cálculo preciso de tiempos P, PP y PS.

En conclusión, se observa una distinción esencial entre el algoritmo propuesto (C.A.Niño et al., 2022) y EIKO4P-S. En contraposición al enfoque de la solución analítica, la red no requiere la modelización del reflector mediante el método de level set ni la fijación de $\phi = 0$ una vez que el modelo ha sido debidamente entrenado. En su lugar, la red únicamente demanda la información relativa a la ubicación en profundidad (z), la velocidad del modelo, la ubicación de la fuente y la de los puntos sobre los cuales se quiere predecir los tiempos, en otras palabras todos los tiempos desde y por encima del reflector como dato de entrada, lo que le permite realizar los cálculos punto por punto de manera eficaz. Con esta metodología, se consigue reconstruir la imagen de los tiempos desde la fuente hasta el reflector en el caso de los eventos P, así como desde el receptor en la superficie para los tiempos PP y PS utilizando los tiempos predichos sobre el reflector en la red anterior.

La solución analítica tradicional precisa que el algoritmo efectúa un proceso de ajuste de contorno de nivel para identificar el reflector, y posteriormente emplea $\phi = 0$ como frontera entre los medios para calcular los tiempos únicamente en Ω_1 mediante el método de Fast Marching, este procedimiento conlleva un incremento sustancial en el tiempo requerido para completar todos los cálculos. El tiempo de cálculo del algoritmo de ajuste de contorno

se encuentra supeditado a factores como la intervención manual necesaria, la configuración de la geometría y la profundidad de la misma, así como el número de iteraciones necesario para detectar el reflector; en la mayoría de las ocasiones, este escenario conlleva un tiempo significativamente mayor, frecuentemente superando los 30 minutos o más, dependiendo de la complejidad y profundidad de la geometría considerada.

En comparación, el enfoque de este trabajo basado en redes neuronales proporciona una alternativa eficiente y con una menor dependencia de ajustes manuales en comparación con la metodología analítica previa. Este enfoque se traduce en tiempos de cálculo más breves y en una mayor adaptabilidad a diversas situaciones de modelización.

7. Recomendaciones

Durante la ejecución de este proyecto, se logró identificar que el incremento en el tamaño de la base de datos fue importante en la capacidad de las redes neuronales para generalizar los datos de manera más efectiva, resultando en mejoras de los resultados obtenidos. Por consiguiente, una recomendación es ampliar la base de datos, incorporando geometrías variadas y ubicadas a profundidades mayores a 1.2 km. Esta recomendación tiene el propósito de mitigar las limitaciones identificadas en este proyecto, como lo es calcular los tiempos para geometrías ubicadas a profundidades mayores a 1.4 km en el modelo espacial.

Se recomienda contar con una mayor capacidad de procesamiento a través de un hardware que no limite el proceso, se sugiere intentar con una GPU V100 con un procesador de 14th generación de intel y 128 Gb de RAM lo cual permitiría la implementación de redes neuronales más profundas y también posibilitará el manejo de bases de datos más extensas. Además, permitiría llevar a cabo entrenamientos más extensos y de mayor calidad para la red neuronal, influyendo positivamente en su capacidad de aprendizaje y generalización.

En el presente proyecto, se abordaron las imágenes de tiempos P, PP y PS con una resolución de 79×79 píxeles. No obstante, se sugiere explorar posibilidades de avance para proyectos futuros mediante la utilización de imágenes con resoluciones distintas a las empleadas en este estudio. Una estrategia viable para lograr esto sería implementar una etapa de preprocesamiento en las imágenes mediante interpolación que permita mayor adquisición de datos y posteriormente un muestreo que se ajuste a la resolución específica desarrollada en este proyecto.

Como sugerencia para futuros avances, se propone la creación de bases de datos de manera similar a la metodología utilizada en este trabajo, pero adaptándolas a diferentes tamaños de resolución que se deseen explorar. Es importante destacar que esta propuesta conlleva a tener que ajustar el tamaño de entrada de las redes neuronales, lo cual debe ser considerado cuidadosamente en la planificación y ejecución de proyectos subsiguientes.

Referencias Bibliográficas

- Aghasi, A., Kilmer, M., & Miller, E. L. (2011). Parametric Level Set Methods for Inverse Problems. *SIAM Journal on Imaging Sciences*, 4(2), 618-650. <https://doi.org/10.1137/100800208>
- Bashir, D., Montañez, G. D., Sehra, S., Segura, P. S., & Lauw, J. (2020). An Information-Theoretic Perspective on Overfitting and Underfitting. *CoRR*, *abs/2010.06076*. <https://arxiv.org/abs/2010.06076>
- Bellman, R. (1978). *An Introduction to Artificial Intelligence: Can Computers Think?* Boyd & Fraser Publishing Company. <https://books.google.com.co/books?id=84xQAAAAMAAJ>
- Bin Waheed, U., Haghigat, E., Alkhalifah, T., Song, C., & Hao, Q. (2020). Eikonal Solution Using Physics-Informed Neural Networks. <https://doi.org/10.3997/2214-4609.202011041>
- C.A.Niño, C.Duarte, Agudelo, W., Sierra, D., & Sacchi, M. (2022). CONVERTED WAVE TRAVEL TIME COMPUTATION USING A FAST MARCHING METHOD AND LEVEL SET INTERFACE LOCATION. <https://geoconvention.com/wp-content/uploads/abstracts/2022/73309-converted-wave-travel-time-computation-using-a-fas.pdf>
- Capozzoli, A., Curcio, C., Liseno, A., & Savarese, S. (2013). A comparison of Fast Marching, Fast Sweeping and Fast Iterative Methods for the solution of the eikonal equation. *2013 21st Telecommunications Forum Telfor (TELFOR)*, 685-688. <https://doi.org/10.1109/TELFOR.2013.6716321>
- Chen, D., Hu, F., Nian, G., & Yang, T. (2020). Deep Residual Learning for Nonlinear Regression. *Entropy*, 22(2). <https://doi.org/10.3390/e22020193>
- Clevert, D.-A., Unterthiner, T., & Hochreiter, S. (2016). Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs).

- Genç, B., & Tunc, H. (2019). Optimal training and test sets design for machine learning. *TURKISH JOURNAL OF ELECTRICAL ENGINEERING COMPUTER SCIENCES*, 27, 1534-1545. <https://doi.org/10.3906/elk-1807-212>
- Geron, A. (2017). *Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems* (1st). O'Reilly Media, Inc.
- Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning* [<http://www.deeplearningbook.org>]. MIT Press.
- Hassoun, M. H. (1995). *Fundamentals of Artificial Neural Networks* (1st). MIT Press.
- He, K., Zhang, X., Ren, S., & Sun, J. (2015). Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. *CoRR*, *abs/1502.01852*. <http://arxiv.org/abs/1502.01852>
- He, K., Zhang, X., Ren, S., & Sun, J. (2016). Identity Mappings in Deep Residual Networks. *CoRR*, *abs/1603.05027*. <http://arxiv.org/abs/1603.05027>
- Jadon, A., Patil, A., & Jadon, S. (2022). A Comprehensive Survey of Regression Based Loss Functions for Time Series Forecasting.
- Janiesch, C., Zschech, P., & Heinrich, K. (2021). Aprendizaje automático y aprendizaje profundo. *Mercados de electrones*, 31, 685-695. <https://doi.org/10.1007/s12525-021-00475-2>
- Kim, B. J., Choi, H., Jang, H., Lee, D., & Kim, S. W. (2023). How to Use Dropout Correctly on Residual Networks with Batch Normalization.
- Kingma, D. P., & Ba, J. (2017). Adam: A Method for Stochastic Optimization.
- Li, W., Leung, S., & Qian, J. (2014). A level-set adjoint-state method for crosswell transmission-reflection travelttime tomography. *Geophysical Journal International*, 199(1), 348-367. <https://doi.org/10.1093/gji/ggu262>
- Mantilla Parra, J. J., & Vargas Blanco, M. R. (2023). *Detección automática de primeros arribos usando redes neuronales artificiales en trazas terrestres reales del catálogo sísmico colombiano*. <https://noesis.uis.edu.co/handle/20.500.14071/12362>

- Mescua, J., Giambiagi, L., & Spagnotto, S. (2021). Información sobre eventos sísmicos no destructivos como una oportunidad para la alfabetización científica. *Journal of Science Communication América Latina*, 04. <https://doi.org/10.22323/3.04010801>
- Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., & Riedmiller, M. (2013). Playing Atari with Deep Reinforcement Learning. *NIPS Deep Learning Workshop*, 1-9. <https://doi.org/10.48550/arXiv.1312.5602>
- Prieto, K., & Dorn, O. (2016). Sparsity and level set regularization for diffuse optical tomography using a transport model in 2D. *Inverse Problems*, 33(1), 014001. <https://doi.org/10.1088/0266-5611/33/1/014001>
- Quituisaca-Samaniego, L., & Álvarez, H. (2017). Aprendizaje no supervisado: agrupamiento o clustering. <https://doi.org/10.13140/RG.2.2.19303.34724>
- Rawlinson, N., & Sambridge, M. (2005). The Fast Marching Method: An Effective Tool for Tomographic Imaging and Tracking Multiple Phases in Complex Layered Media. *Exploration Geophysics*, 36(4), 341-350. <https://doi.org/10.1071/EG05341>
- Reynolds, J. M. (2011). *An Introduction to Applied and Environmental Geophysics* [Pages 33-40]. Preview. <https://doi.org/10.1071/PVv2011n155other>
- Rodríguez P., A. I., & Buitrago R., X. D. P. (2022). How to choose an activation function for deep learning. *Tekhnê*, 19(1), 23-32. <https://revistas.udistrital.edu.co/index.php/tekhne/article/view/20337>
- Rodríguez-Tapia, S., & Camacho-Cañamón, J. (2018). Los métodos de aprendizaje automático supervisado en la clasificación textual según el grado de especialización. *Tonos Digital*, 2018.
- Sanchez, P., & Velaquez, J. (2011). EL ROL DEL ALGORITMO DE ENTRENAMIENTO EN LA SELECCION DE MODELOS DE REDES NEURONALES. *Revista U.D.C.A Actualidad Divulgación Científica*, 14, 149-156. http://www.scielo.org.co/scielo.php?script=sci_arttext&pid=S0123-42262011000100018&nrm=iso

- Sepúlveda-Jaimes, F. J., & Cabrera-Zambrano, F. H. (2018). Tomografía sísmica 3D del nido sísmico de Bucaramanga (Colombia). *Boletín de Geología*, 40(2), 15-33. <https://doi.org/10.18273/revbol.v40n2-2018001>
- Smith, J. (2021). Physics-Informed Neural Networks for Inverse Problems. *Journal of Computational Physics*, 200(1), 100-120. <https://doi.org/10.1016/j.jcp.2020.109067>
- Smith, J. D., Azizzadenesheli, K., & Ross, Z. E. (2021). EikoNet: Solving the Eikonal Equation With Deep Neural Networks. *IEEE Transactions on Geoscience and Remote Sensing*, 59(12), 10685-10696. <https://doi.org/10.1109/TGRS.2020.3039165>
- Zeneli, M., Nikolopoulos, A., Karellas, S., & Nikolopoulos, N. (2021). Chapter 7 - Numerical methods for solid-liquid phase-change problems. En A. Datas (Ed.), *Ultra-High Temperature Thermal Energy Storage, Transfer and Conversion* (pp. 165-199). Woodhead Publishing. <https://doi.org/https://doi.org/10.1016/B978-0-12-819955-8.00007-7>
- Zheglova, P., & Farquharson, C. (2012). Level set method in seismic inversion: 2D reconstruction of boundaries. En *SEG Technical Program Expanded Abstracts 2012* (pp. 1-6). <https://doi.org/10.1190/segam2012-0959.1>
- Zuluaga Herrera, C. D., & Muñoz Cuartas, J. C. (2020). Refracciones y reflexiones simultáneas en estimación de velocidades mediante tomografía basada en rayos. *Revista EIA*, 17(33), 1-15. <https://doi.org/10.24050/reia.v17i33.1321>

Apendices

Repositorios: En el siguiente enlace se encuentra todos los programas, archivos usados para el desarrollo de este proyecto:

<https://drive.google.com/drive/folders/13ipDSQ6khwbw7uM-YAA-KFHTclGGyu0P?usp=sharing>