UNIVERSIDADE FEDERAL DA BAHIA
Curso de Pós-graduação em Geofísica

DISSERTAÇÃO DE MESTRADO

INVERSÃO DA FORMA DE ONDA COMPLETA COM ABORDAGEM MULTIESCALA USANDO REDES NEURAIS RECORRENTES

JORGE LUÍS DOS SANTOS SANTANA

SALVADOR – BAHIA

Documento preparado com o sistema LaTeX.

Documento elaborado com os recursos gráficos e de informática do CPGG/UFBA

Inversão da Forma de Onda Completa com Abordagem Multiescala Usando Redes Neurais Recorrentes

por

Jorge Luís dos Santos Santana

Geofísico (UFBA – 2007)

Orientador: Prof. Dr. Reynam da Cruz Pestana

DISSERTAÇÃO DE MESTRADO

Submetida em satisfação parcial dos requisitos ao grau de

MESTRE EM CIÊNCIAS

EM

GEOFÍSICA

ao

Conselho Acadêmico de Ensino

da

Universidade Federal da Bahia

Comissão Examinadora

Dr. Reynam da Cruz Pestana

Dr. Alexsandro Guerra Cerqueira

Dr. Oscar Fabian Mojica Ladino

Aprovada em 19 de agosto de 2022

A presente pesquisa foi desenvolvida no Centro de Pesquisa em Geofísica e Geologia da UFBA, com recursos próprios e da PETROBRAS.

Santana, Jorge Luís dos Santos,
Inversão da Forma de Onda Completa com Abordagem Multiescala Usando Redes Neurais Recorrentes / Jorge Luís dos Santos Santana. — Salvador, 2022.
97 f.: il.

Orientador: Prof. Dr. Reynam da Cruz Pestana
Dissertação (Mestrado) - de Pós-Graduação em Geofísica. Instituto de Geociências da Universidade Federal da Bahia, 2022.

1. Inversão Geofísica. 2. Inversão Sísmica. 3. Velocidade Sísmica. 4. Redes Neurais. 5. Inteligência Artificial. I. Pestana, Reynam da Cruz. II. Título

A Jusciane, Joana e Júlia, meus amores.

Resumo

Vista como parâmetro fundamental para uma confiável imagem geológica da subsuperfície, e consequente sucesso exploratório, a velocidade sísmica é um dos pré-requisitos na cadeia da exploração sísmica. Tal velocidade tem relação direta na qualidade da construção de imagens sísmicas através de algoritmos robustos como o da migração reversa no tempo (em inglês, *Reverse Time Migration* - RTM) ou outras técnicas de imageamento sísmico. Neste trabalho, recorremos aos ambientes de aprendizado de máquina para obtermos uma velocidade sísmica de alta resolução por meio da técnica de inversão da forma de onda completa (em inglês, *Full Waveform Inversion* - FWI).

Em síntese, a técnica FWI visa comparar dados de observações reais com dados calculados obtidos através da modelagem sísmica a partir da solução de uma equação de onda específica. O resíduo dessa comparação é minimizado e, o gradiente, é utilizado para atualizar, com um algoritmo de otimização iterativa, o modelo de velocidades que no final do processo será capaz de corresponder aos dados reais. Aqui utilizaremos uma rede neural recorrente (*Recurrent Neural Network* - RNN), baseada na física governante (equação da onda acústica), para derivar os dados reais e os dados calculados no que tange a etapa de modelagem sísmica direta, visto que trataremos de dados puramente sintéticos. Além disso os ambientes de aprendizagem, como o *Pytorch*, nos proporcionam ferramentas para o cálculo do gradiente (diferenciação automática) e estratégia de mini-lote (em inglês, *mini-batches*) importante na questão de redução de memória e maior velocidade de processamento.

Como a FWI se baseia na minizimação iterativa de uma função custo entre dados observados e calculados, no intuito de evitar a convergência a mínimos locais utilizamos a abordagem multiescala na frequência. Além disso, testamos a resposta da FWI quando submetida a dados de observações ruidosos e a modelos iniciais menos precisos e comparamos com a resposta da inversão somada à técnica multiescala na frequência, para demonstrar a atuação desta abordagem na mitigação destas limitações.

Os resultados obtidos em três conjuntos de dados demostram a eficiência e aplicabilidade da técnica usada na tentativa de obter campos de velocidades sísmica de alta resolução.

Abstract

Seen as a fundamental parameter for a reliable geological image of the subsurface, and consequent exploratory success, seismic velocity is one of the prerequisites in the seismic exploration chain. Such velocity relates directly to the construction quality of seismic images through robust algorithms such as reverse time migration (RTM) or other seismic imaging techniques. In this work, we use machine learning environments to obtain high resolution seismic velocity through the full waveform inversion (FWI) technique.

In summary, the FWI technique aims to compare data from real observations with calculated data obtained through seismic modeling from the solution of a specific wave equation. The residual of this comparison is minimized and the gradient is used to update, with an iterative optimization algorithm, the velocity model that at the end of the process will be able to correspond to the real data. Here we will use a Recurrent Neural Network (RNN), based on the governing physics (acoustic wave equation), to derive the real data and the calculated data regarding the direct seismic modeling step, since we will deal with purely synthetic data. In addition, learning environments, such as Pytorch, provide us with tools for calculating the gradient (automatic differentiation) and the mini-batch strategy important in terms of reduction memory and higher processing velocity.

As the FWI is based on the iterative minimization of a cost function between observed and calculated data, in order to avoid convergence to local minima, we use the multiscale approach of frequency. In addition, we tested the FWI response when subjected to noisy observation data and less accurate initial models, and compared it with the inversion response added to the multiscale approach of frequency technique, to demonstrate the performance of this approach when it comes to mitigating these limitations.

The results obtained in three sets of data demonstrate the efficiency and applicability of the technique used in the attempt to obtain high resolution seismic velocity fields.

Índice

Índice de Tabelas

Índice de Figuras

Introdução

O advento de novas tecnologias no estudo da subsuperfície terrestre ao longo dos anos tem propiciado importantes avanços no entendimento da sua geologia interna e traz consigo respostas cada vez mais confiáveis e com grande redução de incertezas na investigação dos recursos naturais existentes, a exemplo do grande avanço no imageamento sísmico obtido pela indústria de exploração de petróleo e gás natural.

A grande complexidade geológica observada nos dados sísmicos adquiridos, suportados por estudos de iluminação prévios, parametrizações adequadas, ampla distribuição azimutal e alta densidade de traços por unidade de área, induz a constante busca por ferramentas de processamento robustas capazes de minimizar o risco embutido nas atividades de exploração e produção de grandes reservas de hidrocarbonetos. Nesse cenário, quando se particulariza o processamento dos dados sísmicos registrados, a busca pela melhor imagem da subsuperfície tem alta relevância.

Um dos problemas mais importantes que enfrentamos com frequência no processo exploratório de hidrocarbonetos é a estimativa dos parâmetros do modelo. Nesse contexto, a inversão da forma de onda completa (do inglês, *Full Waveform Inversion* - FWI) (Tarantola, 1984) tem sido um método bastante requerido para derivar um modelo de propriedades de subsuperfície, de alta resolução, como a velocidade da onda sísmica dos dados sísmicos registrados na superfície terrestre.

Neste cenário, por busca de melhorias tecnológicas, o campo da ciência de dados tem crescido em relevância e aplicabilidade no tratamento do conjunto de dados sísmicos. Esse desenvolvimento resulta da ampla disponibilidade de ambientes de aprendizagem, como o Tensorflow (Abadi et al., 2016) e o PyTorch (Paszke et al., 2016), dentre outros, usados na academia e na indústria. O presente trabalho visa agregar conceitos de inteligência artificial, na forma de aprendizado de máquina (em inglês, *machine learning*) e aprendizado profundo (também conhecido como *deep learning*) no fluxo de trabalho da FWI.

Recentemente, técnicas de aprendizado profundo, em particular, a rede neural recorrente (em inglês, *Recurrent Neural Network*), foram empregadas com sucesso numa variedade de

situações da geofísica. Dentre as diversas aplicações, foram utilizadas na reconstrução de dados sísmicos (Yoon, Yeeh e Byun, 2020), na estimativa da velocidade de normal moveout (Fabien-Ouellet e Sarkar, 2020) e capaz de realizar o esquema de diferenças finitas no domínio do tempo para simular a modelagem direta da equação acústica da onda (Richardson, 2018).

Neste trabalho, vamos seguir na direção da abordagem de Richardson (2018) de combinar redes neurais, especificamente o conjunto das RNN, e equações diferenciais parciais para formular a FWI como um problema de aprendizado de máquina guiado pela física. Aqui, como as operações necessárias para simular numericamente a propagação da onda são diferenciáveis, a diferenciação automática (Baydin et al., 2018), disponível nos ambientes de aprendizagem, será usada para calcular o gradiente da função de custo, que é o mesmo calculado pelo método do estado adjunto. Alguns trabalhos interessantes, que possuem este tipo de rede neural específica no fluxo da inversão sísmica, são apresentados por: Ren et al. (2020) que propõem uma rede de inversão da forma de onda sísmica, denominada SWINet; e Conceição (2021) que propõe a inversão num esquema de aprendizado não supervisionado no TensorFlow.

No primeiro capítulo desta dissertação, iremos abordar aspectos básicos para o entendimento da FWI convencional. Em seguida, vamos adentrar ao universo das redes neurais artificiais e explorar seus principais conceitos. No capítulo seguinte, iremos combinar as informações adquiridas nos capítulos iniciais para propiciar o entendendimento da FWI no campo do aprendizado de máquina. Neste terceiro capítulo, vamos mostrar um exemplo de inversão sísmica 1D através de um esquema de aprendizado supervisonado. No entanto, dadas algumas limitações inerentes a esta estratégia, partiremos para investigar o aprendizado não supervisionado FWI, que tem sido amplamente utilizado em geofísica para estimar mapas de velocidade de subsuperfície a partir de dados sísmicos.

Um ponto relevante é que como a FWI convencional é utilizada como solução de um problema de natureza estritamente não-linear, através de métodos de otimização local, acaba por sofrer com o problema dos mínimos locais no processo de obtenção do parâmetro ótimo. Para minimizar tal risco utilizamos a abordagem multiescala na frequência proposta por Bunks et al. (1995). Outros desafios encontrados pela FWI convencional são os dados de observações contaminadas por ruídos e a precisão dos modelos iniciais de velocidades. Como solução para estes últimos Yang e Ma (2021) propõem o método FWIGAN que usa redes neurais do tipo GAN (do inglês, *Generative Adversarial Network*) e Zhu et al. (2021) que utilizam rede neurais para introduzir correlações espaciais, como regularização para o modelo de velocidade gerado. Aqui mostraremos que a abordagem multiescala na frequência, além de minimizar o risco de ficarmos presos a mínimos locais, também consegue entregar

um resultado satisfatório quando submetida a dados ruídosos e a um modelo inicial menos preciso. Tais demonstrações se darão pela adição de ruído branco gaussiano aos dados das observações e a utilização de modelos de partida com diferentes níveis de suavização, e assim como estes os demais resultados obtidos serão expostos no capítulo 4.

1

Fundamentos

1.1 Conceitos Básicos

Devido à sua eficiência e qualidade, grande parte das empresas de exploração de petróleo e gás adotaram em seus fluxos de processamento de dados sísmicos os métodos de inversão sísmica para aumentar a resolução de dados, confiabilidade e melhorar a estimativa de propriedades físicas da geologia da subsuperfície. De acordo com Richter (2020) o termo "problema inverso" não tem uma definição matemática reconhecida, e seu significado depende de noções da física. Assim, suponhamos que exista um mapeamento conhecido

$$\mathbf{G} : \mathbb{M} \rightarrow \mathbb{U}$$

que modela uma lei física ou um dispositivo físico. Aqui, $\mathbb{M}$ é um conjunto de "causas" e $\mathbb{U}$ é um conjunto de "efeitos". O cálculo de um efeito $\mathbf{G}(\mathbf{m})$ para uma dada causa $\mathbf{m}$ é chamado de problema direto. Encontrar uma causa $\mathbf{m} \in \mathbf{G}$ que acarreta um determinado efeito $\mathbf{u} \in U$ é chamado de problema inverso. Resolver um problema inverso significa, portanto, encontrar a solução de uma equação téorica do tipo:

$$\mathbf{u} = \mathbf{G}(\mathbf{m}) \tag{1.1}$$

Um modelo $\mathbf{m}$ pode compreender, entre outras quantidades, as distribuições espaciais da velocidade da onda P, $\upsilon_P(\mathbf{r})$, a velocidade da onda S, $\upsilon_S(\mathbf{r})$, e densidade, $\rho(\mathbf{r})$, isto é

$$\mathbf{m}(\mathbf{r}) = [m_1(\mathbf{r}), m_2(\mathbf{r}), m_3(\mathbf{r}), ...] = [\upsilon_P(\mathbf{r}), \upsilon_S(\mathbf{r}), \rho(\mathbf{r}), ...]$$

onde $\mathbf{r} = (x, y, z)$ é o vetor das coordenadas espaciais.

A obtenção de imagens do subsolo é um objetivo comum do processamento de dados sísmicos, enquanto a inversão sísmica é sinônimo do processo inverso da imagem (Zhou, 2014), ou seja, obter objetivamente as propriedades físicas (ver Figura 1.1).

Neste capítulo, tratatemos dos conceitos básicos de modelagem direta e da solução do problema inverso (inversão sísmica).

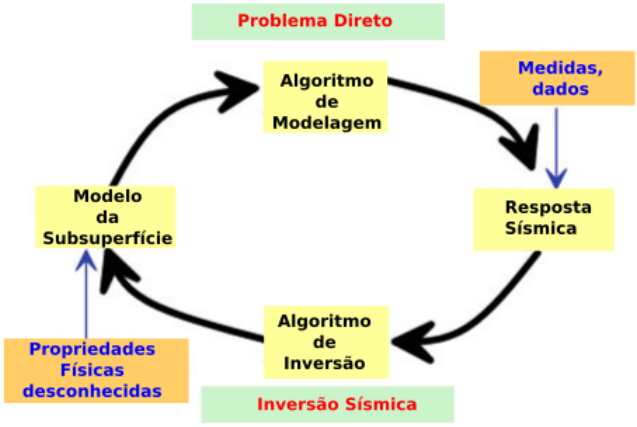


Figura 1.1: Diagrama esquemático representando os processos de modelagem direta e inversa. Adaptado de Maurya, Singh e Singh (2020).

1.2 Modelagem Sísmica Direta

A forma teórica frequentemente utilizada para representar os meios físicos da subsuperfície e que fornece respostas similares a um levantamento sísmico real, é a que utiliza a equação da onda acústica com densidade constante. No domínio do tempo expressa como:

$$\frac{\partial^2 \mathrm{u}(\mathbf{r}, \mathrm{t})}{\partial \mathrm{t}^2} = \upsilon^2(\mathbf{r})\nabla^2 \mathrm{u}(\mathbf{r}, \mathrm{t}) + \mathrm{s}(\mathbf{r}, \mathrm{t}) \tag{1.2}$$

onde $u(\mathbf{r}, \mathrm{t})$ é o campo de ondas (ou campo de pressão), υ é velocidade de propagação (representando o modelo $\mathbf{m}$), s é o termo fonte, t representa o tempo e ∇^2 é o operador Laplaciano dado por:

$$\nabla^2 = \left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2} \right)$$

1.2.1 Método das diferenças-finitas

Uma dos métodos para obtenção da solução numérica da equação da onda acústica para meios com densidade constante é o das diferenças finitas. Podemos reescrever a equação 1.2, a menos do termo fonte s, como:

$$\frac{\partial^2 u\,(\mathbf{r},t)}{\partial\, t^2} = -\,\mathbf{L}^2\, u\,(\mathbf{r},t) \tag{1.3}$$

onde $-\,\mathbf{L}^2 = v^2(\mathbf{r})\,\nabla^2$.

A equação 1.3 possui solução do tipo:

$$u\,(\mathbf{r},t) = A\,\cos(\mathbf{L}\,t) + B\,\sin(\mathbf{L}\,t)\,, \tag{1.4}$$

cujas constantes A e B podem ser calculadas à partir das seguintes condições de contorno:

$$u\,(\mathbf{r},t=0) = u_0 \quad \text{e} \quad \left[\frac{\partial\, u\,(\mathbf{r},t=0)}{\partial\, t}\right]_{t=0} = \dot{u}_0 \tag{1.5}$$

Utilizando as condições expressas em 1.5 e o instante $t = 0$, obtemos:

$$u\,(\mathbf{r},t) = u_0\,\cos(\mathbf{L}\,t) + \frac{\dot{u}_0}{\mathbf{L}}\,\sin(\mathbf{L}\,t) \tag{1.6}$$

Com base na equação 1.6, podemos escrever de maneira análoga, as equações para os instantes $t + \Delta t$ (posteriores) e $t - \Delta t$ (anteriores) respectivamente:

$$u\,(\mathbf{r},t+\Delta\,t) = u_0\,\cos[\mathbf{L}\,(t+\Delta\,t)] + \frac{\dot{u}_0}{\mathbf{L}}\,\sin[\mathbf{L}\,(t+\Delta\,t)] \tag{1.7}$$

e

$$u\,(\mathbf{r},t-\Delta\,t) = u_0\,\cos[\mathbf{L}\,(t-\Delta\,t)] + \frac{\dot{u}_0}{\mathbf{L}}\,\sin[\mathbf{L}\,(t-\Delta\,t)] \tag{1.8}$$

A soma das equações 1.7 e 1.8, desenvolvidos os cossenos e senos, resulta na solução analítica da equação da onda:

$$u\,(\mathbf{r},t+\Delta\,t) + u\,(\mathbf{r},t-\Delta\,t) = 2\,\cos(\mathbf{L}\,\Delta\,t)\,u\,(\mathbf{r},t) \tag{1.9}$$

Fazendo a expansão do termo $\cos(\mathbf{L}\,\Delta\,t)$ presente na equação 1.9 em série de Taylor, pode-se escrever:

$$\cos(\mathbf{L}\,\Delta\,t) = 1 - \frac{(\mathbf{L}\,\Delta\,t)^2}{2} + \frac{(\mathbf{L}\,\Delta\,t)^4}{24} - \frac{(\mathbf{L}\,\Delta\,t)^6}{720} + \dots$$

$$= \sum_{k=0}^{\infty} (-1)^k \cdot \frac{(\mathbf{L}\,\Delta\,t)^{2k}}{2k!} \tag{1.10}$$

Baseado na expansão de Taylor do cosseno, a solução analítica (equação 1.9) é reescrita como:

$$u\,(\mathbf{r}, t + \Delta\,t) + u\,(\mathbf{r}, t - \Delta\,t) = 2\,\left[\sum_{k=0}^{\infty} (-1)^k \cdot \frac{(\mathbf{L}\,\Delta\,t)^{2k}}{2k!}\right] u\,(\mathbf{r}, t) \tag{1.11}$$

Truncando a série apresentada na equação 1.10 até o termo de ordem 2, obtém-se:

$$u\,(\mathbf{r}, t + \Delta\,t) + u\,(\mathbf{r}, t - \Delta\,t) = 2\,\left[1 - \frac{(\mathbf{L}\,\Delta\,t)^2}{2}\right] u\,(\mathbf{r}, t)$$

$$= 2\,u\,(\mathbf{r}, t) - (\mathbf{L}\,\Delta\,t)^2\,u\,(\mathbf{r}, t) \tag{1.12}$$

De forma que podemos definir a solução da equação acústica da onda pelo método das diferenças-finitas de segunda ordem no tempo como:

$$u\,(\mathbf{r}, t + \Delta\,t) - 2\,u\,(\mathbf{r}, t) + u\,(\mathbf{r}, t - \Delta\,t) = -(\mathbf{L}\,\Delta\,t)^2\,u\,(\mathbf{r}, t) \tag{1.13}$$

lembrando que $-\,\mathbf{L}^2 = v^2(\mathbf{r})\,\nabla^2$ e a aproximação numérica da derivada segunda no tempo é:

$$\frac{\partial^2\,u\,(\mathbf{r}, t)}{\partial\,t^2} \approx \frac{u\,(\mathbf{r}, t + \Delta\,t) - 2\,u\,(\mathbf{r}, t) + u\,(\mathbf{r}, t - \Delta\,t)}{\Delta t^2}\,, \tag{1.14}$$

Baseado num procedimento análogo, podemos aumentar a precisão da aproximação da soluçao da equação 1.3 através da inclusão de mais termos da equação 1.10. Para obter, por exemplo, a aproximação de quarta ordem no tempo, basta truncar a expansão do cosseno por série de Taylor até o termo de ordem 4:

$$\frac{\partial^2\,u\,(\mathbf{r}, t)}{\partial\,t^2} \approx \frac{1}{(\Delta t)^2} - \frac{1}{12}u(\mathbf{r}, t + 2\Delta t) + \frac{3}{4}u(\mathbf{r}, t + \Delta t) - \frac{5}{2}u(\mathbf{r}, t)$$
$$+ \frac{3}{4}u(\mathbf{r}, t - \Delta t) - \frac{1}{12}u(\mathbf{r}, t - 2\Delta t) \tag{1.15}$$

No entanto, os algoritmos de modelagem sísmica que se baseiam no método das diferenças finitas, normalmente calculam a segunda derivada no tempo utilizando uma aproximação de segunda ordem, de modo que a solução numérica, incluindo o termo fonte s, assume a seguinte forma iterativa:

$$u\left(\mathbf{r}, t+\Delta t\right) = 2\,u\left(\mathbf{r}, t\right) - u\left(\mathbf{r}, t-\Delta t\right) - \left(\mathbf{L}\,\Delta t\right)^2 u\left(\mathbf{r}, t\right) + s(\mathbf{r}, \mathrm{t}) \tag{1.16}$$

ou ainda

$$u\left(\mathbf{r}, t+\Delta t\right) = 2\,u\left(\mathbf{r}, t\right) - u\left(\mathbf{r}, t-\Delta t\right) + \upsilon^2(\mathbf{r})\Delta t^2\nabla^2 u(\mathbf{r}, t) + s(\mathbf{r}, \mathrm{t}) \tag{1.17}$$

As derivadas espaciais, referentes ao operador Laplaciano (∇^2), da mesma forma que a derivada temporal, podem ser estimadas utilizando o método de diferenças finitas. Para o caso da aproximação de segunda ordem, temos a seguinte expressão:

$$\begin{aligned}\nabla^2 u(\underbrace{(x,y,z)}_{\mathbf{r}}, t) \approx & \frac{u\left(x+\Delta x, y, z, t\right) - 2\,u\left(x, y, z, t\right) + u\left(x-\Delta x, y, z, t\right)}{\Delta x^2} \\ & + \frac{u\left(x, y+\Delta y, z, t\right) - 2\,u\left(x, y, z, t\right) + u\left(x, y-\Delta y, z, t\right)}{\Delta y^2} \\ & + \frac{u\left(x, y, z+\Delta z, t\right) - 2\,u\left(x, y, z, t\right) + u\left(x, y, z-\Delta z, t\right)}{\Delta z^2}\end{aligned} \tag{1.18}$$

1.3 Inversão Geofísica

Os métodos de inversão visam estimar as propriedades geofísicas da subsuperfície a partir da medições feitas na superfície, ou também de perfis oriundos da perfilagem de poços. Essa estimativa é obtida através de métodos de otimização num processo de obtenção do máximo ou mínimo do problema inverso (Gill, Murray e Wright, 1981). Tais técnicas de otimização visam a obtenção, utilizando um esforço computacional, de uma solução ideal ou quase ótima para o problema (ver figura 1.2).

A abordagem inversa tem a vantagem de quantificar objetivamente as propriedades do modelo do subsolo geológico que sejam uma representação das observações. Destaca-se nesse contexto o método FWI que tem tido grande relevância na exploraração sísmica e que será descrito na Seção 1.4.

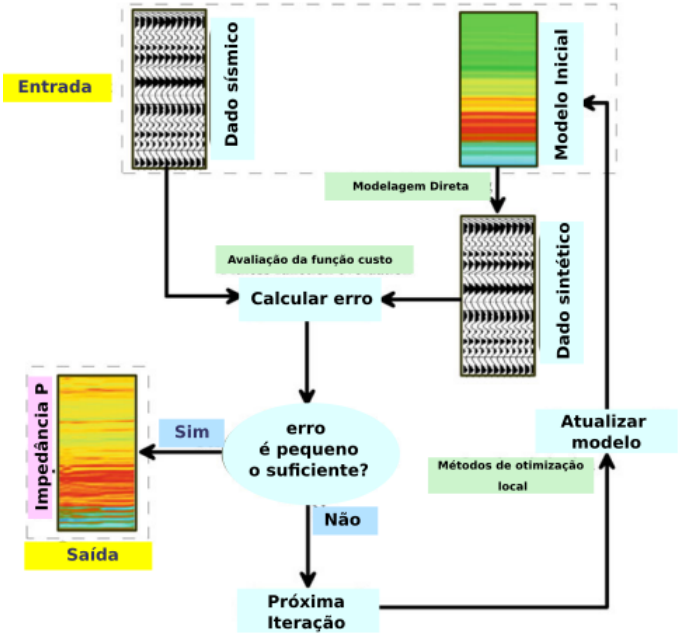


Figura 1.2: Fluxograma de métodos de otimização local. Adaptado de Maurya, Singh e Singh (2020)

1.4 Inversão da Forma de Onda Completa - (FWI)

A técnica da inversão da forma de onda completa (FWI) tem bases no princípio de migração de Claerbout (1971, 1976), posteriormente reformulado por Lailly (1983) e Tarantola (1984), que a descreveram como um problema de otimização local. O objetivo da FWI no sentido determinístico é encontrar um modelo ótimo da subsuperfície terrestre, $\mathbf{m}$, que minimize o funcional de desajuste, $J(\mathbf{m})$, usado para quantificar as discrepâncias entre os sismogramas observados, $\mathbf{d}_{obs}(\mathbf{r}, t)$, e os sismogramas calculados, $\mathbf{d}_{cal}(\mathbf{m}; \mathbf{r}, t)$.

Matematicamente o funcional de desajuste ou função objetivo a ser minimizada é definida como:

$$J(\mathbf{m}) = \frac{1}{2}\Delta\,\mathbf{d}^T\Delta\,\mathbf{d} \tag{1.19}$$

onde $\Delta\mathbf{d}$ é o desvio entre os sismogramas observados e calculados, ou seja,

$$\Delta\mathbf{d} = \mathbf{d}_{obs} - \mathbf{d}_{cal} \tag{1.20}$$

Os dados sintetizados, também chamados de calculados, são obtidos com base na modelagem numérica da equação da onda, sendo esta uma das engrenagens embutidas no fluxo da implementação da FWI. Esta etapa é caracterizada como busca pela solução do problema

direto. A relação entre o campo de onda sísmicas modelados $\mathbf{u}$ e os parâmetros físicos $\mathbf{m}$, de maneira compacta, pode ser compreendida pela relação mostrada na equação 1.1, onde $\mathbf{G}$ representa o operador de modelagem direta ou a física governante.

No contexto de um levantamento sísmico real, imposta uma geometria de aquisição, o campo de pressão gerado por uma fonte sísmica artificial (dinamite, *air guns*) é captado ao longo do tempo nos sensores (geofones, hidrofones, etc.), localizados na superfície ou próxima a ela, para dar origem aos sismogramas observados ($\mathbf{d}_{obs}$). O processo similar é feito, computacionalmente, para geração dos dados sintetizados ($\mathbf{d}_{cal}$), que são extraídos nas mesmas configurações de geometria dos dados observados, dos valores dos campos de onda $\mathbf{u}$ calculados ao longo do tempo.

A equação 1.1 destaca a relação não-linear, expressa por $\mathbf{G}(\mathbf{m})$, entre os parâmetros do modelo e os dados, que nos permite classificar a inversão da forma de onda como um problema de natureza não-linear (Virieux e Operto, 2009). Tal afirmação implica na impossibilidade de obtenção analítica do operador $\mathbf{G}^{-1}$, diferentemente de problemas categorizados como lineares.

A partir deste ponto nos deparamos com outra etapa da implementação da FWI, que é a busca pela solução do problema inverso, ao tentar determinar o parâmetro de modelo $\mathbf{m}$ que forneça um $\mathbf{J}(\mathbf{m})$ cujo valor seja mínimo. No entanto, devido à alta não-linearidade do problema, recorremos a aproximação de Born como estratégia para linearizar e transformar a FWI como um problema de otimização local iterativo, dado por:

$$\mathbf{m} = \mathbf{m}_0 + \Delta\mathbf{m} \tag{1.21}$$

tendo $\mathbf{m}_0$ como modelo inicial somado a uma perturbação $\Delta\mathbf{m}$.

Aproximando a função objetivo dada pela equação 1.19 via expansão de Taylor, em torno da vizinhança de $\mathbf{m}_0$, temos

$$J(\mathbf{m}_0 + \Delta\mathbf{m}) = J(\mathbf{m}_0) + \sum_{j=1}^{M} \frac{\partial J(\mathbf{m}_0)}{\partial m_j}\Delta\mathbf{m}_j + \frac{1}{2}\sum_{j=1}^{M}\sum_{k=1}^{M} \frac{\partial^2 J(\mathbf{m}_0)}{\partial m_j \partial m_k}\Delta\mathbf{m}_j\Delta\mathbf{m}_k + O(\mathbf{m}^3) + \dots$$

Com a aproximação de 2^a ordem, temos que:

$$J(\mathbf{m}_0 + \Delta\mathbf{m}) \approx J(\mathbf{m}_0) + \sum_{j=1}^{M} \frac{\partial J(\mathbf{m}_0)}{\partial m_j}\Delta\mathbf{m}_j + \frac{1}{2}\sum_{j=1}^{M}\sum_{k=1}^{M} \frac{\partial^2 J(\mathbf{m}_0)}{\partial m_j \partial m_k}\Delta\mathbf{m}_j\Delta\mathbf{m}_k \tag{1.22}$$

Como o objetivo da FWI é minimizar $\mathbf{J}(\mathbf{m})$ na busca pelo parâmetro ótimo, isso corresponde ao ponto no qual a aproximação em torno de $\mathbf{m}_0$ é um ponto de mínimo, ou seja:

$$\frac{\partial \mathbf{J}(\mathbf{m})}{\partial m_l} = 0 \tag{1.23}$$

Assim, tomando a derivada da equação 1.22 com relação a m_l temos

$$\frac{\partial J\left(\mathbf{m}\right)}{\partial m_l} = \frac{\partial J\left(\mathbf{m}_0\right)}{\partial m_l} + \sum_{j=1}^{M} \frac{\partial^2 J\left(\mathbf{m}_0\right)}{\partial m_j \partial m_l} \Delta \mathbf{m}_j + \frac{1}{2} \sum_{j=1}^{M} \sum_{k=1}^{M} \frac{\partial^3 J\left(\mathbf{m}_0\right)}{\partial m_j \partial m_k \partial m_l} \Delta \mathbf{m}_j \Delta \mathbf{m}_k \tag{1.24}$$

Desprezando os termos com derivadas de grau maior que 2 chegamos a

$$\frac{\partial J\left(\mathbf{m}\right)}{\partial m_l} = \frac{\partial J\left(\mathbf{m}_0\right)}{\partial m_l} + \sum_{j=1}^{M} \frac{\partial^2 J\left(\mathbf{m}_0\right)}{\partial m_j \partial m_l} \Delta \mathbf{m}_j, \tag{1.25}$$

igualando a zero e escrevendo em notação vetorial resulta em:

$$\Delta \mathbf{m} = - \overbrace{\left[\frac{\partial^2 J\left(\mathbf{m}_0\right)}{\partial \mathbf{m}^2} \right]^{-1}}^{\text{Hessiana}} \underbrace{\frac{\partial J\left(\mathbf{m}_0\right)}{\partial \mathbf{m}}}_{\text{Gradiente}} \tag{1.26}$$

Assim, obtemos o termo referente a perturbação do modelo da aproximação de Born, que é a quantidade necessária para que a atualização dos parâmetros do modelo $\mathbf{m}$ produza o resíduo mínimo.

A minimização é realizada de forma iterativa, e baseado na formulação do método de Newton, uma atualização do modelo pode ser obtida (Ma e Hale, 2012) como:

$$\mathbf{m}_{k+1} = \mathbf{m}_k - \mathbf{H}_k^{-1} \mathbf{g}_k, \tag{1.27}$$

onde k é o índice da iteração, $\mathbf{H}_k$ é a matriz Hessiana e $\mathbf{g}_k$ é o gradiente da função $\mathbf{J}(\mathbf{m})$.

Todavia, o cálculo da matriz Hessiana ($\mathbf{H}$), e de sua inversa, normalmente tem caráter proibitivo devido ao custo computacional (Virieux e Operto, 2009). A alternativa é lançar mão de outros métodos aproximados que se baseiam em esquemas iterativos e possuem a seguinte forma:

$$\mathbf{m}_{k+1} = \mathbf{m}_k + \alpha_k \mathbf{p}_k, \tag{1.28}$$

onde α_k é o comprimento do passo e $\mathbf{p}_k$ é o vetor de direção de busca.

Por sua vez, $\mathbf{p}_k$ pode ser escrito como (Nocedal e Wright, 2006):

$$\mathbf{p}_k = -\mathbf{B}_k^{-1}\mathbf{g}_k \tag{1.29}$$

onde $\mathbf{B}_k$ representa uma matriz que é uma aproximação da matriz Hessiana.

1.4.1 Cálculo do Gradiente - $\mathbf{g}_k$

Os primeiros trabalhos a propor o cálculo do gradiente da função objetivo aplicado ao problema sísmico de forma eficiente foram de Lailly (1983) e Tarantola (1984). O cálculo do gradiente da função objetivo é dado por:

$$\nabla J(\mathbf{m}) = \frac{\partial J(\mathbf{m})}{\partial \mathbf{m}} = \frac{\partial}{\partial \mathbf{m}}\left(\frac{1}{2}\Delta\,\mathbf{d}^T\Delta\,\mathbf{d}\right)$$

$$\nabla J(\mathbf{m}) = \frac{1}{2}\left(\frac{\partial}{\partial \mathbf{m}}\Delta\,\mathbf{d}^T\Delta\,\mathbf{d}\right)$$

$$\nabla J(\mathbf{m}) = \frac{1}{2}\left(-\frac{\partial \mathbf{d}_{cal}^T}{\partial \mathbf{m}}\Delta\,\mathbf{d} - \Delta\,\mathbf{d}^T\frac{\partial \mathbf{d}_{cal}}{\partial \mathbf{m}}\right)$$

$$\nabla J(\mathbf{m}) = -\frac{\partial \mathbf{d}_{cal}^T}{\partial \mathbf{m}}\Delta\,\mathbf{d} \tag{1.30}$$

onde o termo $\frac{\partial \mathbf{d}_{cal}}{\partial \mathbf{m}}$ é denominado de derivada de Fréchet.

Estimar iterativamente esse gradiente da função objetivo por meio da matriz de Fréchet, também dita matriz Jacobiana ou sensibilidade, requer grande esforço computacional. Como alternativa, Lailly (1983) e Tarantola (1984) utilizaram o método baseado no estado adjunto (do inglês, *adjoint-state method*), introduzido na teoria dos problemas inversos por Chavent (1974), para calcular esse gradiente sem a necessidade do cálculo das tais derivadas de Fréchet.

Pelo método do estado adjunto o gradiente pode ser calculado no domínio do tempo através da seguinte relação (Apêndice A):

$$\frac{\partial J(\mathbf{m})}{\partial \mathbf{m}} = \frac{2}{\upsilon^3(\mathbf{r})}\sum_{s,r}\int_0^T \lambda(\mathbf{r},t)\frac{\partial^2 \mathbf{u}}{\partial t^2}dt \tag{1.31}$$

onde o parâmetro $\mathbf{m}$ representa a velocidade de propagação das ondas (υ), a variável de estado $\mathbf{u}$ que representa o campo de ondas da fonte, calculado com a equação 1.2, T representa o tempo máximo de registro do dado observado e λ é a chamada variável adjunta, obtida através da solução da equação (Apêndice A):

$$\frac{1}{\upsilon^2(\mathbf{r})}\frac{\partial^2\lambda(\mathbf{r},t)}{\partial t^2} = \nabla^2\lambda(\mathbf{r},t) + \Delta\mathbf{d} \tag{1.32}$$

onde $\Delta\mathbf{d}$ é o resíduo entre o dado sísmico observado e o dado calculado para o modelo corrente ($\mathbf{m}_k$). Revisitando novamente a equação 1.16 percebe-se uma analogia em relação a equação 1.32. A leitura que pode ser feita da equação 1.32 é que λ é o campo de ondas resultante da propagação do resíduo ($\Delta\mathbf{d}$), termo fonte, no modelo de velocidades corrente. Entretanto, essa propagação é feita de forma reversa no tempo, devido às condições de contorno impostas à equação 1.32.

As semelhanças entre o algoritmo para o cálculo do gradiente da função objetivo, via método adjunto, com o método de migração, foram apontadas por Tarantola (1984) e apoiadas no princípio de imageamento proposto por Claerbout (1971). Da mesma maneira que a migração reversa no tempo (em inglês, *Reverse Time Migration* - RTM) ocorre por correlação cruzada, o cálculo do gradiente também é obtido pela correlação de lag zero entre dois campos ($\mathbf{u}$ e $\lambda(\mathbf{r},t)$). Em linhas gerais, o gradiente nada mais é do que a migração (RTM) do resíduo $\Delta\mathbf{d}$ para o modelo corrente $\mathbf{m}_k$, a menos do termo $\frac{2}{\upsilon(\mathbf{r})^3}$.

1.4.2 Direção de busca ($\mathbf{p}_k$) e comprimento do passo (α_k)

A direção de atualização que minimiza a função objetivo tem papel essencial no fluxo da FWI. Vimos na equação 1.27 que uma direção de busca eficiente é dada pela inversa da matriz Hessiana aplicada ao vetor gradiente. Todavia, no processamento de dados sísmicos a eficiência computacional tem grande relevância, e como citado anteriormente o alto custo atribuído à inversa da Hessiana nos leva a lançar mão de métodos baseados em aproximações para obtenção da direção de busca como o da equação 1.29.

Dentre os métodos baseados em aproximações se destacam os métodos gradientes, nos quais a equação 1.29 é obtida por:

$$\mathbf{p}_k = -\mathbf{g}_k \tag{1.33}$$

onde é assumido que $\mathbf{B}_k$ equivale à matriz identidade $\mathbf{I}$. Sendo esta a forma mais simples de calcular a direção de busca. Aqui o termo $-\mathbf{g}_k$ é uma direção decrescente da função, em

que um novo modelo ($\mathbf{m}_{k+1}$), obtido nessa direção caminha em direção ao mínimo da função objetivo J, com a ressalva de um comprimento do passo (α_k) adequado.

O método gradiente que se baseia na relação 1.33 é conhecido como *steepest-descent method* ou método de decrescimento mais rápido. Nesta abordagem a direção de busca assumida tende a ser ortogonal à direção da iteração anterior (Zhou, Gao e Dai, 2006), o que confere um aspecto de "zigue-zague" ao processo de convergência, como ilustra a Figura 1.3.

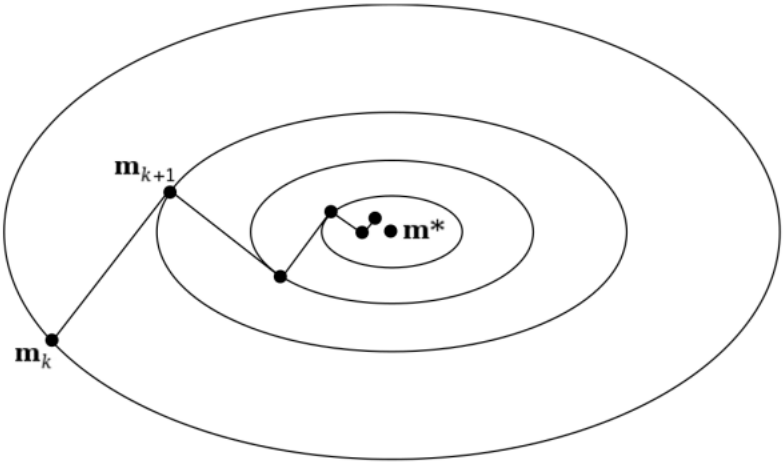


Figura 1.3: Convergência do método steepest-descent (Nocedal e Wright, 2006).

A outra classe de métodos aproximados são chamados de quasi-Newton e se baseiam na seguinte propriedade:

$$\mathbf{H}_{k+1}(\mathbf{m}_{k+1} - \mathbf{m}_k) \approx (\mathbf{g}_{k+1} - \mathbf{g}_k) \tag{1.34}$$

que realiza a estimativa de $\mathbf{B}$ sem o cálculo explícito das segundas derivadas de J, nas proximidades do modelo verdadeiro. Nesse caso a equação 1.34 pode ser reescrita como (Nocedal e Wright, 2006):

$$\mathbf{B}_{k+1}\,\mathbf{s}_k = \mathbf{y}_k \tag{1.35}$$

onde $\mathbf{s}_k = \mathbf{m}_{k+1} - \mathbf{m}_k$ e $\mathbf{y}_k = \mathbf{g}_{k+1} - \mathbf{g}_k$. O mais popular desses métodos é denominado de BFGS (nomeado a partir dos criadores Broyden, Fletcher, Goldfarb e Shanno), que define a atualização da inversa de $\mathbf{B}$ (dos Santos, 2013) como:

$$\mathbf{B}_{k+1}^{-1} = \mathbf{B}_k^{-1} - \frac{\mathbf{B}_k^{-1}\mathbf{y}_k\mathbf{y}_k^T\mathbf{B}_k^{-1}}{\mathbf{y}_k^T\mathbf{B}_k^{-1}\mathbf{y}_k} + \frac{\mathbf{s}_k\mathbf{s}_k^T}{\mathbf{y}_k^T\mathbf{s}_k} \tag{1.36}$$

Todavia armazenar modelos e gradientes de iterações passadas demanda muitos dos recursos computacionais, fato este que levou ao surgimento de uma variação ao método denominada L-BFGS (BFGS de memória limitada - *Limited-memory BFGS*) onde não é preciso ter a matriz $\mathbf{B}$ explicitamente armazenada. O método L-BFGS em geral é muito melhor escalado que o *steepest-descent* e o comprimento do passo $\alpha_k = 1$, normalmente, é a melhor escolha (dos Santos, 2013).

Determinada a direção de busca do novo modelo ($\mathbf{m}_{k+1}$) é preciso encontrar o fator de escala (α_k), que será aplicado ao vetor $\mathbf{p}_k$, de forma a propiciar uma atualização coerente com os valores de velocidade do modelo, capaz de minimizar o valor da função objetivo J em relação à iteração anterior. Uma discussão teórica aprofundada sobre os tópicos aqui apresentados pode ser encontrada em dos Santos (2013).

2

Rede Neurais

2.1 Aprendizado de Máquina

O interesse por aprendizado de máquina (em inglês, *machine learning*), um campo da inteligência artificial (IA), tem se mostrado mais evidente na geofísica pelo número crescente de trabalhos voltados para este tema. Essa técnica é fundamentada na utilização de algoritmos para extrair informações de dados brutos e representá-los através de algum tipo de modelo matemático. Dentre os algoritmos capazes de realizar essa tarefa, as redes neurais artificiais (ANNs - *Artificial Neural Networks*) tem tido grande destaque.

Antes de aprofundarmos o conhecimento nesse ramo da ciência é de fundamental importância buscar compreender a lógica de funcionamento das redes neurais, como alguns conceitos básicos referentes ao funcionamento do cérebro humano e seus componentes, os neurônios.

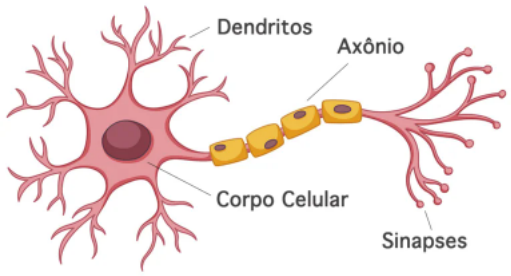
2.1.1 Neurônio Biológico

O cérebro humano é considerado altamente poderoso e complexo, capaz de processar uma grande quantidade de informações rapidamente. O entendimento dessa complexidade se tornou mais fácil pelo trabalho pioneiro de Ramón y Cajál (1911), que introduziu a ideia dos neurônios, que são tidos como unidades principais do cérebro e por onde as informações são transmitidas e processadas. Estruturalmente o neurônio é composto por 3 segmentos principais: corpo celular ou soma, os dendritos e o axônio como mostra a Figura 2.1.

A começar pelos dendritos, estes têm o papel de captar os sinais elétricos de outros

neurônios através de junções denominadas sinapses neurais, que serão conduzidos ao corpo celular. Por sua vez, o corpo celular realiza a coleta dos sinais recebidos pelos dendritos, e assume a responsabilidade de processar e combinar essas informações. Em seguida, um fator de ativação de natureza química indicará se ocorre, ou não, a propagação de um impulso elétrico em direção ao axônio, que desempenhará a função de transmitir tal impulso para outros neurônios. Assim, quando expandimos esse entendimento aos bilhões de neurônios existentes em nosso cerébro, que não estão isolados, esse processo complexo e fascinante forma uma enorme rede de comunicação, a rede neural.

Como pode ser observado na Figura 2.1 o corpo celular e os dendritos formam a superfície de entrada do neurônio e o axônio a superfície de saída do fluxo de informação. Essa observação é relevante, pois servirá para o entendimento do modelo do neurônio matemático, que será descrito a seguir, desenvolvido por pesquisadores inspirados no neurônio biológico e se tornou base da Inteligência Artificial.



Fonte: https://growiz.com.br/redes-neurais-artificiais-criando-um-perceptron-de-uma-camada-em-c/

Figura 2.1: Representação Simplificada do Neurônio Biológico.

2.1.2 Neurônio Matemático

A analogia neurobiológica serviu como fonte inspiradora para pesquisadores tentarem simular a estrutura e funcionamento do neurônio biológico em computador. Dentre os modelos, o mais bem aceito foi proposto por McCulloch e Pitts (1943). No funcionamento deste neurônio (ver Figura 2.2), o conjunto de informações de entrada (x_j) são somados e ponderados por pesos sinápticos (w_{kj}). Após o processamento desta soma ($\sum$) por uma função de ativação (φ), caso o resultado supere um limiar (específico de cada neurônio), a saída resultante é não nula. O modelo de McCulloch e Pitts (1943) possui uma natureza binária, eles assumiam que o seu modelo formal de um neurônio seguia uma lei de "tudo ou nada".

Com base na analogia feita com o neurônio biológico, os dendritos seriam representados pelos sinais de entrada, o corpo celular compreendido pelo somatório dos sinais de entrada

multiplicado pelo seu fator excitatório e a saída, obtida pela aplicação da função de ativação, o axônio.

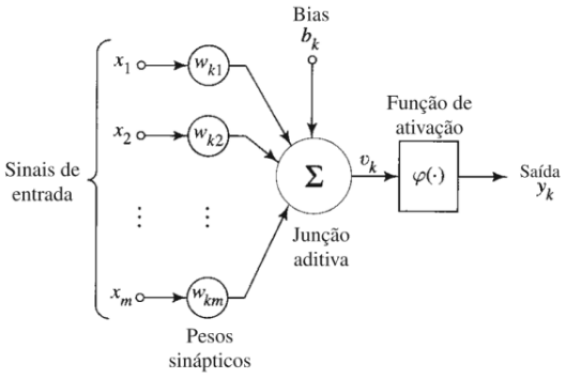


Figura 2.2: Representação do Neurônio Matemático (Haykin, 2007).

Matematicamente podemos descrever um neurônio k utilizando o seguinte par de equações (Haykin, 2007):

$$u_k = \sum_{j=1}^{m} w_{kj} x j \tag{2.1}$$

e

$$y_k = \varphi(u_k + b_k) \tag{2.2}$$

onde o índice j representa o elo de ligação entre cada entrada x_j e seu respectivo peso sináptico w_{kj}, b_k é o bias, $\varphi(\cdot)$ é a função de ativação e y_k é o sinal de saída do neurônio.

Podemos reescrever o argumento de $\varphi(\cdot)$ na equação 2.2 da seguinte forma:

$$u_k + b_k = v_k \tag{2.3}$$

onde v_k é chamado de campo local induzido ou potencial de ativação. Essa relação demonstra o efeito de aplicar o bias à saída do combinador linear u_k e está ilustrada na Figura 2.3. Desta forma, é possível observar que o viés ou bias visa aumentar o grau de liberdade de $\varphi(\cdot)$ e tem o efeito de aumentar ou diminuir o seu argumento, caso seja positivo ou negativo, respectivamente.

O bias também pode ser enxergado como um parâmetro externo do neurônio artificial k (Haykin, 2007). O que nos permite reformular a equação 2.3 como:

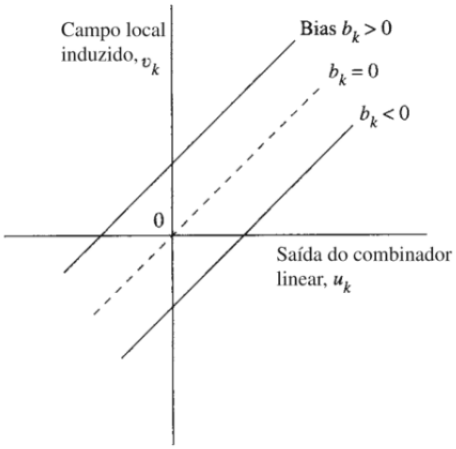


Figura 2.3: Transformação afim produzida pela presença de um bias (Haykin, 2007).

$$v_k = \sum_{j=1}^{m} w_{kj} xj + b_k \tag{2.4}$$

ou ainda adicionando um nova sinapse, composta pelo sinal de de entrada $x_0 = +1$ e peso $w_{k0} = b_k$, temos:

$$v_k = \sum_{j=0}^{m} w_{kj} xj \tag{2.5}$$

Assim, temos que:

$$y_k = \varphi(v_k) \tag{2.6}$$

e o modelo do neurônio k é reformulado como mostrado na Figura 2.4.

A função de ativação tem o objetivo de restringir a amplitude da saída do neurônio. Existem vários tipos de funções de ativação, normalmente são de natureza não-linear e possuem valores máximos e mínimos contidos em intervalos determinados. As funções de ativação mais utilizadas são:

- $\varphi(v) = \begin{cases} 0 \ \ se \ \ v \geq 0 \\ 1 \ \ se \ \ v < 0 \end{cases}$ (Função de Limiar)

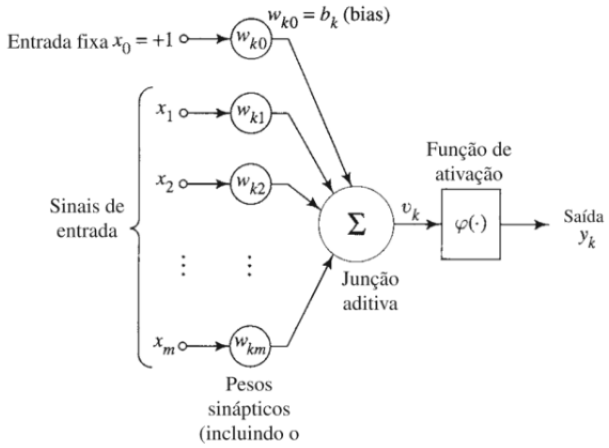


Figura 2.4: Representação de outro modelo de Neurônio Matemático (Haykin, 2007).

- $\varphi(v) = \begin{cases} 1 \ \ se \ \ v \geq +\frac{1}{2} \\ v \ \ se \ \ +\frac{1}{2} > v > -\frac{1}{2} \\ 0 \ \ se \ \ v \leq -\frac{1}{2} \end{cases}$ (Função Linear por Partes);
- $\varphi(v) = \frac{1}{1+\exp(-\alpha v)}$ (Função Sigmóide);
- $\varphi(v) = tanh(v)$ (Função Tangente Hipérbolica).

O conceito do neurônio matemático funcionou como propulsor para várias arquiteturas e modelos com diferentes combinações entre eles. Análogo ao cerébro humano, com seus inúmeros neurônios (rede neural), a composição dos vários neurônios artificiais, baseadas em diferentes técnicas matemáticas e estatísticas, formam as redes neurais artificiais e propiciaram o surgimento e criação de arquiteturas avançadas de apredizagem profunda (*Deep Learning*).

2.2 Arquitetura de Redes

De uma forma simplificada, mostramos que um neurônio matemático de uma rede neural artificial é um componente que calcula a soma ponderada de várias entradas, aplica uma função restritiva e gera uma saída. A forma como um conjunto destes neurônios estão dispostos numa rede tem relação intrínseca com o algoritmo de aprendizagem usado para treiná-la. Em geral, essa organização é subdividida em 3 classes (Haykin, 2007): redes alimentadas adiante com camada única, redes alimentadas diretamente com múltiplas camadas e redes recorrentes.

2.2.1 Redes Alimentadas Adiante com Camada Única

As redes deste tipo também são conhecidas como redes *feedforward*. Neste tipo de arquitetura os neurônios são organizados em forma de camadas, como pode ser visto na Figura 2.5. Os nós da camada de entrada se comunicam diretamente com a camada de saída. É dita de camada única em referência à camada de saída (nós computacionais). A camada de entrada não é levada em consideração por não haver qualquer computação nela.

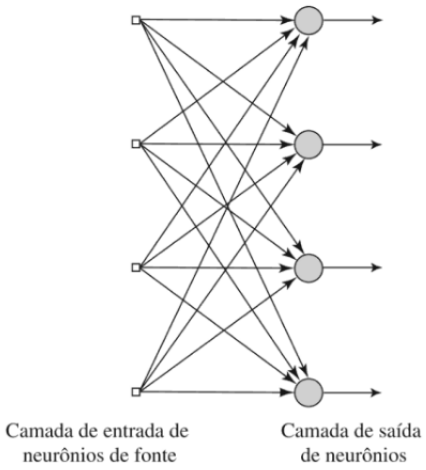


Figura 2.5: Rede alimentada adiante com uma única camada de neurônios (Haykin, 2007).

2.2.2 Redes Alimentadas Diretamente com Múltiplas Camadas

Este tipo de rede, mostrada na Figura 2.6, pode ser entendida como um melhoramento da arquitetura de camada única. A distinção se dá pela presenca de uma ou mais camadas ocultas (do inglês, *hidden layers*). Estes neurônios ocultos têm a função de intervir no elo entre a camada de entrada e a saída da rede. A adição de uma ou mais camadas ocultas possibilita a rede extrair estatísticas de maior complexidade e adquirir perspectiva global (Churchland e Sejnowski, 1992). Por ser classificada também como uma rede do tipo *feedforward*, as conexões se dão sempre no sentido da camada de entrada para a de saída. Quando a rede possuir todos os nós de uma camada comunicando-se com todos os nós da camada posterior, ela é dita totalmente conectada. Em situações que alguns elos de comunicação estiverem ausentes na rede, dizemos que a rede é parcialmente conectada.

2.2.3 Redes Recorrentes

Diferentemente das redes do tipo *feedforward* que apenas transmitem a informação da camada de entrada em direção a saída, sem memorizar nada, as redes neurais recorrentes (em

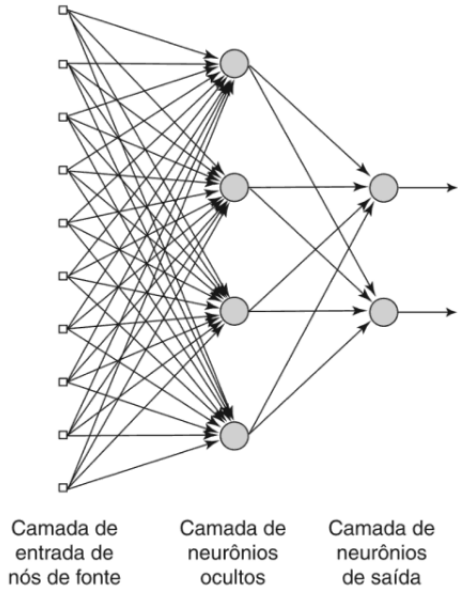


Figura 2.6: Rede alimentada adiante totalmente conectada com uma camada oculta e uma camada de saída (Haykin, 2007).

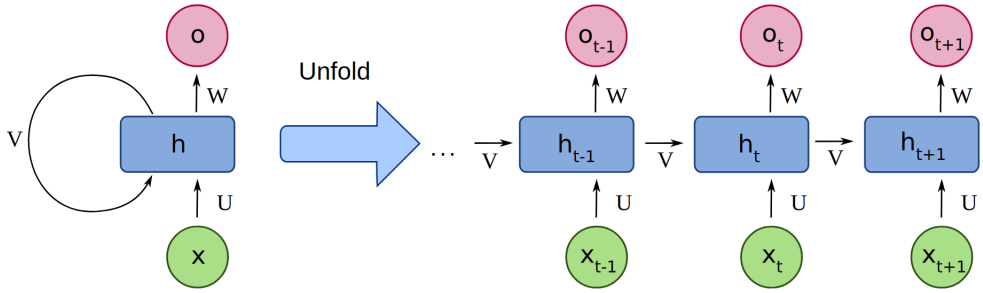
inglês, *Recurrent Neural Network* - RNN) incluem um conceito de memória ao ser executada. Essa classe de redes neurais artificiais são especialmente úteis para o processamento de dados sequenciais, dados de séries temporais ou linguagem natural (Hughes et al., 2019). Assim como as redes *feedforward* que tiveram diversas configurações, uma variedade de redes recorrentes foram propostas, como as redes de Elman (Elman, 1990), redes de Jordan (Jordan, 1986), rede neural com atraso de tempo (Lang, Waibel e Hinton, 1990) e redes de estado de eco (Jaeger, 2001). Neste trabalho, o foco são as redes recorrentes de Elman (ver Figura 2.7), cuja implementação é descrita pelas seguintes equações

$$\mathbf{h}_t = \varphi^{(h)}(\mathbf{V}\mathbf{h}_{t-1} + \mathbf{U}\mathbf{x}_t + \mathbf{b}^{(h)})$$

$$\mathbf{o}_t = \varphi^{(o)}(\mathbf{W}\mathbf{h}_t + \mathbf{b}^{(o)}) \tag{2.7}$$

onde $\mathbf{U}$, $\mathbf{V}$ e $\mathbf{W}$ são matrizes de peso, $\mathbf{b}^{(h)}$ e $\mathbf{b}^{(o)}$ os viés ou bias e $\varphi^{(h)}$ e $\varphi^{(o)}$ as funções de ativação.

As redes recorrentes de Elman diferem das redes *feedforward* pela inclusão de um loop de retroalimentação, pelo qual o estado oculto do passo $\mathbf{h}_{t-1}$ é alimentado de volta à rede para afetar o resultado do passo $\mathbf{h}_t$, e assim por diante para cada etapa subsequente. Essa característica permite dizer que elas possuem memória, semelhantes à forma como humanos



Fonte: https://commons.wikimedia.org/wiki/File:Recurrent_neural_network_unfold.svg

Figura 2.7: Rede Neural Recorrente em suas representações compacta (esquerda) e desenrolada (direita).

processam informação, que lhes confere a capacidade de reconhecimento de um contexto através da memória.

2.3 Processos de Aprendizagem

Descritas as arquiteturas das redes, é preciso destacar um aspecto importante: o algoritmo de aprendizado que será implementado, sendo as formas mais comuns o algoritmo supervisionado e o não supervisionado. Treinar uma rede neural artificial significa encontrar valores ideais para os pesos sinápticos e bias, daí a importância do algoritmo de aprendizagem de máquina. Um algoritmo de aprendizagem consiste em uma função objetivo, também chamada de perda ou custo, e uma técnica de otimização.

2.3.1 Aprendizado Supervisionado

Suponhamos um conjunto de dados fornecidos a uma rede neural, contendo pares de dados de entrada e saída conhecidos. O aprendizado supervisionado consiste em inferir uma função que mapeia uma entrada para uma saída com base nos pares de entrada-saída fornecidos (dados rótulados), e que pode ser usada para mapear novos pares. Em outras palavras, a tarefa do algoritmo de aprendizagem é aprender os parâmetros da rede (pesos), pois estes parâmetros descrevem a probabilidade dos padrões que a rede está aprendendo e que refletem os relacionamentos reais nos dados.

Em síntese, o algoritmo de aprendizado compara a saída estimada com a saída rotulada e encontra erros para modificar os parâmetros da rede. O ajuste é feito passo a passo, até que o erro seja mínimo. Dentre as técnicas mais conhecidas para resolver problemas de aprendizado

supervisionado estão regressão linear, regressão logística, e redes neurais artificiais.

Rosenblatt (1958) propôs o perceptron como o primeiro modelo para aprendizagem supervisionada, chamada de rede perceptron ou perceptron de camada única. Esta rede foi a estrutura mais simples de uma ANN, usada com sucesso em classificação de padrões. A generalização do perceptron é a sua versão multicamadas (em inglês, *multilayer perceptron* - MLP).

2.3.2 Aprendizado Não Supervisionado

Já no aprendizado não supervisionado, não há professor para supervisionar o processo de aprendizagem, ou seja, o método extrai significado dos dados sem treinar um modelo em dados rotulados. Em vez disso, são dadas condições para realizar uma medida independente da tarefa de qualidade da representação que a rede deve aprender, e os parâmetros livres da rede são otimizados em relação a esta medida (Haykin, 2007). A aprendizagem não superivisonada pode desempenhar um papel importante na previsão, tanto para problemas de regressão como de classificação.

2.3.3 Função Objetivo

A função objetivo pode ser entendida como uma função que busca maximizar ou minimizar, dependendo do objetivo do problema. No ambiente das redes neurais, em problemas supervisionados, a função de perda nos informa o nível de precisão de nossa rede ao fazer previsões para uma determinada entrada, através do processo de minimização da mesma.

Consideremos o exemplo de um fenômeno estocástico descrito por um vetor aleatório $\mathbf{X}$ (conjunto de variáveis independentes) e um escalar aleatório D (variável dependente). Vamos supor também que tenhamos N realizações do vetor X representadas por $\{\mathbf{x}_i\}_{i=1}^{N}$ e o par correspondente de D representado por $\{d_i\}_{i=1}^{N}$. Estas medidas constituem a amostra de treinamento representada por

$$\mathcal{T} = \{\mathbf{x}_i, d_i\}_{i=1}^{N} \tag{2.8}$$

A ideia central numa rede neural é codificar o conhecimento empírico, representado pela amostra de treinamento $\mathcal{T}$ em um conjunto correspondente de vetores de pesos sinápticos, $\mathbf{w}$, como mostrado por

$$\mathcal{T} \rightarrow \mathbf{w} \tag{2.9}$$

Para completar vamos supor que a resposta real da rede neural, produzida em resposta ao vetor de entrada $\mathbf{x}$, seja

$$Y = F(\mathbf{X}, \mathbf{w}) \tag{2.10}$$

onde $F(\cdot, \mathbf{w})$ é a função entrada-saída realizada pela rede neural. Então, conhecidos os dados de treinamento da equação (2.8), $\mathbf{w}$ é obtido pela minimização da função custo:

$$J(\mathbf{w}) = \frac{1}{2}\sum_{i=1}^{N}(d_i - F(\mathbf{x}_i, \mathbf{w}))^2 \tag{2.11}$$

A equação 2.11 representa a função objetivo mais utilizada em problemas de regressão, a de erro quadrático, em razão da sua simplicidade, diferenciabilidade, velocidade de processamento ou eficácia (Conceição, 2021).

Por fim, podemos fazer a analogia com o mesmo tipo de função objetivo, também de erro quadrático, que já vimos na equação (1.19), onde nesse caso d é a resposta desejada (dados observados), y resposta da rede (dados calculados) e a diferença entre eles o erro.

2.3.4 Técnicas de Otimização

Consideremos que a função custo dada pela equação (2.11) seja uma função continuamente diferenciável de $\mathbf{w}$. O objetivo da otimização é encontrar a solução ótima ($\mathbf{w}*$) que satisfaz a condição

$$J(\mathbf{w}*) \leq J(\mathbf{w}) \tag{2.12}$$

ou seja, minimizar $J(\mathbf{w})$ em relação $\mathbf{w}$. A condição para tal otimização é

$$\nabla J(\mathbf{w}*) = 0 \tag{2.13}$$

onde ∇ é o operador gradiente.

Vale ressaltar que a utilização destes métodos de otimização em geral não leva ao mínimo da função custo num único passo, mas iterativamente, sendo estes baseados na ideia da descida iterativa local em que

$$J(\mathbf{w}_{t+1}) < J(\mathbf{w}_t) \tag{2.14}$$

onde $\mathbf{w}_t$ é o valor antigo do vetor peso e $\mathbf{w}_{t+1}$ é o seu valor atualizado. A seguir descreveremos algumas técnicas de otimização baseadas em gradiente descendente e utilizadas em *machine learning*.

Descida do Gradiente

No método descida do gradiente (do inglês, *gradient descent*), método gradiente ou método de descida mais íngreme (Haykin, 2007), as atualizações sucessivas aplicadas ao vetor peso $\mathbf{w}$ são na direção da descida mais íngreme, ou seja, na direção oposta ao vetor gradiente. Então, seja

$$\mathbf{g} = \nabla J(\mathbf{w}) \tag{2.15}$$

O algoritmo do método gradiente é descrito formalmente por

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \eta \mathbf{g}_t \tag{2.16}$$

onde η é uma constante positiva denominada taxa de aprendizagem, e $\mathbf{g}_t$ é o vetor gradiente calculado no ponto $\mathbf{w}_t$. Assim, passando da iteração t para a iteração $t+1$ o algoritmo aplica a correção

$$\Delta \mathbf{w} = \mathbf{w}_{t+1} - \mathbf{w}_t$$

$$\Delta \mathbf{w} = -\eta \mathbf{g}_t \tag{2.17}$$

Cada vez que um parâmetro ($\mathbf{w}$) é atualizado, todos os conjuntos de dados de amostras de treinamento são usados, mas se forem grandes acarretam em longo tempo de treinamento e baixa velocidade de convergência do método, ou seja, converge lentamente para a solução ótima ($\mathbf{w}*$). Uma iteração do algoritmo é chamada de um lote e esta forma de descida do gradiente é referida como descida do gradiente em lote (*Batch Gradient Descent*).

Descida do gradiente estocástico

A descida do gradiente estocástico (do inglês, *stochastic gradient descent* - SGD) é um metodo de otimização baseado no método gradiente. A palavra estocástico carrega consigo o significado de um sistema ou processo que está vinculado a uma probabilidade aleatória e foram introduzidas em Robbins e Monro (1951). Diferente do método gradiente, descrito anteriormente, a otimização SGD é baseada em uma única amostra por vez. Consideremos o par $(\mathbf{x}_i, d_i)$ amostrado aleatoriamente do treinamento, a atualização dos parâmetros é dada (Bottou, 1991) como:

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \eta \nabla J(\mathbf{w}_t; \mathbf{x}_i, d_i) \tag{2.18}$$

Desse modo o processo de treinamento torna-se bastante rápido e devido as amostras estocásticas, o algoritmo ajuda a função custo a não ficar presa em um mínimo local como no método gradiente. Entretanto, processo iterativo é menos regular (mais ruidoso) que o método gradiente e o resultado final não é precisamente a solução ótima ($\mathbf{w}*$), mas algo perto do mínimo para função custo. Isso significa que, em média, o gradiente estocástico é uma boa estimativa do gradiente (ver Figura 2.8).

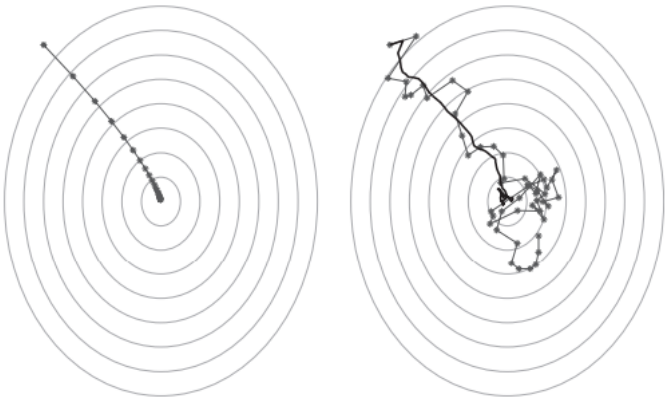


Figura 2.8: Uma ilustração do algoritmo de descida do gradiente (esquerda) e o algoritmo SGD (direita). A função a ser minimizada foi $1,25(x+6)^2+(y-8)^2$. Para o caso estocástico, a linha sólida representa o valor médio de $\mathbf{w}$ (Shalev-Shwartz e Ben-David, 2014).

Descida do Gradiente *Mini-batch*

Este tipo de otimização é uma combinação entre as duas versões anterios baseadas na descida do gradiente. Aqui o conjunto de treinamento é dividido em vários subconjuntos (*mini-batches*) com elementos aleatórios, onde os pesos do modelo são ajustados a cada *mini-batch*.

De modo que apresenta convergência mais rápida do que o descida do gradiente em lote e mais lenta do que o SGD.

Momento

Este algoritmo é um esquema de ajuste do passo de aprendizagem, η, simples e que ajuda a acelerar a convergência do método descida do gradiente e amortece as oscilações (Rumelhart, Hinton e Williams, 1986). Com este método, a equação 2.16 assume a forma (Qian, 1999):

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \eta \mathbf{g}_t + \gamma(\mathbf{w}_t - \mathbf{w}_{t-1}) \tag{2.19}$$

onde γ é o parâmetro de momento, escolhido no intervalo $[0, 1[$ o que torna um fator de decaimento exponencial.

Adam

O método estimativa de momento adaptável (em inglês, *Adaptive moment estimation* - Adam) é uma técnica que calcula as taxas de aprendizado adaptativo (Kingma e Ba, 2014), de maneira a armazenar uma média exponencialmente decrescente de gradientes quadrados passados ($\mathbf{g}_t^2$), como Adadelta (Zeiler, 2012) e RMSprop (Tieleman e Hinton, 2012) e, além disso, mantém uma média exponencialmente decrescente de gradientes passados ($\mathbf{g}_t$), semelhante ao momento. Estas duas medidas, $\mathbf{v}_t$ e $\mathbf{m}_t$, são obtidas da seguinte maneira

$$\mathbf{m}_t = \beta_1 \mathbf{m}_{t-1} + (1 - \beta_1)\mathbf{g}_t$$

$$\mathbf{v}_t = \beta_2 \mathbf{v}_{t-1} + (1 - \beta_2)\mathbf{g}_t^2 \tag{2.20}$$

onde $\mathbf{m}_t$ e $\mathbf{v}_t$ são estimativas do primeiro momento (a média) e do segundo momento (a variância não centrada) dos gradientes, respectivamente. Entretanto, $\mathbf{m}_t$ e $\mathbf{v}_t$ são inicializados como vetores nulos, de modo que os autores notaram que as estimativas dos momentos são inicialmente enviesadas. A solução proposta por eles foi modificar para:

$$\hat{\mathbf{m}}_t = \frac{\mathbf{m}_t}{1 - \beta_1^t}$$

$$\hat{\mathbf{v}}_t = \frac{\mathbf{v}_t}{1 - \beta_2^t} \tag{2.21}$$

de maneira a neutralizar esses vieses calculando estimativas de primeiro e segundo momento corrigidas.

Desta forma a atualização dos parâmetros baseado nessa metodologia é dada por

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \frac{\eta}{\sqrt{\hat{\mathbf{v}}_t} + \epsilon}\hat{\mathbf{m}}_t \tag{2.22}$$

Os autores propõem valores padrão de $0,9$ para β_1, $0,999$ para β_2 e 10^{-8} para ϵ. Apesar de bastante recente, o método de otimização Adam é o mais utilizado no treinamento de redes neurais atualmente.

2.3.5 *Backpropagation*

Na seção anterior, vimos como as redes neurais podem aprender seus pesos sinápticos ($\mathbf{w}$) usando o algoritmo de otimização baseados no gradiente descendente, mas não abordamos, no contexto do aprendizado de máquina, como obter o gradiente da função custo. O cálculo desse gradiente é obtido via algoritmo *backpropagation* (Rumelhart, Hinton e Williams, 1986), que é indiscutivelmente o algoritmo mais importante na história das redes neurais, tornando-se o principal motor do universo *Deep Learning*, sem ele seria praticamente impossível treinar tais redes de apredizagem profundas. O algoritmo de backpropagation consiste em duas fases:

1. O passo para frente (*forward pass*): entradas são passadas através da rede e as previsões de saída obtidas;

2. O passo para trás (*backward pass*): cálculo do gradiente da função custo na camada final da rede e utilização desse gradiente para aplicar recursivamente a regra da cadeia (*chain rule*) para atualizar os pesos da rede (também conhecida como retropropagação).

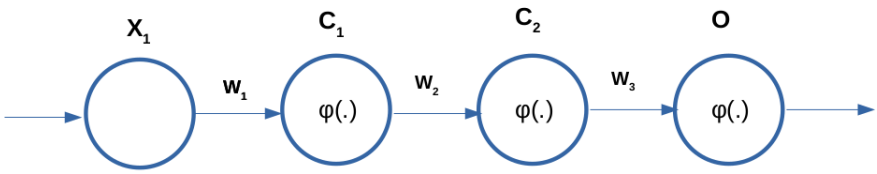


Figura 2.9: Rede neural com duas camadas ocultas.

Para demonstrar, de maneira simples, a ideia do algoritmo *backpropagation*, tomemos como exemplo a rede neural da Figura 2.9. Neste exemplo, X_1 representa uma entrada, W_1, W_2 e W_3 os pesos sinápticos, C_1 e C_2 *hidden layers*, $\varphi(\cdot)$ uma função de ativação e a saída da rede representada por O. Como pode ser verificado, cada neurônio é uma função do anterior conectado a ele. De maneira que, ao alterar o valor de W_1, os neurônios C_1 e C_2, e consequentemente a saída, também mudam. Devido a essa noção de dependências funcionais, podemos representar matematicamente a saída:

$$O = \varphi(W_3 * C_2)$$

$$C_2 = \varphi(W_2 * C_1)$$

$$C_1 = \varphi(W_1 * X_1) \tag{2.23}$$

ou simplesmente:

$$O = \varphi(W_3 * \varphi(W_2 * \varphi(W_1 * X_1))) \tag{2.24}$$

Ao calcular a derivada da função 2.24 com relação a algum peso arbitrário (por exemplo, W_1), sendo $\varphi(\cdot)$ uma função diferenciável, através da regra da cadeia, obtemos:

$$\frac{\partial O}{\partial W_1} = \frac{\partial O}{\partial C_2}\frac{\partial C_2}{\partial C_1}\frac{\partial C_1}{\partial W_1} \tag{2.25}$$

Agora, vamos anexar mais uma operação a nossa rede neural da Figura 2.9. Esta operação irá calcular e retornar o erro (usando a função custo) da nossa saída como mostra a Figura 2.10, caraterizando o passo para frente.

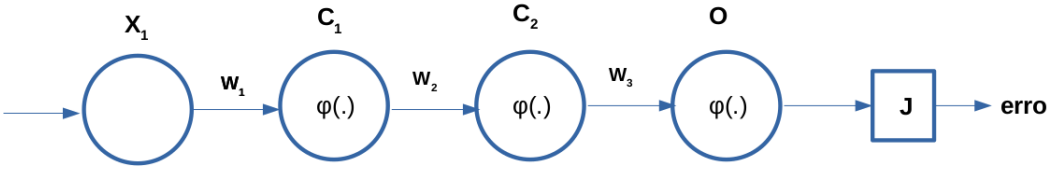


Figura 2.10: Inserção do cálculo do erro na saída da rede neural.

No entanto, se desejássemos calcular a derivada do erro com qualquer peso arbitrário (novamente, W_1), o resultado seria:

$$\frac{\partial erro}{\partial W_1} = \frac{\partial erro}{\partial O}\frac{\partial O}{\partial C_2}\frac{\partial C_2}{\partial C_1}\frac{\partial C_1}{\partial W_1} \tag{2.26}$$

Desta forma, podemos calcular as derivadas do erro para todos os outros pesos na rede e aplicar o gradiente descendente para atualizar os pesos da rede, configurando o passo para trás. Deste modo, definimos a retropropagação (*backpropagation*), pois estamos usando o erro de saída para atualizar os pesos, via regra da cadeia, no intuito de otimizar os pesos para que a rede neural possa aprender a mapear corretamente as entradas para as saídas.

Entendido o conceito abordado anteriormente, podemos resumir as etapas para o treinamento/aprendizado de uma rede neural, para um lote de dados, da seguinte maneira:

1. Inicialização: limpeza dos gradientes de todas as variáveis otimizadas;
2. Passo para frente: cálculo das saídas previstas passando as entradas para o modelo;
3. Cálculo do erro (através da função custo);
4. Passo para trás: cálculo do gradiente da função custo em relação aos parâmetros do modelo;
5. Atualização dos parâmetros (etapa de otimização);
6. Atualizar a perda média de treinamento.

O processo é feito iterativamente apresentando novas épocas de exemplos de treinamento para a rede, até que seja satisfeito algum critério de parada.

3

FWI usando técnicas de *Deep Learning*

Nos capítulos anteriores discutimos, de forma geral, as bases de entedimento para avançarmos ainda mais e entrelaçar a capacidade das arquiteturas das redes neurais profundas (em inglês, *Deep Neural Network* - DNN) em lidar com problemas não-lineares complexos, como a inversão de velocidade sísmica. Neste capítulo trataremos da FWI numa abordagem usando rede neural, tendo em vista que este tipo de implementação pode se beneficiar de algumas ferramentas embutidas nos ambiente de aprendizado de máquina (TensorFlow, PyTorch), como a diferenciação automática para obtenção do gradiente, funções custo, otimizadores (Adam, SGD, Momento) e estratégia de treinamento em lotes (*mini-bacth*).

Como pontapé inicial iremos mostrar um estudo de caso de FWI 1D utilizando treinamento de uma rede neural profunda, que exemplifica um aprendizado supervisionado numa rede neural *feedforward* com múltiplas camadas (descritas na seção 2.2.2). Tendo em vista limitações inerentes a este tipo de arquitetura, como a necessidade de dados rotulados, vamos nos valer da analogia que pode ser feita entre as arquiteturas de rede que envolvem retroalimentação e a modelagem a partir da equação da onda acústica, no intuito de incluir a física como guia no processo de aprendizagem de máquina, não supervisionada, de modo a combinar o conhecimento que dispomos sobre propagação de ondas e as vantagens do aprendizado de máquina, principal foco deste trabalho.

3.1 Inversão 1D como treinamento de uma rede *feed-forward*

Uma das aplicações que tem sido explorada no aprendizado de máquina em sísmica, especificamente em inversão, é o uso de uma rede neural profunda (DNN) para produzir um modelo de velocidades a partir de dados sísmicos. Nesta seção utilizaremos um conjunto de dados rótulados (velocidades sísmicas, traço sísmico) para treinar uma rede neural com múltiplas camadas de modo a produzir velocidades sísmicas. Este exemplo é meramente uma reprodução da rede neural de autoria de Richardson (2017). A única alteração feita deu-se no código de modelagem sísmica utilizado para gerar o conjunto de dados sísmicos, originalmente escrito em linguagem *Fortran*, e transcrito para *Python* dada maior familiaridade do autor deste trabalho com a referida linguagem.

3.1.1 Dados de Entrada

O conjunto de dados 1D utilizados como entrada para alimentar está rede neural foram os pares modelo de velocidades e traço sísmico. Cada modelo consiste em duas velocidades escolhidas aleatoriamente: uma para a parte superior do modelo e outra para a parte inferior, e a localização da transição entre elas (a profundidade do refletor) também é escolhida aleatoriamente. Os valores mínimo e máximo de velocidades, possíveis em cada modelo, são de $1500\,m/s$ a $5000\,m/s$, respectivamente. De forma que cada modelo dá origem a um traço sísmico (ver Figura 3.1) que possue um único refletor, livre de ruído, obtido através de modelagem numérica 1D por diferenças finitas, com um esquema de segunda ordem no domínio do tempo e de oitava ordem no espaço, e uma fonte *Ricker* com frequência dominante de $50\,Hz$.

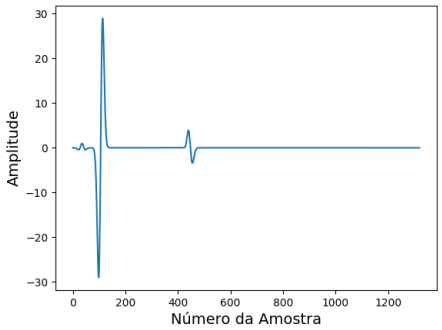


Figura 3.1: Exemplo de traço sísmico fornecido a rede neural.

A Figura 3.2 mostram uma visão ampla dos dados apresentados a rede neural e a Figura 3.3 exibe uma amostra dos modelos de velocidade, que espera-se que a rede neural aprenda

a obter quando receber os dados como os da Figura 3.2.

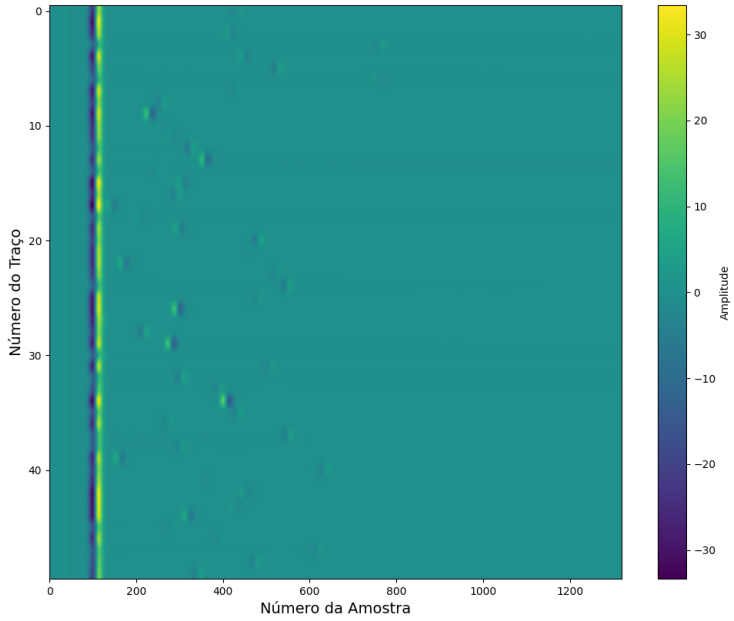


Figura 3.2: Amostra de traços sísmicos fornecidos à rede.

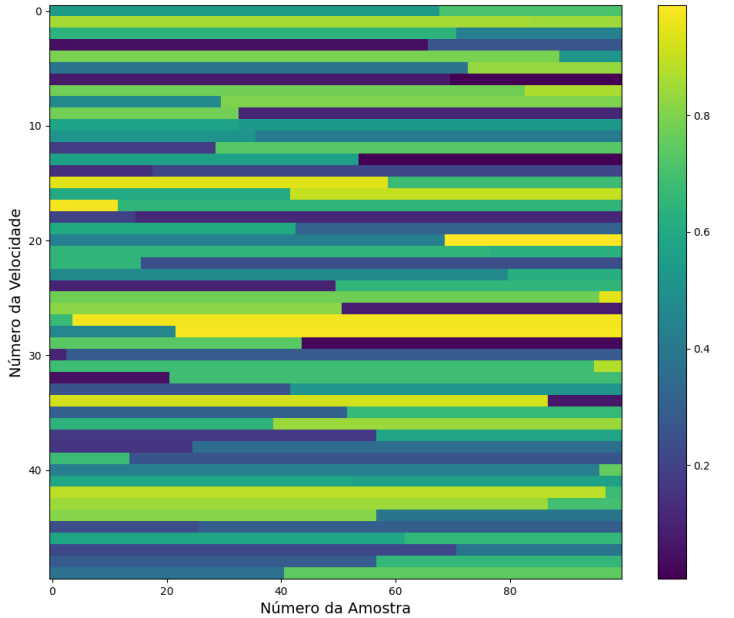


Figura 3.3: Amostra de velocidades (normalizadas) usadas no treinamento da rede neural.

3.1.2 Treinamento

Como citado anteriormente, neste treinamento foi utilizada um rede *feedforward* com múltiplas camadas, ou MLP, composta por oito camadas totalmente conectadas, sendo as sete primeiras com 250 unidades (neurônios) cada e 1 camada de saída que fornece a velocidade calculada pela rede.

Para este treinamento foi gerado um conjunto com 2000 pares de dados (velocidade - traço), deste total 95% foram utilizados para treinar a rede. Este primeiro subconjunto de dados são chamados de dados de treinamento, que confere a rede a capacidade de generalizar as relações estudadas mesmo para observações nas quais ele não foi treinado. O restante dos dados (5%), que a rede não recebeu durante o treinamento, compõem o subconjunto denominado de dados de validação ou também dados de teste. De acordo com as orientações fornecidas por Richardson (2017), fizemos um processamento de normalização das velocidades.

3.1.3 Resultados

As Figuras 3.4 e 3.5 mostram os resultados fornecidos pela rede neural (curva azul) para dados que a rede não foi treinada, ou seja, a rede nunca viu esses dados antes. Como o objetivo da rede era receber os dados gravados como entrada e, em seguida, tentar adivinhar o modelo de velocidade, pode se verificar que nesse exemplos os resultado foram bons.

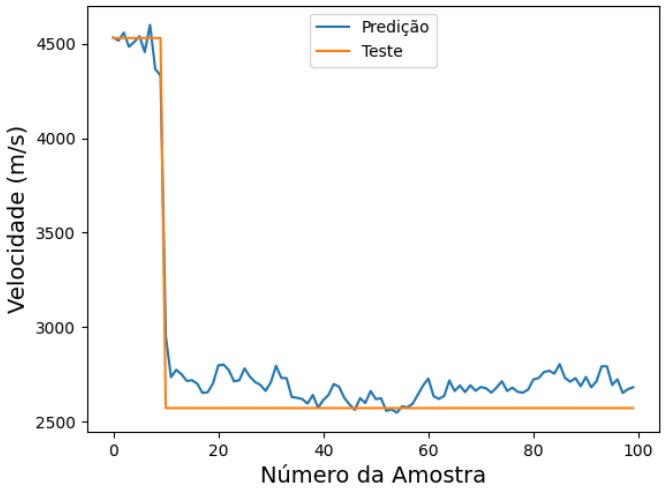


Figura 3.4: Resultado fornecido pela rede NNFWI 1D.

De uma maneira mais ampla, na Figura 3.6, o primeiro gráfico mostra os modelos previstos pela rede e o segundo os modelos verdadeiros. No geral, observa-se que há um ajuste

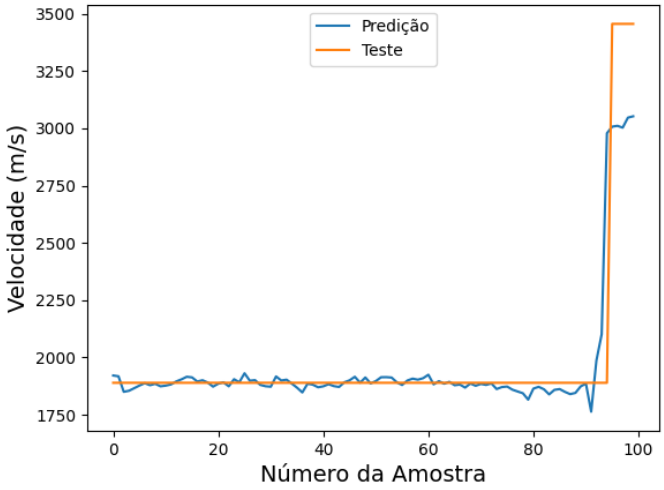


Figura 3.5: Outro resultado fornecido pela rede NNFWI 1D.

muito bom entre eles. Então, certamente é possível escolher exemplos em que o método funcionou muito bem e exemplos em que não funcionou bem, mas o resultado médio demonstra que a metodologia funciona razoavelmente bem.

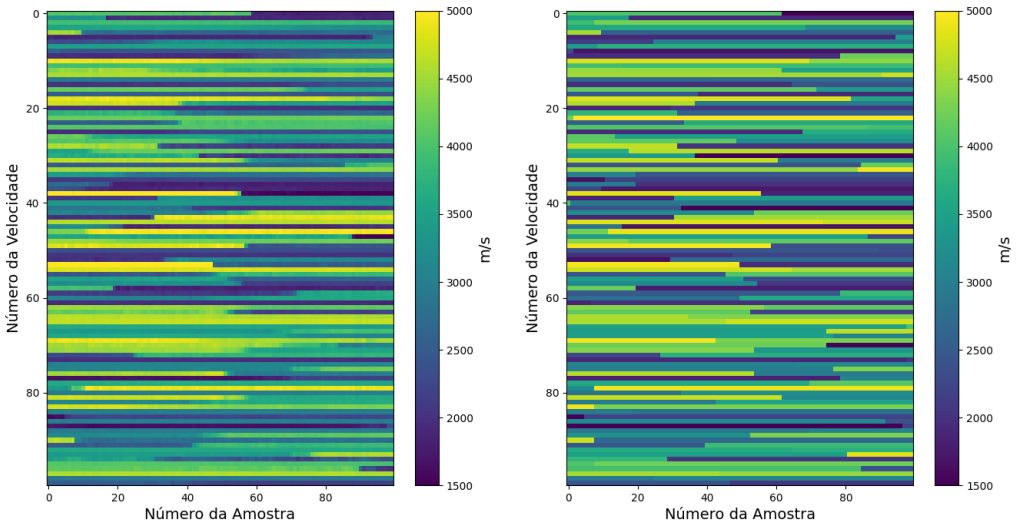


Figura 3.6: Modelos de velocidades previstos pela rede (esquerda) e modelos de velocidades verdadeiros (direita).

Tais resultados corroboram com a popularidade emergente das técnicas de aprendizado profundo, em que a rede neural profunda (DNN) demonstra uma capacidade impressionante em lidar com problemas não-lineares complexos, incluindo a inversão de velocidade sísmica. Entretanto, na prática, devido à dependência de um conjunto de dados rotulados, ou seja, o modelo de velocidade real quase inacessível correspondente aos dados sísmicos reais, esta

abordagem de inversão de aprendizagem profunda supervisionada pode sofrer de limitações na generalização. Uma alternativa para mitigar esse problema é impor a física governante a esse tipo de método puramente baseado em dados, como será abordado na proxima seção. A ideia é mostrar que a modelagem direta baseada na equação da onda pode ser feita através de uma RNN. De maneira que ao tratarmos os dados observados e sua posições de disparo como pares de treinamento, seguindo os procedimentos de inversão de forma de onda completa tradicional, o modelo de velocidade invertido pode ser obtido como o parâmetro treinável de rede.

3.2 Modelagem da onda acústica como uma rede neural recorrente

Como já mencionado anteriormente, as RNNs são uma classe de rede neural artificial com laço de retroalimentação, que dentro do seu espectro de utilização, a exemplo da previsão de séries temporais, nos permite tratar problemas de modelagem sísmica direta. Essa propriedade nos possibilita simular a propagação do campo de ondas dentro de uma RNN treinável. Assim, voltemos a equação 1.3, reescrita na forma:

$$\frac{\partial^2 \mathrm{u}(\mathbf{r},\mathrm{t})}{\partial \mathrm{t}^2} = -L^2 \mathrm{u}(\mathbf{r},\mathrm{t}) + \mathrm{s}(\mathbf{r},\mathrm{t}), \tag{3.1}$$

onde $-L^2 = v^2(\mathbf{r})\nabla^2$, cuja solução numérica discreta pode ser obtida da seguinte forma:

$$\begin{cases} u^{n+1} = 2\cos(L\Delta t)u^n - u^{n-1} + s^n \\ u^n = u^n \end{cases} \tag{3.2}$$

que na forma matricial assume a seguinte forma:

$$\begin{bmatrix} u^{n+1} \\ u^n \end{bmatrix} = \begin{bmatrix} 2\cos(L\Delta t) & -I \\ I & 0 \end{bmatrix} \times \begin{bmatrix} u^n \\ u^{n-1} \end{bmatrix} + \begin{bmatrix} s^n \\ 0 \end{bmatrix} \tag{3.3}$$

e na forma de uma RNN como:

$$\begin{cases} h_t = W^{(h)} h_{t-1} + W^{(x)} x_t \\ y_t = W^{(y)} h_t \end{cases} \tag{3.4}$$

onde

$$\begin{cases} h_t = \begin{bmatrix} u^{n+1} \\ u^n \end{bmatrix} \\ W^{(h)} = \begin{bmatrix} 2\cos(L\Delta t) & -I \\ I & 0 \end{bmatrix} \\ h_{t-1} = \begin{bmatrix} u^n \\ u^{n-1} \end{bmatrix} \\ x_t = \begin{bmatrix} s^n \\ 0 \end{bmatrix} \end{cases}$$

Os termos $W^{(x)}$, $W^{(y)}$ e $W^{(h)}$ são as chamadas de matrizes de peso. Na rede descrita acima $W^{(x)}$ é a matriz de peso para injetar a fonte, $W^{(y)}$ a matriz de amostragem dos dados e $W^{(h)}$ a matriz que descreve a atualização dos campos de ondas em u^n e u^{n-1}. Nessa estrutura de rede treinável, apenas o peso $W^{(h)}$ precisa ser treinado para encontrar o campo de velocidade que melhor se ajusta aos dados observados, enquanto as outras matrizes não são alteradas pelo processo de treinamento (Hughes et al., 2019).

3.3 Inversão como treinamento de uma rede recorrente

Assim como os algoritmos tradicionais de FWI que minimizam os resíduos de dados para estimar um modelo de velocidade, com base na suposição de que o modelo atualizado é a soma de um modelo de fundo e uma perturbação do modelo estimado (ver Equação 1.21), o processo de inversão de forma de onda com base no treinamento de uma rede neural recorrente é semelhante, mas com os dados calculados fornecedidos pela saída da rede recorrente e os gradientes calculados por diferenciação automática ao invés do método de estado adjunto. A diferenciação automática calcula o gradiente da função custo de erro quadrático de forma equivalente ao procedimento realizado no método adjunto (ver Apêndice B).

É evidente a semelhança entre aprendizado profundo e FWI, e a vantagem de combiná-los reside na facilidade de uso dos *softwares* de aprendizado profundo (*PyTorch*, *TensorFlow*), que permite que implementações sofisticadas sejam rapidamente desenvolvidas, pois estes propiciam ferramentas necessárias para treinar modelos de aprendizado de máquina com eficiência e controle. Uma vez que um algoritmo é expresso usando esses ambientes, ele pode ser executado em diferentes tipos de unidades de processamento (CPUs, GPUs) utilizando recursos disponíveis em um *cluster* (processamento paralelo, memória distribuída), além dos já comentados otimizadores e funções custo.

3.4 Abordagem Multiescala

Tanto a FWI convencional quanto a FWI baseada em técnicas de *deep learning*, estão sujeitas à convergência para mínimos locais, devido à alta não-linearidade inerente ao problema inverso. Até mesmo a escolha de um modelo de velocidades inicial próximo ao mínimo global não garantirá a convergência da função objetivo para um modelo ótimo com sentido geológico. Uma das causas atribuídas ao problema pode ser explicada pelo salto de ciclo (*cycle skipping*).

De acordo com Virieux e Operto (2009), o salto de ciclo ocorre quando existe uma diferença de fase maior que meio período do comprimento de onda (Figura 3.7), entre o dado calculado e observado, de modo que o método de otimização ajuste o dado calculado a um ciclo do traço sísmico observado, com defasagem de um ou mais comprimentos de onda. De forma que, o acúmulo dos ajustes entre dados defasados, poderá carrear para modelo de velocidades, resultante da inversão, com estruturas espúrias sem significado geológico ou até mesmo estruturas que geologicamente fazem sentido, mas que não condizem com a realidade. Dessa forma, é preciso fazer uso de estratégias de inversão capazes de contornar o problema de salto de ciclo e, consequentemente, reduzir as chances de convergência para mínimos locais, a um custo computacional aceitável, como é o caso da abordagem multiescala proposta por Bunks et al. (1995).

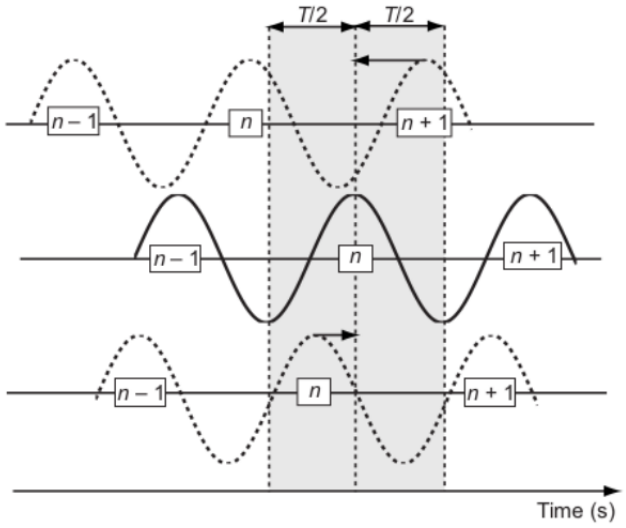


Figura 3.7: Formação de artefatos de salto de ciclo, para uma componente de onda monocromática, na FWI. A linha sólida representa uma componente do dado observado, e as linhas pontilhadas, do dado modelado, ambas com um delay maior que $T/2$. No caso da modelagem superior, a FWI vai atualizar o modelo como se houvesse correlação entre o ciclo $(n+1)$ dos sismogramas modelados e o ciclo n do dado observado, gerando um modelo errôneo (Virieux e Operto, 2009).

A ilustração da abordagem multiescala pode ser visualizada na Figura 3.8, onde as escalas maiores do problema, que representam as baixas frequências, apresentam uma função objetivo mais suave, sujeitas a poucos mínimos locais. A medida que se caminha na direção de diminuição da escala do problema, ou seja, para as altas frequências, o número de mínimos locais aumenta, porém o modelo inicial já está mais próximo do mínimo global, devido às rodadas anteriores como pode ser melhor visualizado na Figura 3.9. Cabe ressaltar que este tipo de abordagem não consiste em um método de inversão, mas sim uma maneira de contornar o problema do salto de ciclo, a fim de reduzir as chances de convergência para mínimos locais.

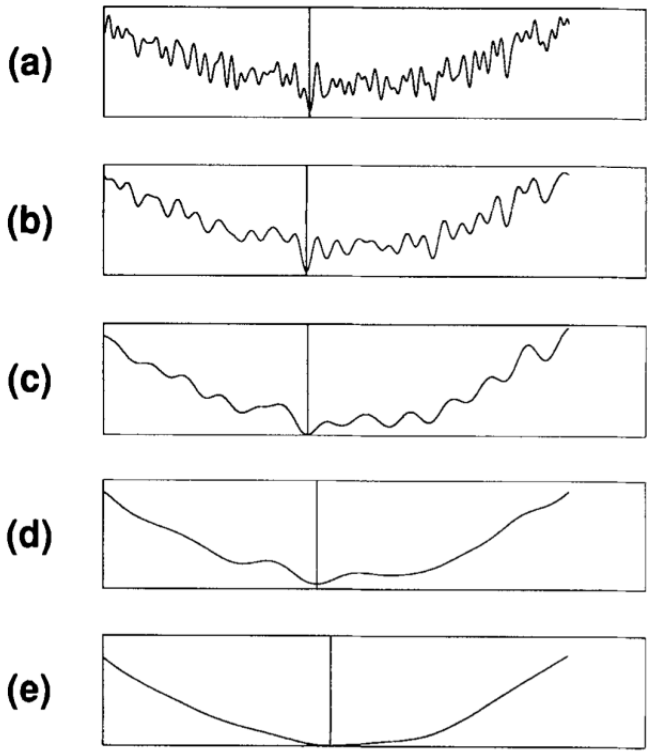


Figura 3.8: Interpretação heurística da abordagem multiescala. Cada painel representa o gráfico de uma função objetivo, desde a menor escala (maior frequência) (a) até a maior escala (menor frequência) (e) (Bunks et al., 1995).

Em suma, na estratégia apresentada por Bunks et al. (1995), a escala é a escala do comprimento de onda λ, mas relembrando a relação $\lambda = v/f$, esta também nos permite separar as escalas por meio de bandas de frequência. De modo que, no domínio do tempo, a seleção da banda de frequência requer uma filtragem no dado observado, a fim de delimitar a faixa que será utilizada em cada etapa. Também se faz necessário garantir que o dado modelado tenha o mesmo conteúdo de frequência do dado observado, para que o resíduo entre eles seja relevante. Uma forma de garantir isso é filtrando a fonte sísmica utilizada na modelagem acústica de maneira que tenha o mesmo conteúdo de frequência do dado

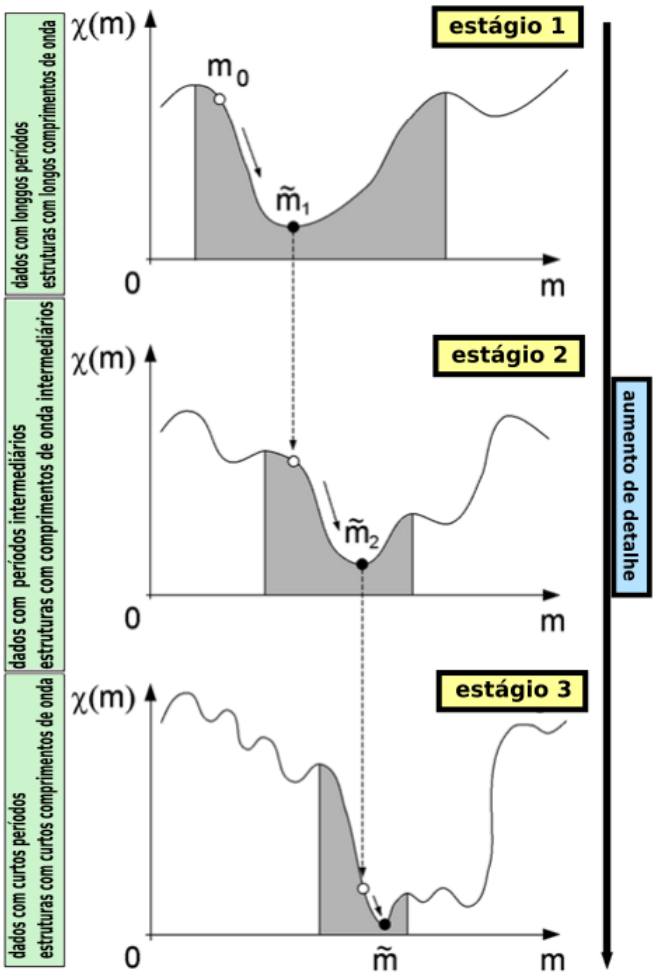


Figura 3.9: Representação do comportamento da função objetivo no sentido da maior escala (gráfico superior) para menor escala (gráfico inferior). Adaptado de (Fichtner, 2011)

observado filtrado. Dessa forma, os dados modelados não necessitarão ser filtrados novamente (Bunks et al., 1995).

Dentre os diversos tipos de filtros de frequência, que cumprem a tarefa de limitar a banda de frequência de um sinal, o escolhido para este trabalho foi o filtro de Wiener que é definido no domínio da frequência (Boonyasiriwat et al., 2009) por:

$$f_{Wiener}(\omega) = \frac{W_{alvo}(\omega) W^*_{original}(\omega)}{|W_{original}(\omega)|^2 + \epsilon^2}, \tag{3.5}$$

onde f_{Wiener} é o filtro que será aplicado ao dado, ω a frequência temporal, $W_{original}$ é a *wavelet* original do dado, W_{alvo} é a *wavelet* com conteúdo de frequência limitado, $*$ é o operador complexo conjugado e ϵ é um fator pequeno usado para estabilizar o resultado. Neste trabalho, para cada banda de frequência, o fator f_{Wiener}, implementado em linguagem Python, foi aplicado traço a traço.

Na prática a FWI no domínio do tempo utilizando a abordagem multiescala pode então ser resumida (Koehne, 2014) nos seguintes passos:

1. Para a banda de frequências atual:
   (a) Filtrar o dado observado;
   (b) Utilizar uma *wavelet* com frequência máxima correspondente à banda atual, para gerar os dados modelados;
   (c) Realizar k iterações da FWI;
2. Definir o resultado da rodada atual como modelo inicial;
3. Aumentar a banda de frequências.

Esse ciclo é repetido até que o número de bandas de frequência planejado tenha sido atingido ou algum outro critério de parada tenha sido satisfeito. A abordagem multiescala também é aplicável à FWI formulada como treinamento de uma rede recorrente, e os resultados obtidos com base nessa metodologia serão expostos no próximo capítulo.

4

Metodologia e Resultados

Neste capítulo serão apresentados os resultados obtidos durante o desenvolvimento desta pesquisa. Na primeira parte, será discutida a metodologia utilizada no processo de inversão. Em seguida, serão apresentados os resultados com e sem a abordagem multiescala para diferentes modelos geológicos. Além disso serão comparados os resultados obtidos através de dois otimizadores distintos, o Adam e o Momento, na solução do problema inverso. Para fechar o capítulo serão mostrados as respostas da abordagem adotada na inversão sísmica quanto ao modelo inicial fornecido como entrada no processo.

4.1 Metodologia

Nossos experimentos foram realizados utilizando modelos sintéticos 2D. A otimização foi realizada remotamente utilizando os recursos computacionais do SENAI/CIMATEC (GPU Nvidia V100 NVLink) e implementado no PyTorch, especialmente o pacote de ferramentas chamado *Deepwave* (Richardson, 2021) [1], que fornece módulos de propagação de ondas e permite calcular gradientes em relação ao nosso modelo automaticamente via operações em cadeia e a retropropagação. O módulo de propagação atua como a rede recorrente de Elman que incorpora a equação da onda acústica em sua formulação por meio da técnica das diferenças finitas e é capaz de realizar modelagem sísmica. Como tratamos de dados puramente sintéticos, a fonte é conhecida e foi desconsiderada como um parâmetro a ser estimado no processo de inversão.

1. https://github.com/ar4/deepwave

4.1.1 Preparação dos Dados

Para ilustrar a aplicabilidade do método proposto, testamos o desempenho da inversão para três modelos sintéticos que são adequados para pesquisas geofísicas. São eles: um pequeno trecho do modelo SEAM Fase I na porção sedimentar, o modelo SEAM Fase I completo e o modelo Marmousi. Para evitar problemas com a atual limitação de equipamentos computacionais, todos os modelos foram reamostrados.

SEAM Fase I - Porção Sedimentar

Em 2007, um conjunto de 24 empresas se uniram para criar o modelo SEAM Fase I (Fehler e Keliher, 2011), distribuído no ano de 2014, com o objetivo de abordar os desafios da imagem do pré-sal em bacias terciárias, com ênfase nas águas profundas do Golfo do México. Originalmente o modelo possui 1751 amostras na direção horizontal e 1501 amostras na direção vertical, com intervalo de amostragem de $20\,m$ e $10\,m$, respectivamente. A faixa de velocidade varia de $1490\,m/s$ a $4800\,m/s$.

Em nosso primeiro teste, utilizamos somente um pequeno trecho do modelo original, cujo modelo de velocidade da onda P foi reamostrado, de modo que as dimensões resultantes foram de 500 amostras na direção horizontal e 251 amostras na direção vertical, com incremento espacial de $5\,m$ em ambas as direções. A velocidade varia de $1490\,m/s$ a $2830\,m/s$. O modelo adotado como verdadeiro é mostrado na Figura 4.1, e vamos aqui denominá-lo de SEAM Fase I Sedimentar.

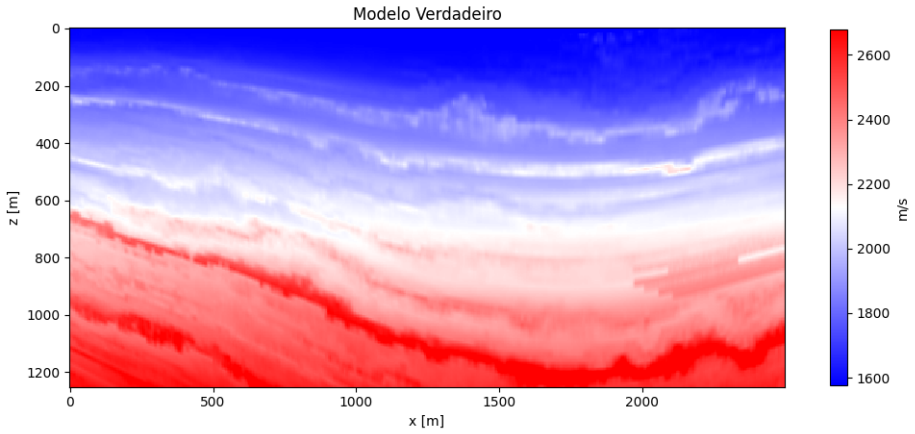


Figura 4.1: Modelo de velocidades verdadeiro SEAM Fase I Sedimentar.

SEAM Fase I

Na sequência estendemos o estudo da FWI para todo o modelo, novamente reamostrado, no intuito de incluir complexidade geológica. De maneira que as dimensões resultantes foram de 576 amostras na direção horizontal e 351 amostras na direção vertical, com intervalo de amostragem de 12 m em ambas as direções. A velocidade também varia de 1490 m/s a 4800 m/s. O modelo tido como verdadeiro é mostrado na Figura 4.2. Devido à amostragem espacial adotada, as dimensões finais também mudaram em relação a original.

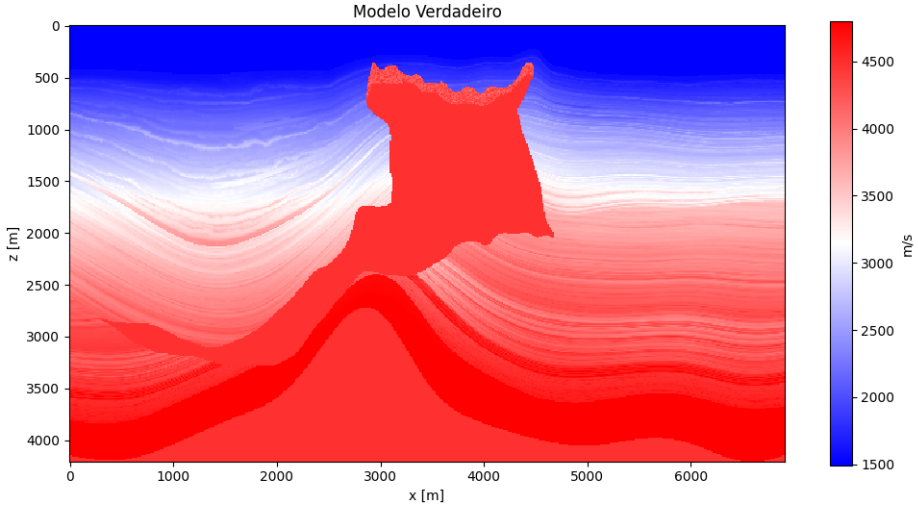


Figura 4.2: Modelo de velocidades verdadeiro SEAM Fase I.

Marmousi

O modelo Marmousi foi criado pelo Instituto Francês do Petróleo em 1988 (Bourgeois et al., 1991) e a sua geometria foi desenhada com base em perfis sísmicos na Bacia do Cuanza, Angola. É um dos modelos de velocidade acústica mais conhecidos e frequentemente encontrado nas pesquisas de geofísica de exploração. Reamostramos o modelo orginal de velocidade da onda P, que resultou no modelo com 575 amostras na direção horizontal e 250 amostras na direção vertical, cujos incrementos de amostragem foram de 12 m e 16 m, respectivamente. A velocidade varia de 1500 m/s a 5500 m/s. O modelo verdadeiro é mostrado na Figura 4.3. Novamente, devido à amostragem espacial adotada, as dimensões finais diferem em relação a original.

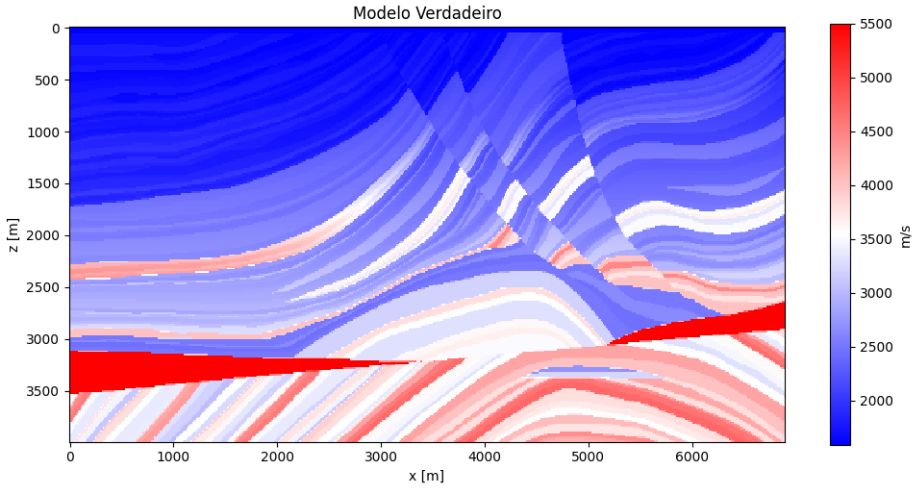


Figura 4.3: Modelo de velocidades verdadeiro Marmousi.

4.1.2 Definição dos Parâmetros

A simulação da propagação das ondas via modelagem acústica, foi baseada no método das diferenças finitas com um esquema de segunda ordem no domínio do tempo e de quarta ordem no espaço. Para evitar reflexões indesejadas das bordas do domínio de simulação, usamos uma camada perfeitamente combinada (em inglês, *perfectly matched layer* - PML) (Pasalic e Mcgarry, 2010) como condição de contorno absorvente. Uma *wavelet* de Ricker foi escolhida como a fonte verdadeira e a frequência de pico, assim como os demais parâmetros da modelagem, podem ser visualizados na Tabela 4.1. Os receptores foram dispostos uniformemente na superfície e a quantidade é igual à largura do modelo. Para a inversão multiescala as bandas de frequências serão explicitadas na seção de resultados.

Parâmetros	SEAM Fase I Sedimentar	SEAM Fase I	Marmousi
f_{peak} (Hz)	25,0	12,0	12,0
Intervalo de Amostragem (ms)	4,0	4,0	4,0
Tempo de Registro (s)	2,0	5,0	4,0
dx (m)	5,0	12,0	12,0
dz (m)	5,0	12,0	16,0
Intervalo de tiro (m)	25,0	34,5	69,0
Intervalo receptores (m)	10,0	12,0	12,0
Número de Tiros	100	200	100

Tabela 4.1: Tabela de parâmetros utilizados na modelagem sísmica.

No processo de inversão (FWI), beneficiou-se da diferenciação automática para o cálculo do gradiente, uso dos otimizadores (ver Tabela 4.2) e estratégia de dados de mini-lotes do PyTorch. O tamanho do lote foi definido para cada experimento (ver Tabela 4.3), de maneira que o número total de fontes (tiros) dividido pelo tamanho do respectivo lote resulta na quantidade de tiros comuns em cada lote. A estimativa inicial do modelo foi obtida através da suavização do modelo verdadeiro pela função gaussiana. De acordo com as orientações fornecidas por Richardson (2021), fizemos um processamento de normalização das amplitudes que consiste na divisão pelo máximo valor absoluto, tanto para os dados observados quanto para os dados de mini-lotes simulados.

Otimizador(parâmetro(s))	SEAM Fase I Sedimentar	SEAM Fase I	Marmousi
Momento (γ)	0,9	0,9	0,9
Adam ($\beta_1/\beta_2/\epsilon$)	0,9/ 0,999/ 10^{-8}	0,9/ 0,999/ 10^{-8}	0,9/ 0,999/ 10^{-8}

Tabela 4.2: Parâmetros dos Otimizadores.

Quanto a definição da taxa de aprendizagem, o critério de escolha foi baseado em valores obtidos na literatura e testados em cada experimento.

Parâmetro	SEAM Fase I Sedimentar	SEAM Fase I	Marmousi
Tamanho do lote	10	40	20

Tabela 4.3: Tamanho do lote em cada experimento.

4.2 Resultados

Nesta seção iremos exibir os resultados de inversão FWI com a banda de frequências completa comparados com a inversão FWI com a abordagem multiescala, para os modelos propostos. Para simplificação chamaremos o método convencional apenas de FWI, enquanto que para se referir a abordagem multiescala utilizaremos o termo FWI multiescala. Além disso iremos comparar os resultados da inversão multiescala ao alterarmos o tipo de otimizidor envolvido no processo, também para todos os modelos propostos. Por último, vamos evidenciar a sensibilidade ao modelo inicial nas duas abordagens de inversão.

4.2.1 Resultados do Modelo SEAM Fase I Sedimentar

Em termos de configuração a FWI foi obtida utilizando uma frequência dominante de 25 Hz, após 30 épocas com o otimizador Adam e taxa de aprendizagem (η) 10. Para a FWI multi-escala, foram escolhidas arbitrariamente as frequências dominantes de 5 Hz, 10 Hz, 15 Hz, 20 Hz e 25 Hz. Em cada banda de frequências foram feitas 30 épocas utilizando o método de otimização Adam e taxa de aprendizagem 10. O modelo inicial foi obtido a partir da suavização do modelo verdadeiro. Tal suavização foi feita via convolução com um filtro gaussiano, cuja eficiência da atenuação depende do tamanho σ do *kernel* gaussiano, também chamado de desvio padrão do filtro. Para este modelo utilizamos o valor $\sigma = 15$.

Os resultados da inversão FWI e da FWI multiescala são ilustrados na Figura 4.4. Como esperado, com um modelo inicial relativamente bom, o FWI produz um resultado razoável (ver Figura 4.4c). No entanto, a estratégia de aprendizagem multiescala (ver Figura 4.4d) consegue entregar um resultado melhor, especialmente no que tange as estruturas profundas.

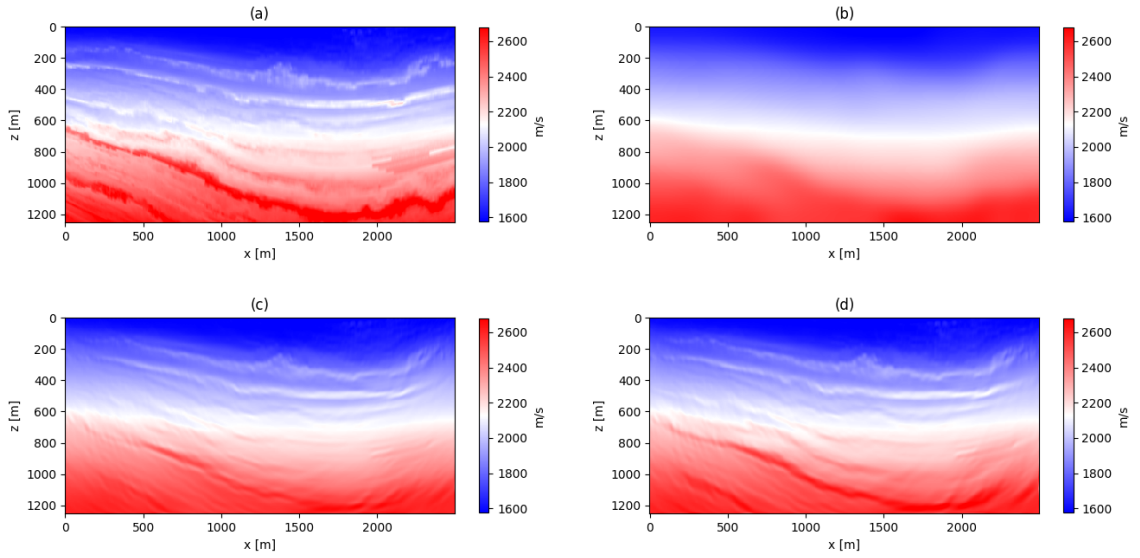


Figura 4.4: Comparação dos resultados da inversão do modelo SEAM Fase I Sedimentar. (a) Modelo Verdadeiro, (b) Modelo inicial, (c) FWI e (d) FWI multiescala.

Para corroborar o resultado obtido com a FWI multiescala em relação a FWI convencional, foi selecionado um perfil de velocidade obtido na posição central de cada modelo de velocidades envolvido no processo, de modo a permitir uma análise mais detalhada dos valores de velocidade obtidos. A Figura 4.5 mostra esses perfis de velocidades em profundidade para os modelos verdadeiro, inicial e invertido para cada estratégia. É visível como a FWI

multiescala conseguiu propiciar um melhor ajuste ao modelo verdadeiro do que a FWI. Essa observação pode ser melhor visualizada em detalhe como mostra a Figura 4.6.

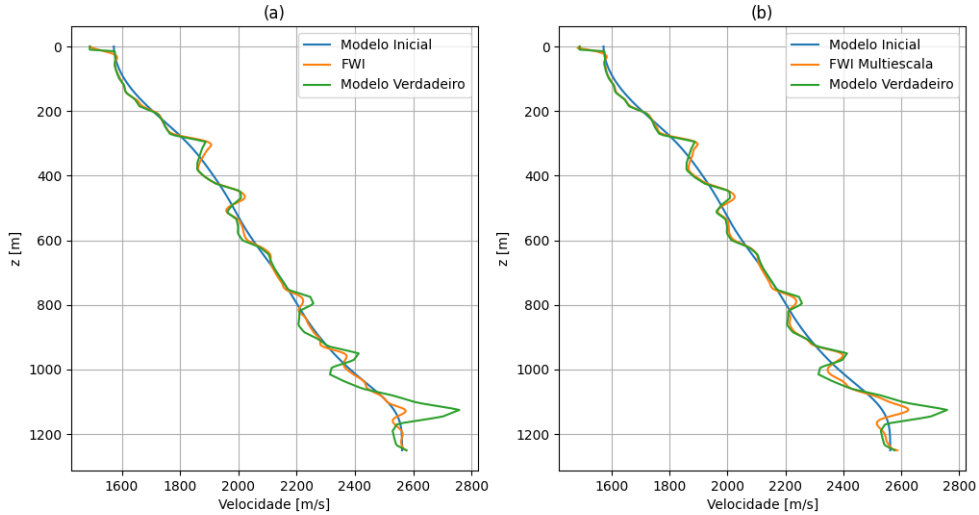


Figura 4.5: Perfis de velocidade. (a) Velocidades envolvidas na FWI e (b) Velocidades envolvidas na FWI multiescala.

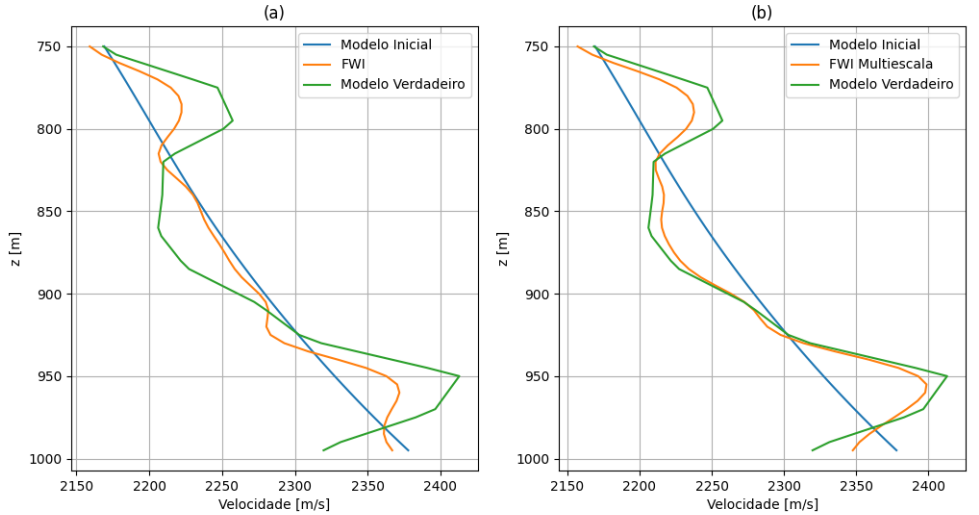


Figura 4.6: Perfis de velocidades em detalhe.

Os resultados das inversões nas etapas intermediárias da FWI multiescala até antes da ultima faixa de frequências podem ser visualizados na Figura 4.7, bem como os respectivos gráficos de erro da função objetivo por época na Figura 4.8.

Podemos avaliar qualitativamente o resultado da proposta multiescala, através dos sismogramas gerados com o modelo de velocidades produto dessa abordagem, em comparação

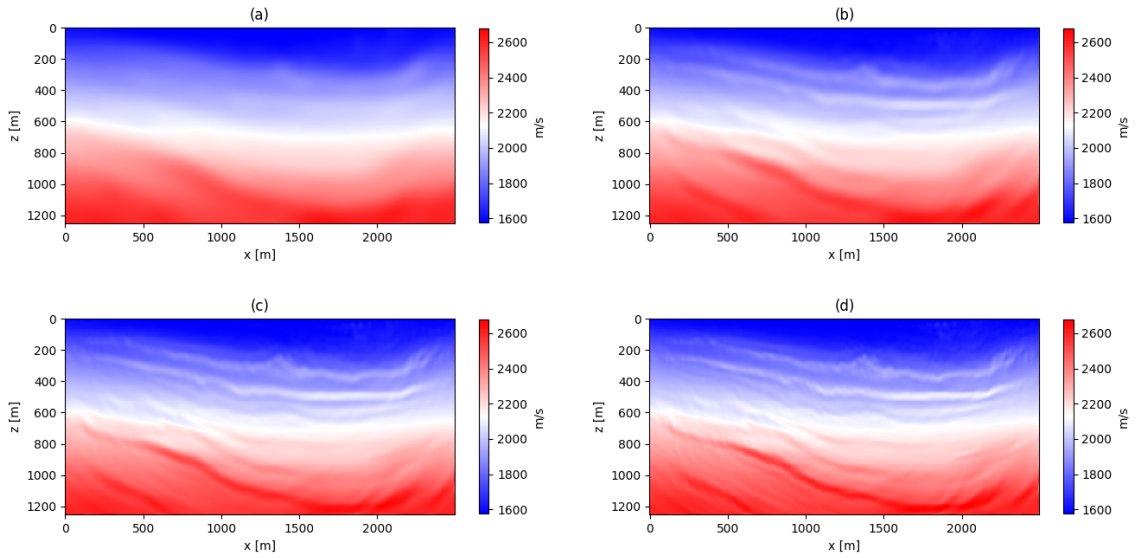
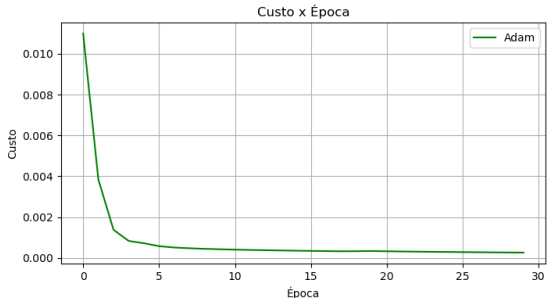


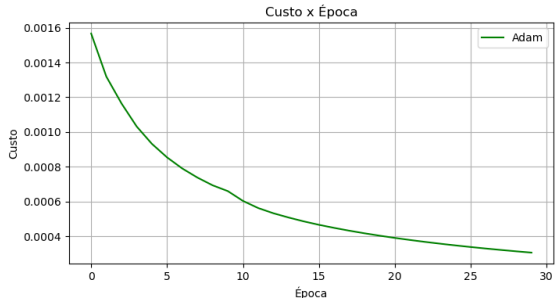
Figura 4.7: Resultados FWI multiescala modelo SEAM Fase I Sedimentar. (a) $f_{peak} = 5\,Hz$, (b) $f_{peak} = 10\,Hz$, (c) $f_{peak} = 15\,Hz$ e (d) $f_{peak} = 20\,Hz$.

aos sismogramas computados com o modelo verdadeiro, conforme pode ser visualizado nas Figuras 4.9 e 4.10.

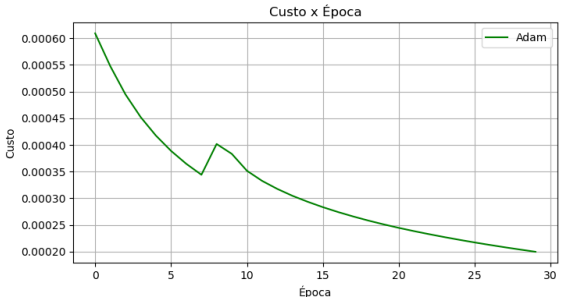
Por fim, a Figura 4.11 exibe o resultado da FWI multiescala ao utilizarmos otimizadores distintos no processo, como o Momento e o Adam, já discutidos anteriormente. Os modelos foram obtidos com os mesmo parâmetros, exceto a taxa de aprendizagem que foi de 10 para o Adam e 10^6 para o Momento. É vísivel a boa qualidade da resposta entregue pelo Adam, enquanto o Momento só consegue atualizar as porções mais rasas do modelo.



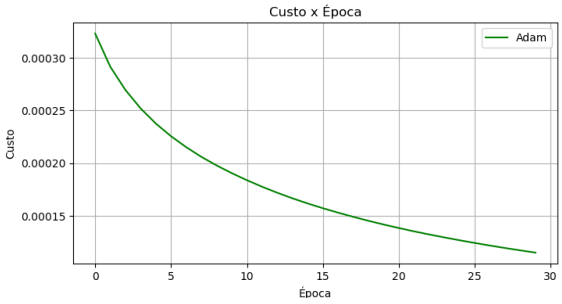
(a) $f_{peak} = 5\,Hz$



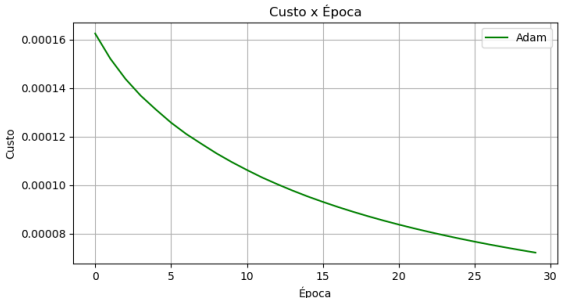
(b) $f_{peak} = 10\,Hz$



(c) $f_{peak} = 15\,Hz$



(d) $f_{peak} = 20\,Hz$



(e) $f_{peak} = 25\,Hz$

Figura 4.8: Gráficos de erros fornecidos pela função custo por época para cada escala da FWI multiescala do modelo SEAM Fase I sedimentar.

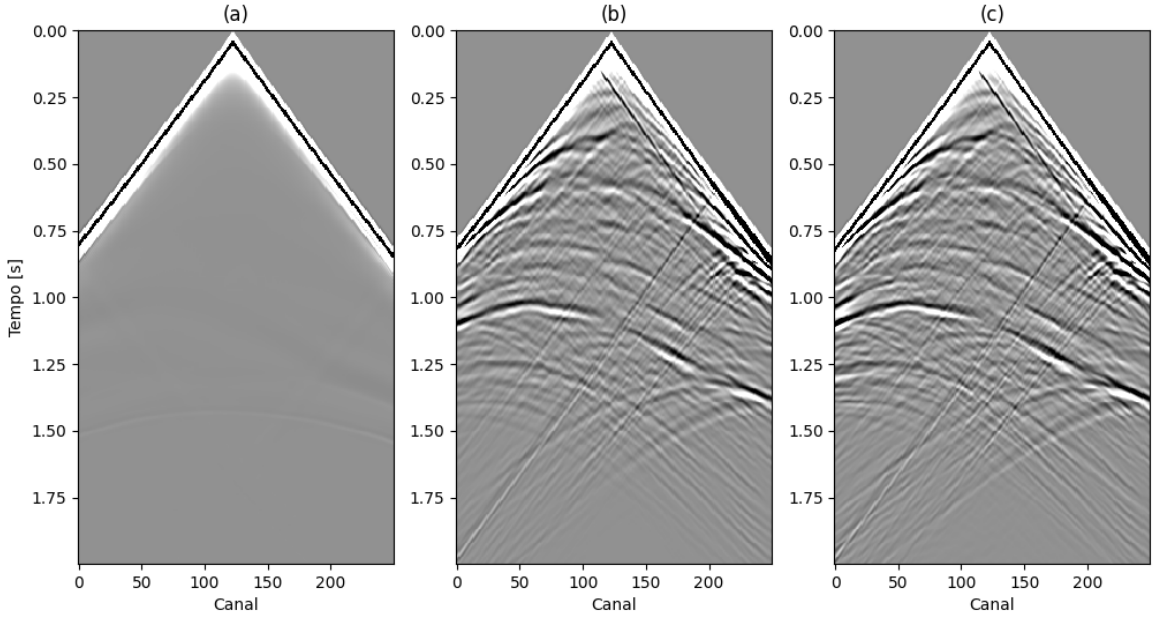


Figura 4.9: Tiro na posição central do modelo. (a) Obtido a partir do modelo de velocidades inicial, (b) a partir do modelo FWI multiescala e (c) modelo verdadeiro.

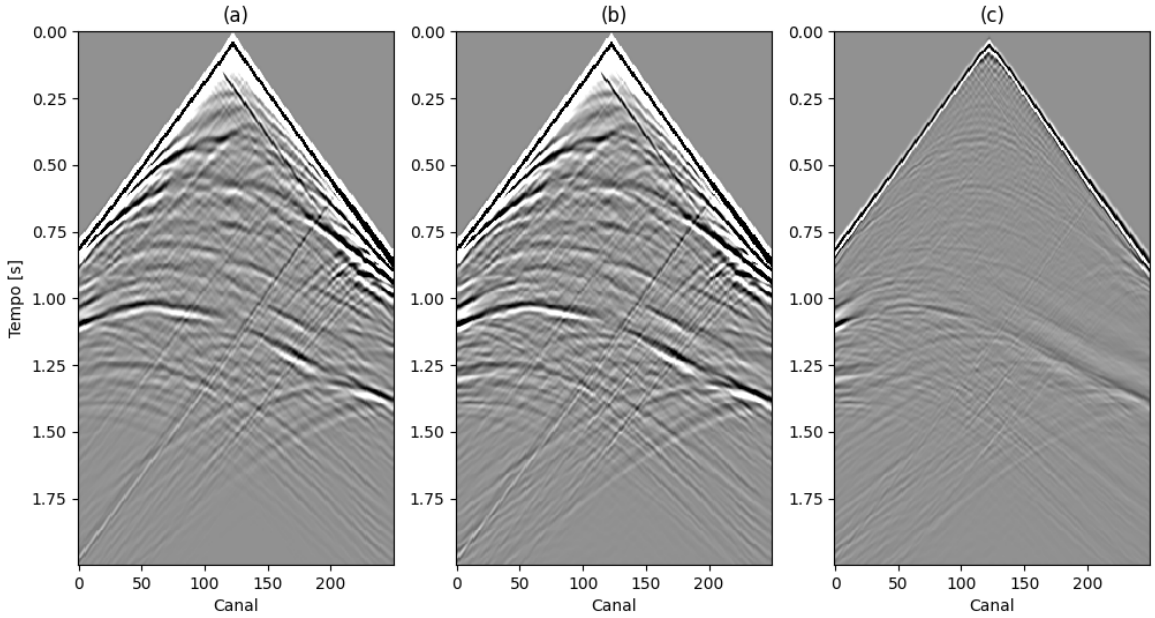
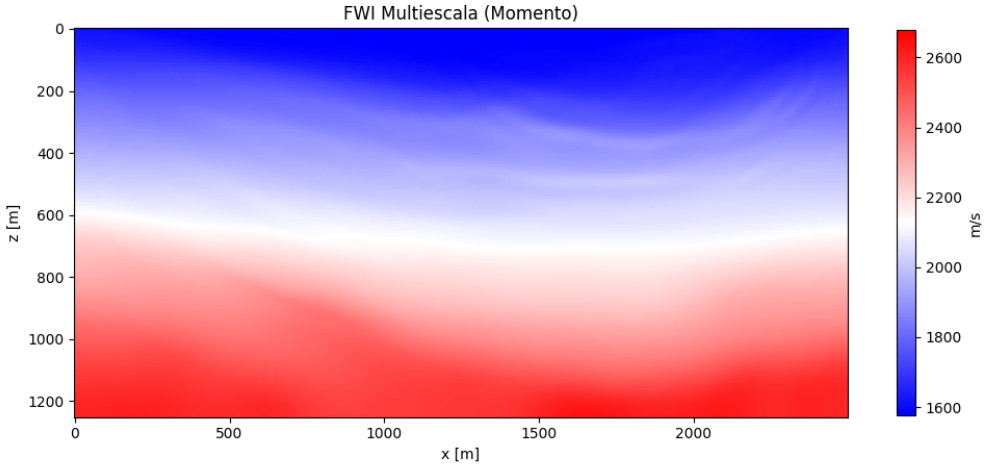
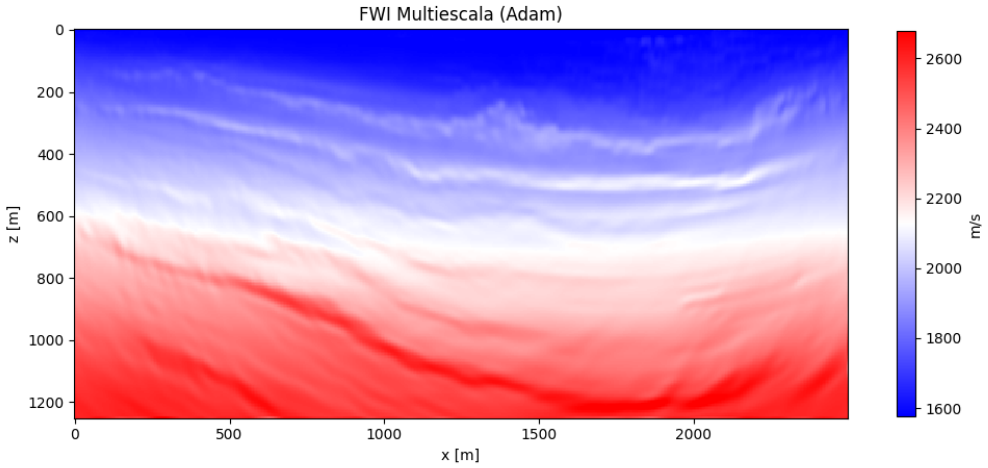


Figura 4.10: Tiro na posição central do modelo. (a) Obtido a partir do modelo FWI multiescala, (b) modelo verdadeiro e (d) diferença $((b) - (a))$.



(a) Momento



(b) Adam

Figura 4.11: Resultado da FWI multiescala para otimizadores distintos.

4.2.2 Resultados do Modelo SEAM Phase I

Neste exemplo, de maior complexidade geológica, a FWI foi obtida para uma frequência dominante de $12\,Hz$, passadas 250 épocas com o otimizador Adam e taxa de aprendizagem (η) 10. Quanto à FWI multiescala, foram escolhidas arbitrariamente as frequências dominantes de $3\,Hz$, $6\,Hz$, $9\,Hz$ e $12\,Hz$. Para cada banda de frequências foram feitas 250 épocas utilizando o método de otimização Adam e taxa de aprendizagem 10. Aqui também o modelo inicial foi obtido pela suavização do modelo verdadeiro, via convolução com filtro gaussiano, com $\sigma = 15$.

Os resultados da inversão FWI e da FWI multiescala são ilustrados na Figura 4.12. Como pode ser observado, apesar de um modelo inicial relativamente bom, o FWI produz um resultado instável na porção rasa do modelo (ver Figura 4.12c). No entanto, a estratégia de aprendizagem multiescala (ver Figura 4.12d) consegue novamente entregar um resultado melhor.

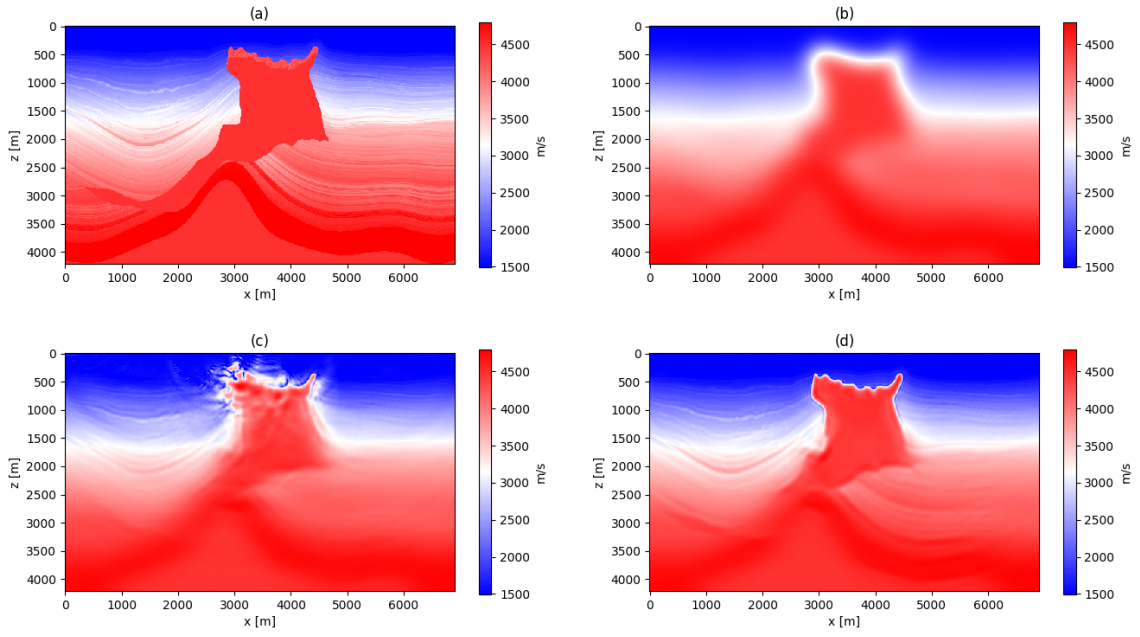


Figura 4.12: Comparação dos resultados da inversão do modelo SEAM Fase I. (a) Modelo Verdadeiro, (b) Modelo inicial, (c) FWI e (d) FWI multiescala.

Como controle de qualidade da inversão, foi selecionado um perfil de velocidade obtido na posição $2880\,m$ de cada modelo de velocidades envolvido no processo, de modo a permitir uma análise mais contundente sobre os valores de velocidade obtidos. As Figuras 4.13 e 4.14 exibem os resultados desta extração.

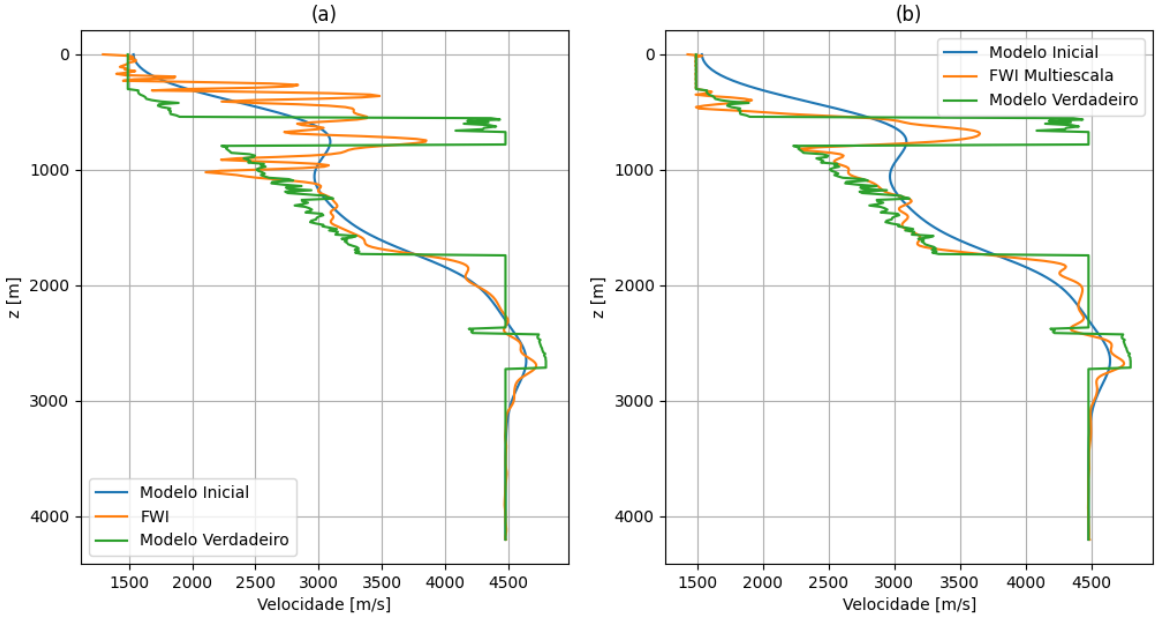


Figura 4.13: Perfis de velocidade na posição 2880 m. (a) Velocidades envolvidas na FWI e (b) Velocidades envolvidas na FWI multiescala.

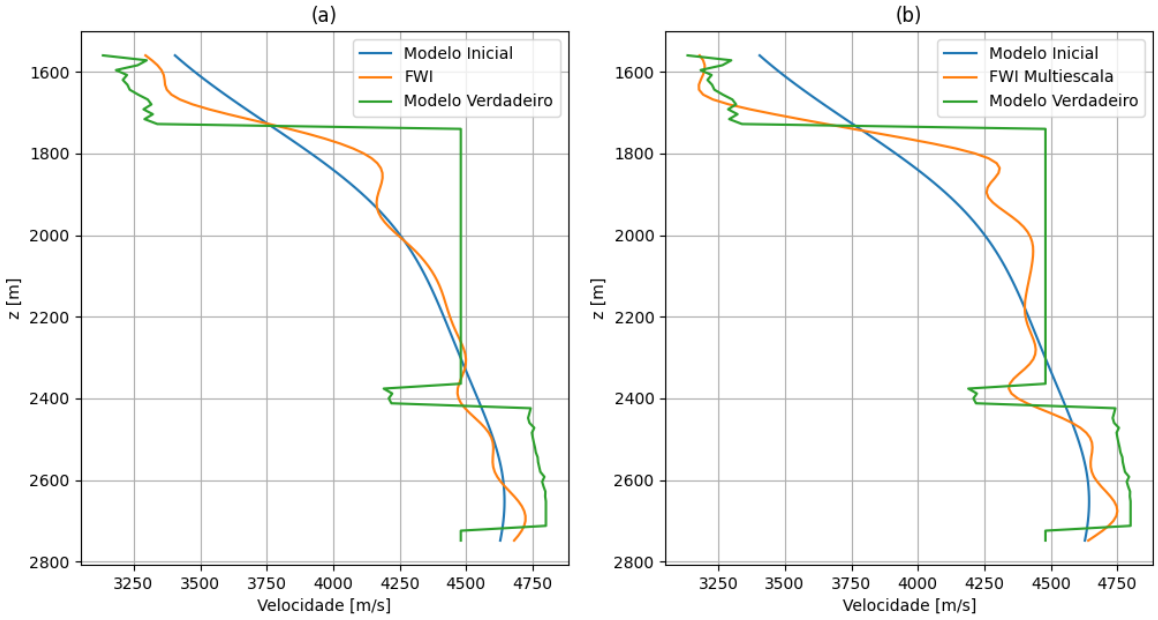


Figura 4.14: Perfis de velocidade na posição 2880 m em detalhe.

Assim como exibida no exemplo anterior, os resultados das inversões nas etapas intermediárias da FWI multiescala até a última faixa de frequências podem ser visualizados na Figura 4.15, bem como os respectivos gráficos de erro da função objetivo por época (Figura 4.16).

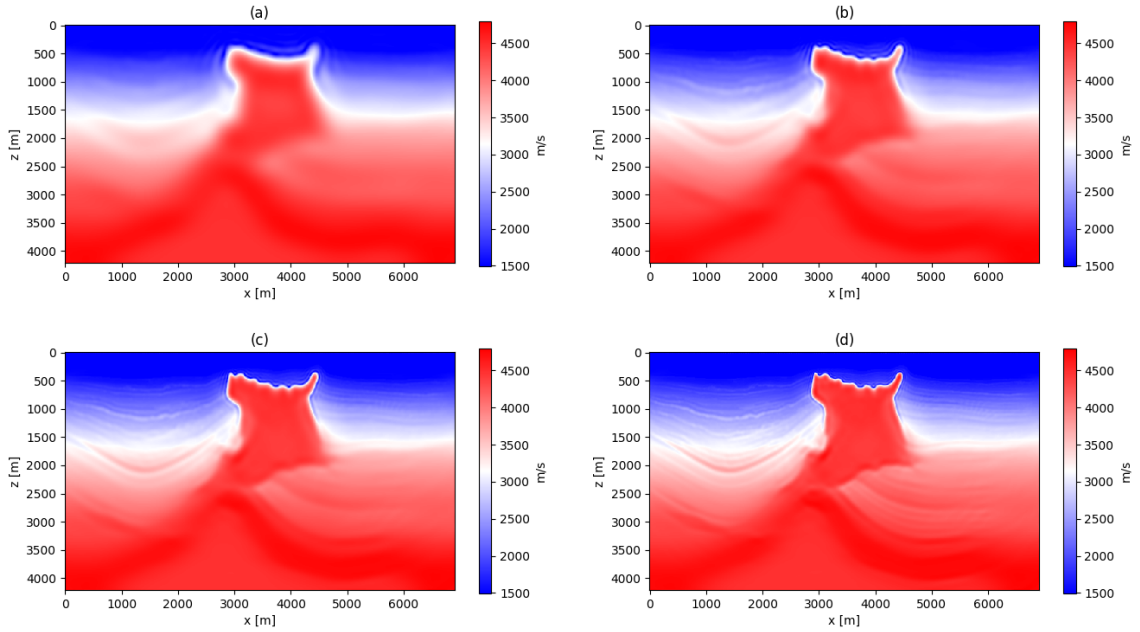
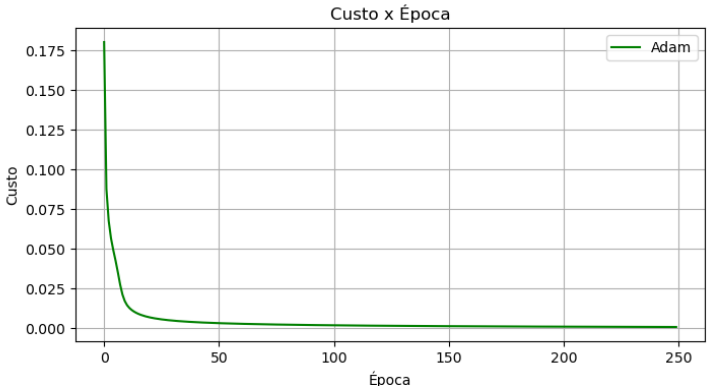


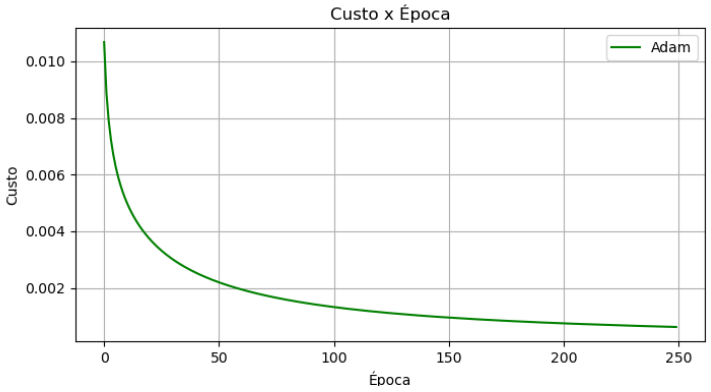
Figura 4.15: Resultados FWI multiescala modelo SEAM Fase I. (a) $f_{peak} = 3\,Hz$, (b) $f_{peak} = 6\,Hz$, (c) $f_{peak} = 9\,Hz$ e (d) $f_{peak} = 12\,Hz$.

De maneira qualitativa, o resultado da proposta multiescala comparado ao resultado fornecido pelo modelo verdadeiro pode ser avaliado nas Figuras 4.17 e 4.18, que mostram os sismogramas gerados a partir dos respectivos modelos de velocidades.

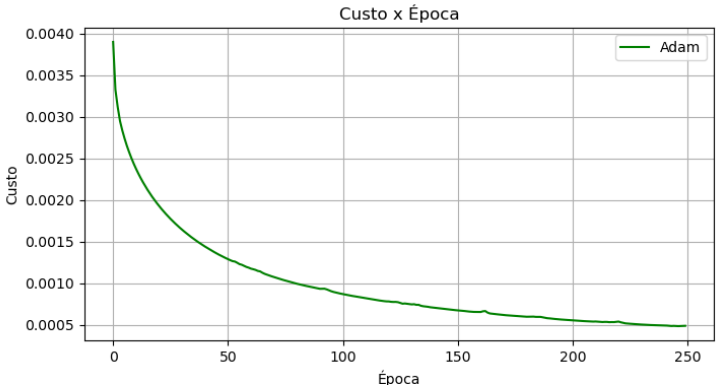
Novamente são comparados os resultados da inversões com o Adam e o Momento, obtidos com os mesmos parâmetros, exceto a taxa de aprendizagem que foi de 10 para o Adam e 10^6 para o Momento. Mais uma vez, a qualidade da resposta entregue pelo Adam é superior ao obtido com o otimizador Momento que também só concentra atualizações nas porções mais rasas do modelo.



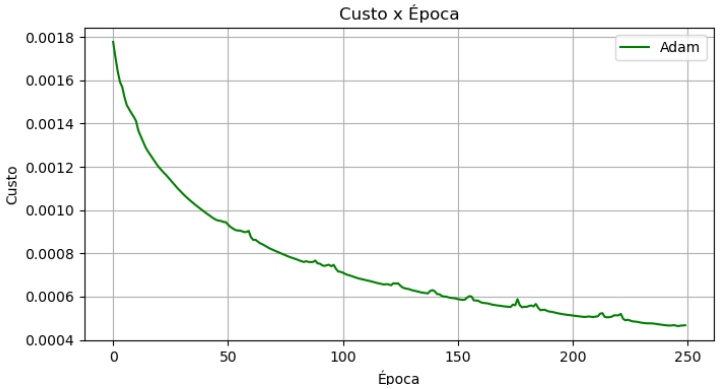
(a) $f_{peak} = 3\,Hz$



(b) $f_{peak} = 6\,Hz$



(c) $f_{peak} = 9\,Hz$



(d) $f_{peak} = 12\,Hz$

Figura 4.16: Gráficos de erros fornecidos pela função custo por época para cada escala da FWI Multiescala do modelo SEAM Fase I.

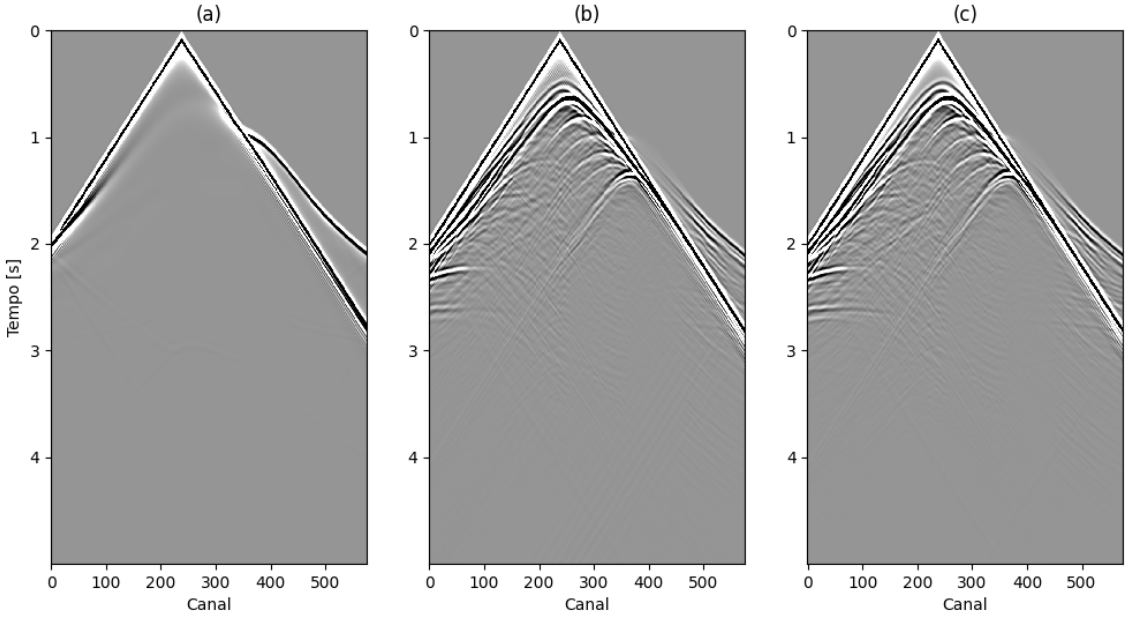


Figura 4.17: Tiro na posição posição 2880 m do modelo. (a) Obtido a partir do modelo de velocidades inicial, (b) a partir do modelo FWI multiescala e (c) modelo verdadeiro.

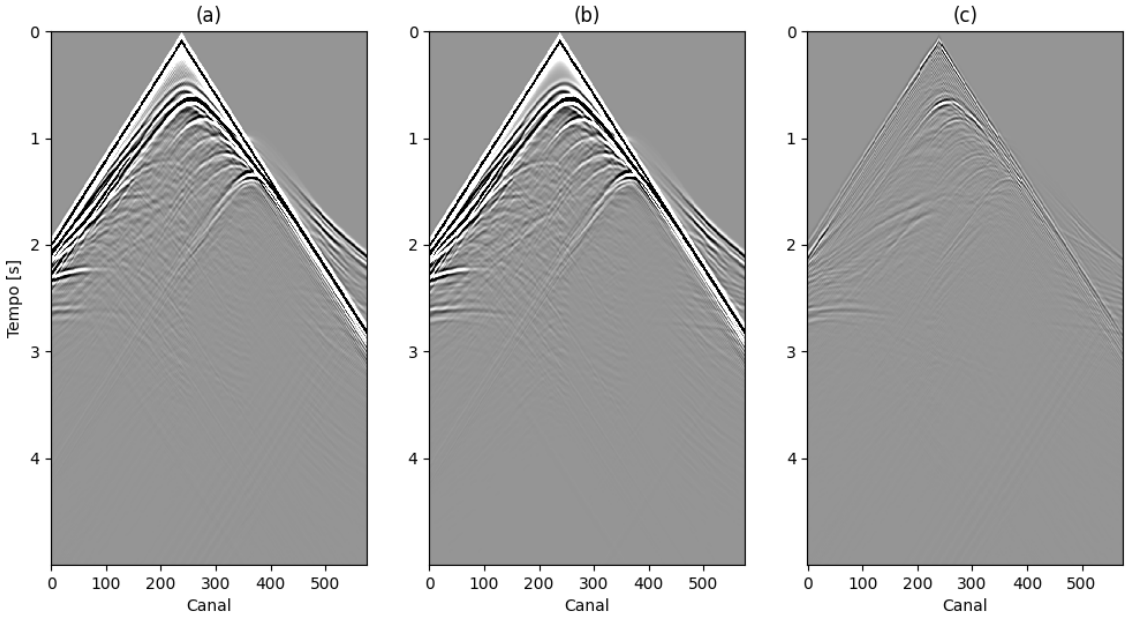
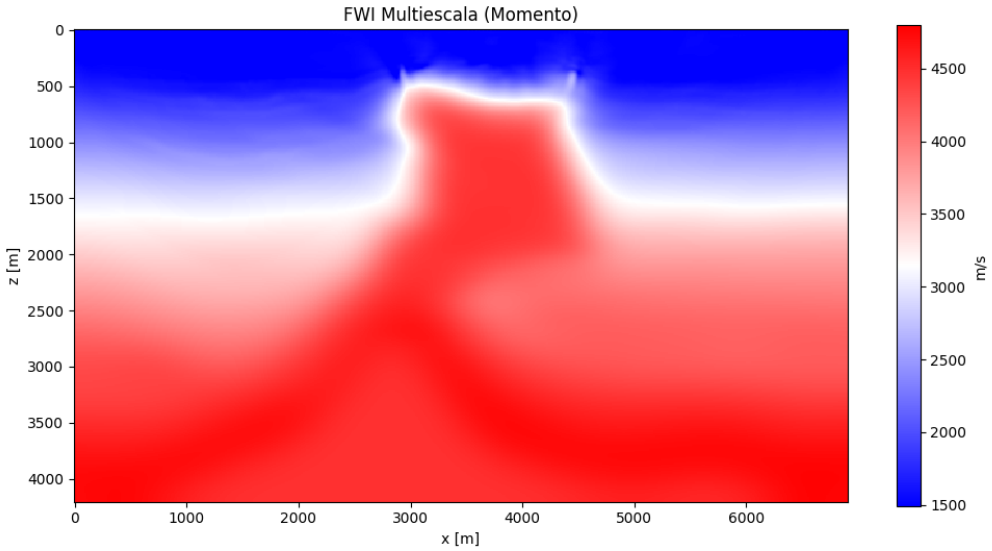
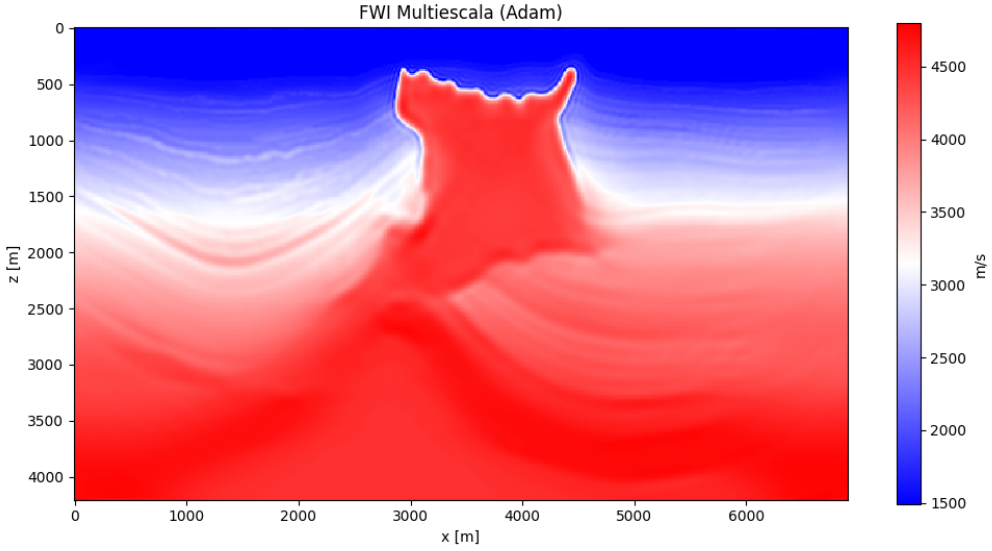


Figura 4.18: Tiro na posição posição 2880 m do modelo. (a) Obtido a partir do modelo FWI multiescala, (b) modelo verdadeiro e (d) diferença ($(b) - (a)$).



(a) Momento



(b) Adam

Figura 4.19: Resultado da FWI multiescala para otimizadores distintos.

4.2.3 Resultados do Modelo Marmousi

Nosso último teste foi realizado com o modelo Marmousi, no qual para o FWI utilizamos uma frequência dominante de $12\,Hz$, iteradas 500 épocas com o otimizador Adam e com alteração da taxa de aprendizagem (η) para 5. Quanto a FWI multiescala, foram escolhidas arbitrariamente as frequências dominantes de $3\,Hz$, $6\,Hz$, $9\,Hz$ e $12\,Hz$. Para cada banda de frequências foram executadas 500 épocas utilizando o método de otimização Adam e taxa

de aprendizagem 5. O modelo inicial foi obtido pela suavização do modelo verdadeiro, via convolução com filtro gaussiano, com $\sigma = 15$.

Os resultados da inversão FWI e da FWI multiescala são ilustrados na Figura 4.20. Como pode ser observado, apesar de um modelo inicial relativamente bom, o FWI apresenta um resultado subestimado (ver Figura 4.20c). No entanto, a estratégia de aprendizagem multiescala (ver Figura 4.20d) consegue novamente entregar um resultado como uma qualidade superior e no geral mais próxima do modelo verdadeiro.

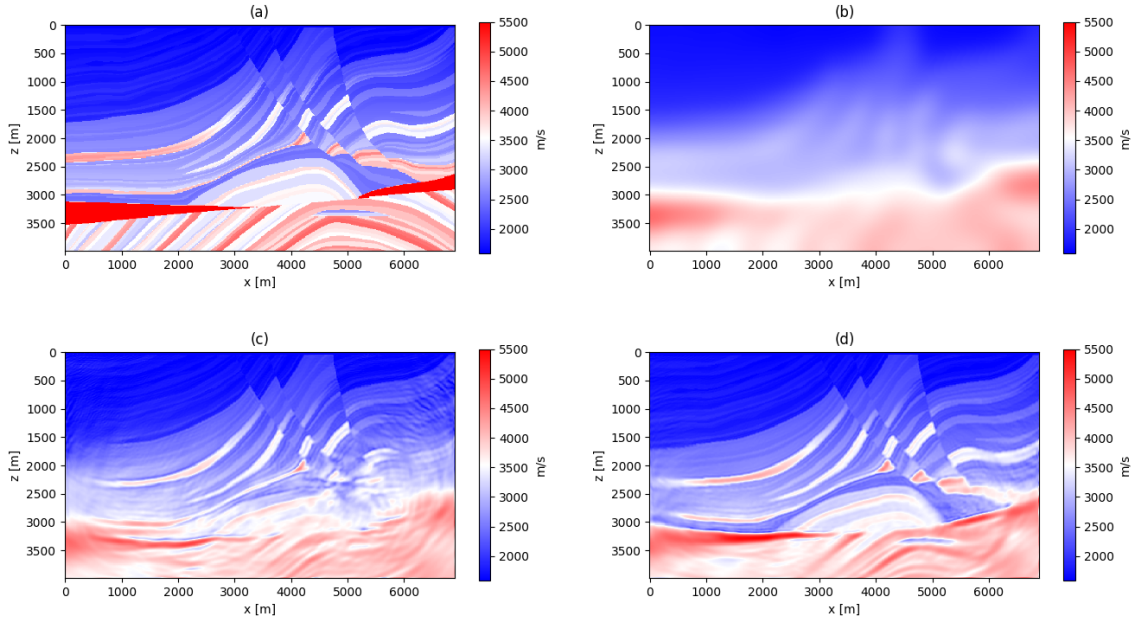


Figura 4.20: Comparação dos resultados da inversão do modelo Marmousi. (a) Modelo Verdadeiro, (b) Modelo inicial, (c) FWI e (d) FWI Multiescala.

Como controle de qualidade da inversão, foi selecionado um perfil de velocidade obtido na posição central de cada modelo de velocidades envolvido no processo, de modo a permitir uma análise mais criteriosa sobre os valores de velocidade obtidos. As Figuras 4.21 e 4.22 exibem os resultados dessa extração.

Do mesmo modos que nos exemplos anteriores, os resultados das inversões nas etapas intermediárias da FWI multiescala podem ser visualizados na Figura 4.23, bem como os respectivos gráficos de erro da função objetivo por época na Figura 4.24.

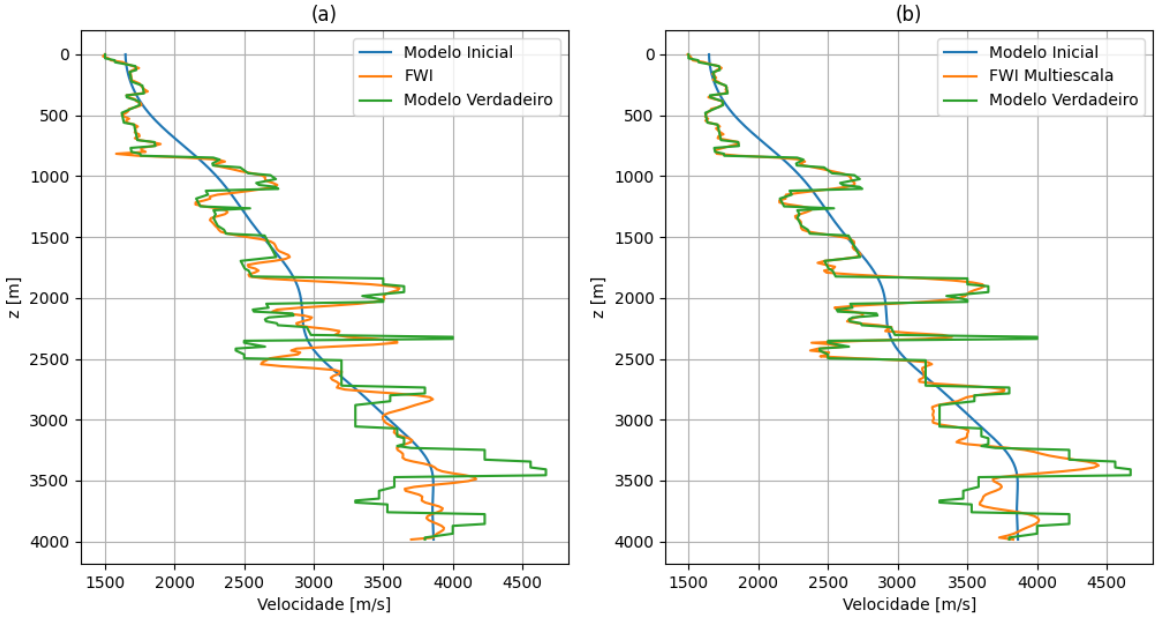


Figura 4.21: Perfis de velocidade na posição central do modelo. (a) Velocidades envolvidas na FWI e (b) Velocidades envolvidas na FWI multiescala.

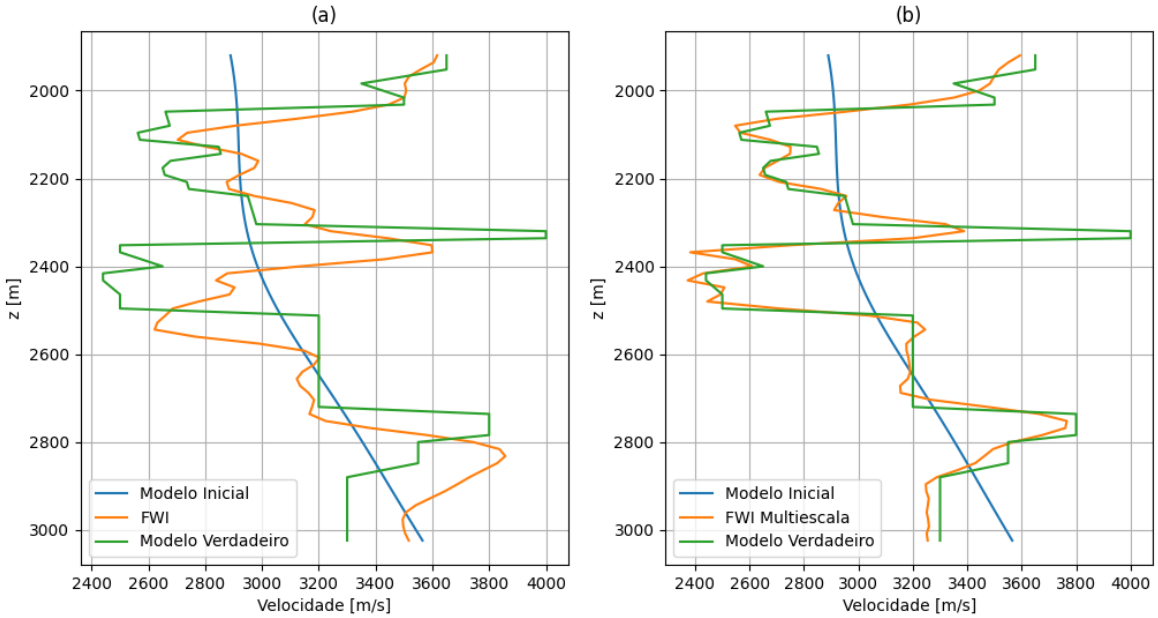


Figura 4.22: Perfis de velocidade na posição na posição central do modelo em detalhe.

Qualitativamente o resultado da proposta multiescala comparado ao resultado fornecido pelo modelo verdadeiro pode ser avaliado através dos computos dos sismogramas, com as respectivas velocidades, que são mostrados nas Figuras 4.25 e 4.26. Os resultados reforçam a qualidade do modelo de velocidades obtido através da abordagem proposta nesse trabalho.

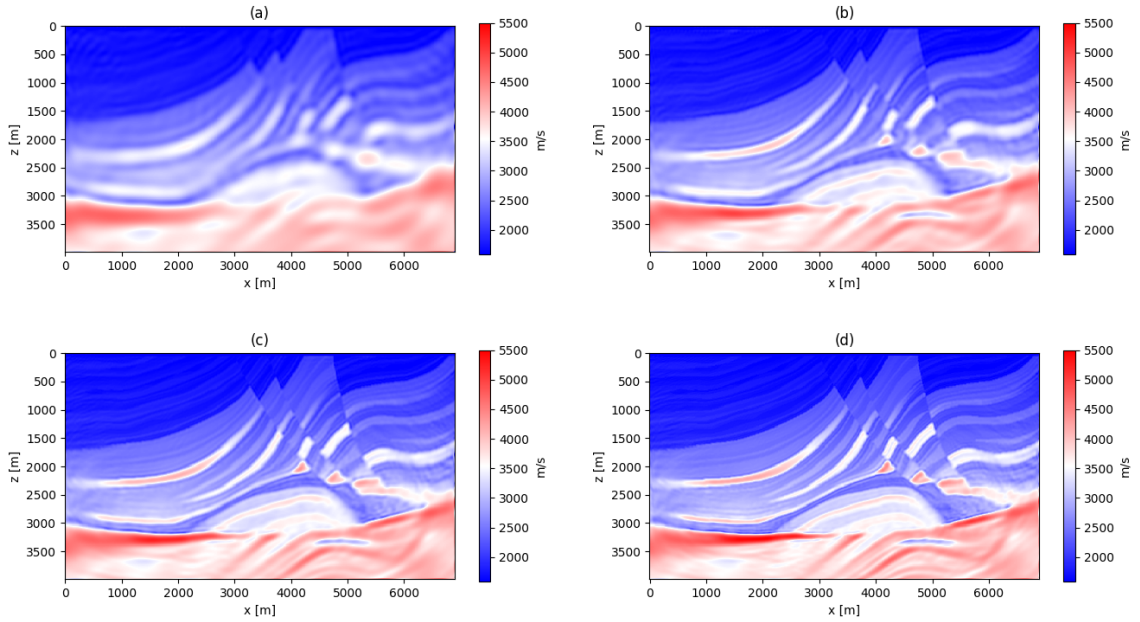
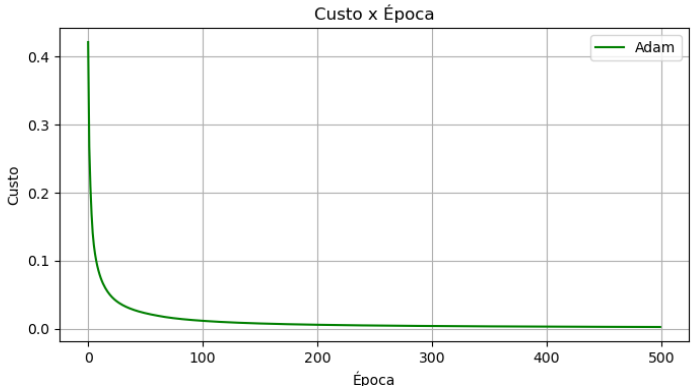
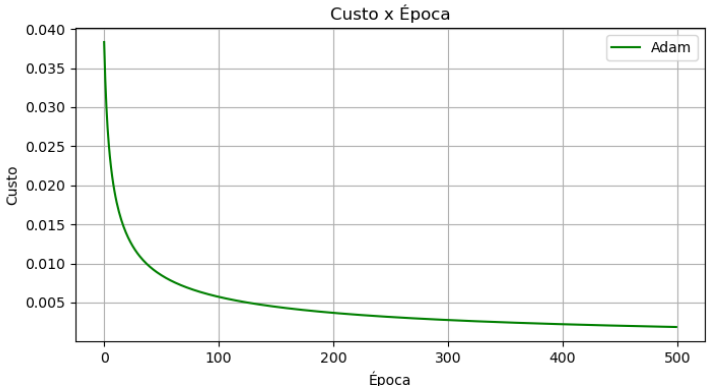


Figura 4.23: Resultados FWI multiescala modelo Marmousi. (a) $f_{peak} = 3\,Hz$, (b) $f_{peak} = 6\,Hz$, (c) $f_{peak} = 9\,Hz$ e (d) $f_{peak} = 12\,Hz$.

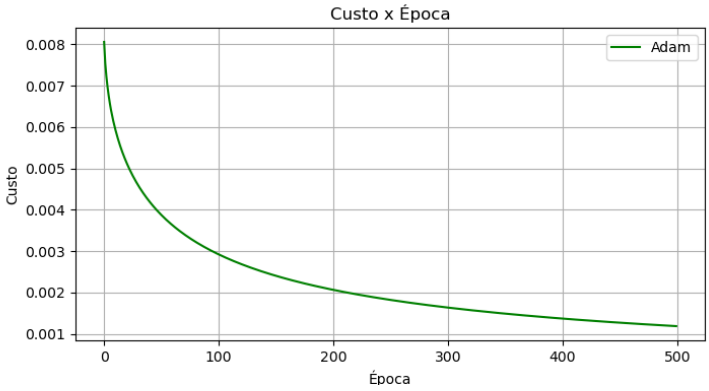
Na sequência, são comparados os resultados da inversões com o Adam e o Momento, obtidos com os mesmo parâmetros, exceto a taxa de aprendizagem que foi de 5 para o Adam e 10^6 para o Momento. Mais uma vez a qualidade da resposta entregue pelo Adam é superior ao obtido com o otimizador Momento que também só concentra atualizações nas porções mais rasas do modelo, confirmando a robustez do método Adam no processo de otimização.



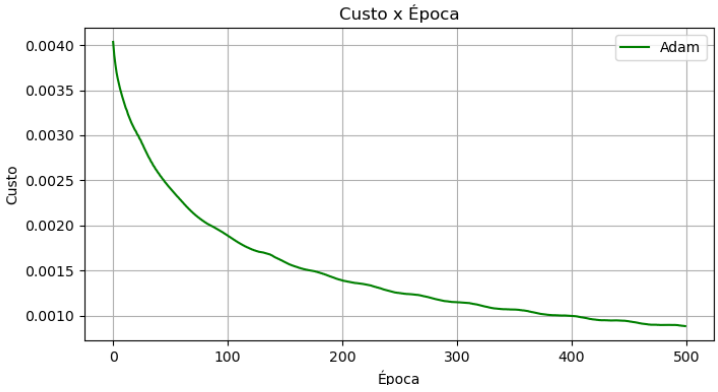
(a) $f_{peak} = 3\,Hz$



(b) $f_{peak} = 6\,Hz$



(c) $f_{peak} = 9\,Hz$



(d) $f_{peak} = 12Hz$

Figura 4.24: Gráficos de erros fornecidos pela função custo por época para cada escala da FWI Multiescala do modelo Marmousi.

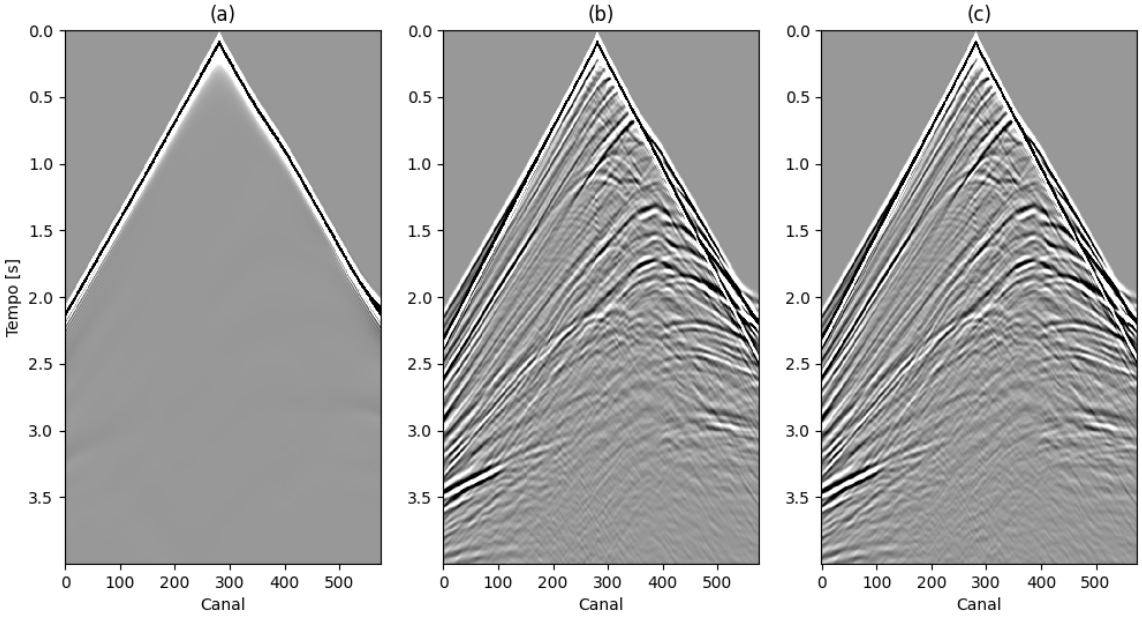


Figura 4.25: Tiro na posição central do modelo. (a) Obtido a partir do modelo de velocidades inicial, (b) a partir do modelo FWI multiescala e (c) modelo verdadeiro.

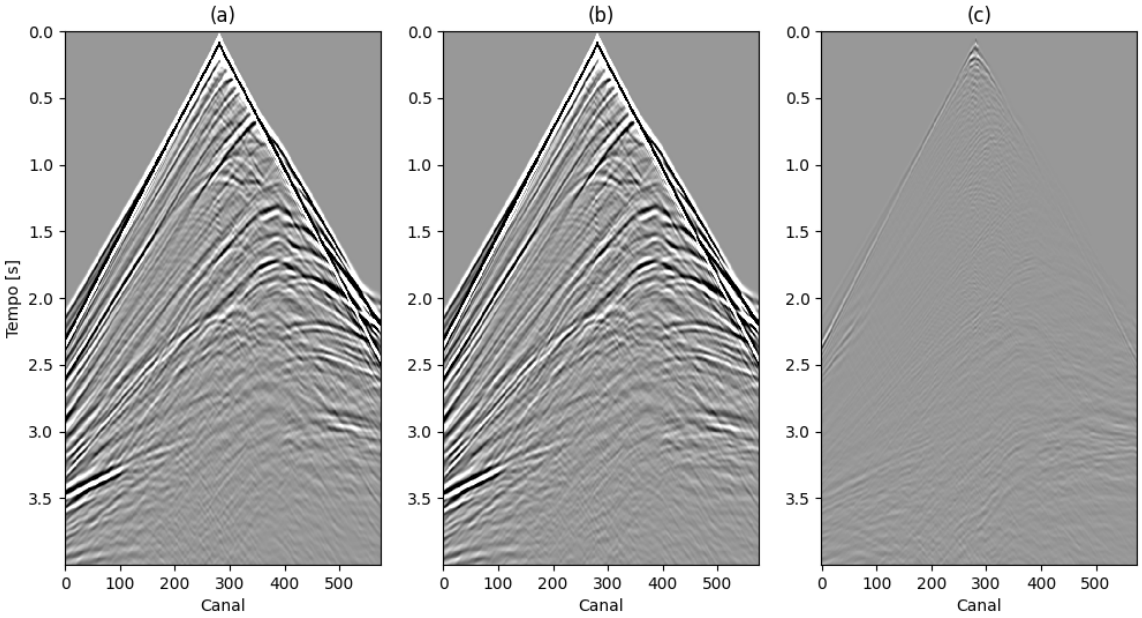
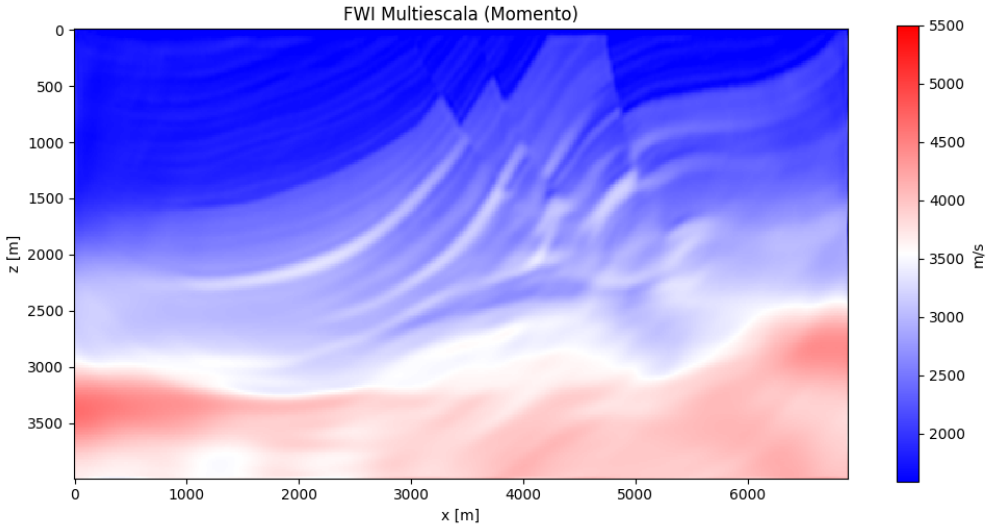
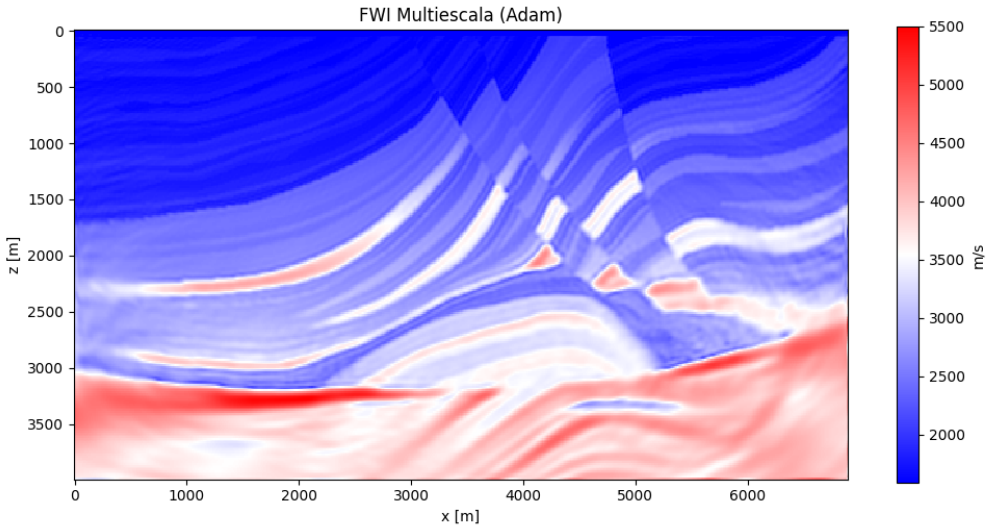


Figura 4.26: Tiro na posição central do modelo. (a) Obtido a partir do modelo FWI multiescala, (b) modelo verdadeiro e (d) diferença $((b) - (a))$.

Por fim, com o objetivo de tornar a observação mais realista, foi feita a adição de ruído branco gaussiano aos dados sísmicos observados, como mostra a Figura 4.28.



(a) Momento



(b) Adam

Figura 4.27: Resultado da FWI multiescala para otimizadores distintos.

De acordo com os resultados mostrados na Figura 4.29, pode-se observar a capacidade da FWI multiescala de inverter os modelos mesmo quando os dados sísmicos estão contaminados com ruído, no entanto o resultado da inversão da FWI continua a subestimar o modelo de velocidades.

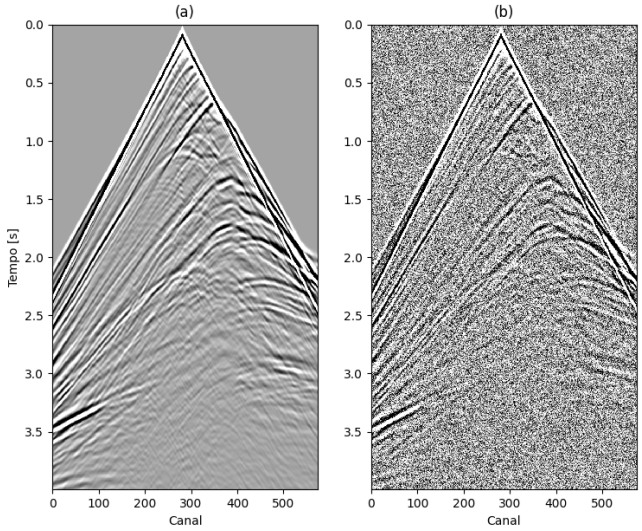


Figura 4.28: Dado observado gerado com modelo verdadeiro do Marmousi. (a) Dado sísmico livre de ruído (b) Dado sísmico ruidoso (SNR = 10 db).

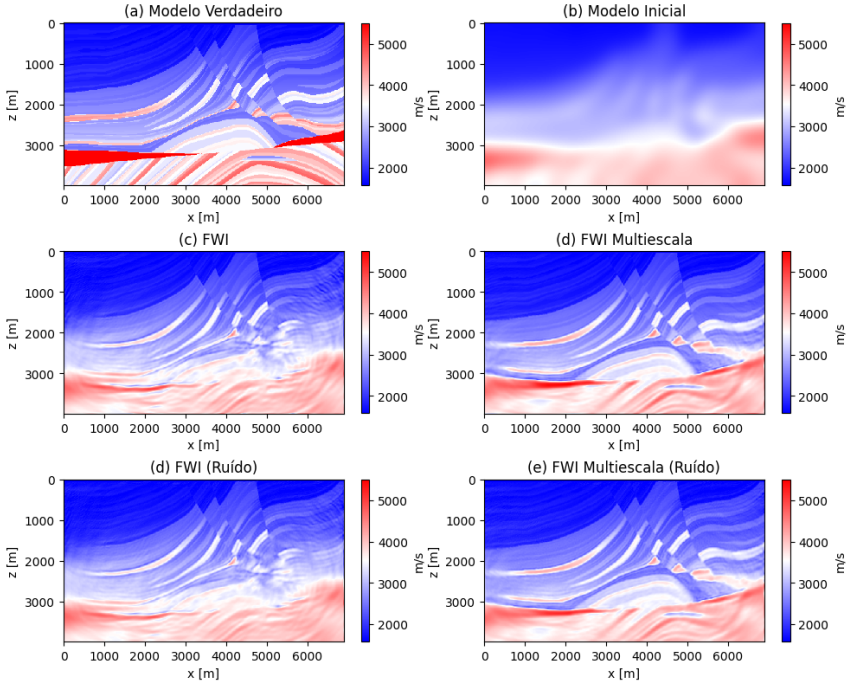


Figura 4.29: Comparação dos resultados da inversão do modelo Marmousi. (a) Modelo Verdadeiro, (b) Modelo inicial, (c) e (d) Resultados da inversão por FWI e FWI multiescala com dados sísmicos observados sem ruído, respectivamente (e) e (f) Resultados da inversão por FWI e FWI multiescala com dados sísmicos observados com ruído, respectivamente.

4.2.4 Avaliação Quantitativa

Para estimar a qualidade da abordagem multiescala na frequência, avaliamos quantitativamente a qualidade da velocidade invertida em relação ao modelo verdadeiro. Nesta avaliação, utilizamos como métrica o erro relativo entre o modelo de velocidade invertido ($\hat{v}$) e a velocidade verdadeira (v) definido como:

$$Erro\,(\hat{v}, v) = \frac{\|v - \hat{v}\|_2}{\|v\|_2} \times 100 \tag{4.1}$$

onde $\|\cdot\|_2$ é a norma l_2. Quanto menor o erro melhor a reconstrução (ver Tabela 4.4).

Método	SEAM Fase I Sedimentar	SEAM Fase I	Marmousi
FWI	0,88%	2,92%	8,68%
FWI Multiescala	**0,66**%	**1,11**%	**5,41**%

Tabela 4.4: Erro relativo entre os modelos de velocidades usando FWI e FWI Multiescala com dados sísmicos sem ruído.

Da mesma maneira, para o único teste com dados observados com ruído aditivo, o erro relativo calculado entre os modelos em cada processo de inversão é mostrado na Tabela 4.5.

Método	Marmousi
FWI	8,57%
FWI Multiescala	**5,94**%

Tabela 4.5: Erro relativo entre os modelos de velocidades usando FWI e FWI Multiescala com dados sísmicos com ruído.

4.2.5 Sensibilidade ao Modelo Inicial

A medida que o grau de complexidade geológica nos modelos testados foi elevada, pela inclusão, por exemplo, de estruturas como corpos de sal complexos, trouxe consigo dificuldades adicionais para a FWI convencional, devido aos já mencionados salto de ciclo e outros fatores como forte contrastes de amplitudes. Outro ponto crucial é a qualidade do modelo inicial, aqui fizemos a comparação da abordagem multiescala na frequência com a FWI convencional aumentando o grau de suavização deste modelo de partida.

Resultados da Inversão com Modelo Suavizado

A Figura 4.30 mostra o desempenho de inversão do modelo Marmousi aplicando FWI e FWI multiescala. O modelo inicial foi construído via suavização pela função gaussiana variando o

desvio padrão. Um valor mais alto, implica numa suavização mais forte e consequentemente mais longe do modelo verdadeiro. Notadamente, a FWI subestima dramaticamente a distribuição de velocidade e não consegue inverter de forma precisa as estruturas geológicas para as três situações. À medida que o grau de suavização aumenta, o resultado apresenta artefatos espúrios de alta frequência, que é um problema típico de mínimos locais. Ao contrário, a FWI multiescala obtém reconstruções de alta qualidade, estando mais próximos do modelo verdadeiro. Recupera satisfatoriamente a maioria das esturuturas do modelo Marmousi com valores de erro considerados baixos.

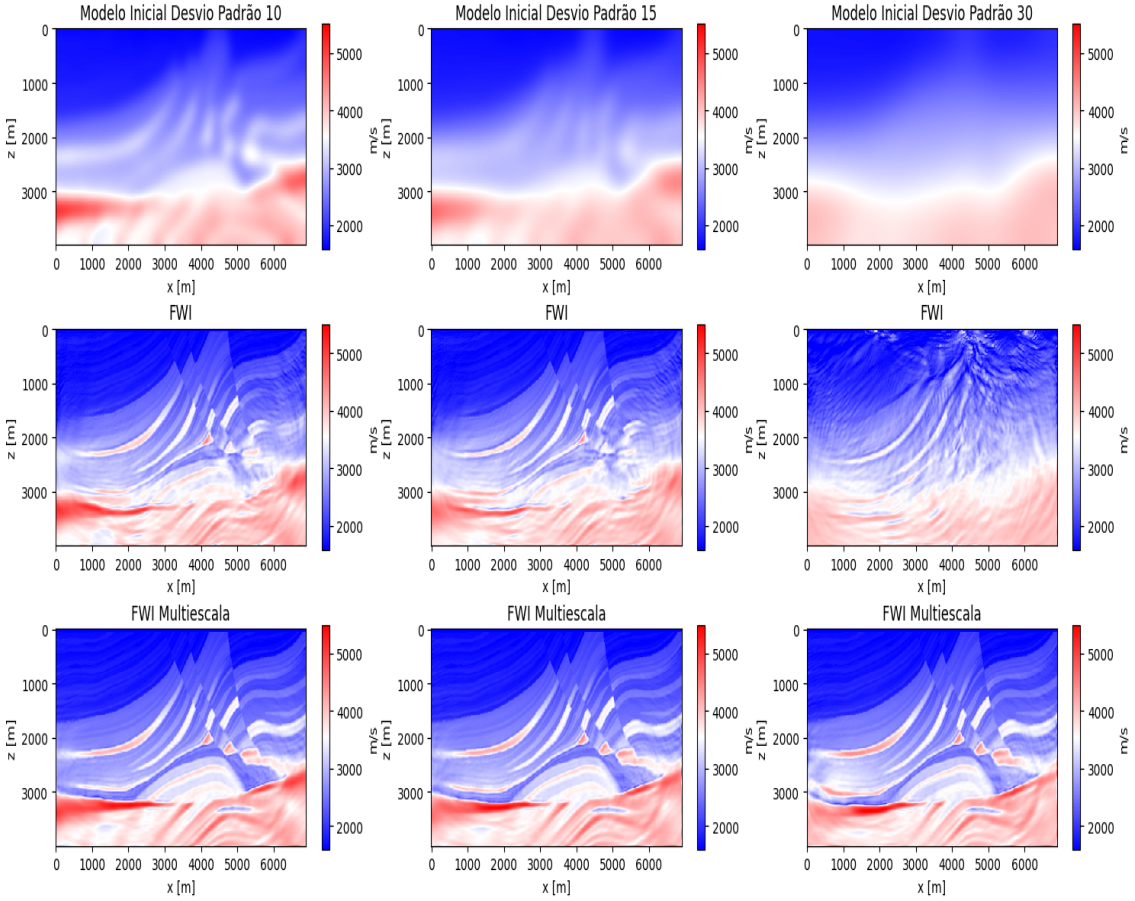


Figura 4.30: Resultados da inversão do modelo Marmousi com FWI (meio) e FWI multiescala (base), a partir de diferentes suavizações do modelo inicial (topo).

5

Conclusões

Neste trabalho testou-se, em modelos sísmicos bidimensionais, uma eficiente metodologia de aprendizado não supervisionado, que aproveita as vantagens de equações diferenciais parciais e técnicas de aprendizado profundo para inversão da forma de onda completa. Além disso foi proposta e implementada a abordagem multiescala na frequência como um elemento adicional à técnica convencional, que impactou positivamente a qualidade final dos resultados obtidos.

A integração da equação acústica da onda como uma rede neural profunda a FWI, simplificou o processo de otimização de modo a não requerer dados extras de treinamento, além de se mostrar flexível devido à diferenciação automática.

Nos experimentos realizados, a abordagem foi validada em modelos sintéticos, conhecidos e disponíveis abertamente, como Marmousi e Seam, que possuem características geológicas distintas e condições desafiadoras, como a qualidade do modelo inicial forncecido. Os experimentos numéricos demonstraram que o método proposto com a abordagem multiescala supera o algoritmo FWI convencional em todas as configurações. Além disso, a escolha pelo otimizador Adam, entregou uma convergência melhor e mais rápida, no sentido em que recupera informações detalhadas de velocidade em zonas rasas e profundas, em direção ao modelo de velocidades verdadeiro do que o otimizador Momento. Ademais, os resultados também confirmaram que a abordagem proposta é de fato uma solução para contornar os problemas de mínimos locais, de situções com dados de observação ruidosos e de modelos iniciais menos precisos, ao demonstrar insensibilidade ao modelo de partida.

Um fator relevante, é que o campo de ondas da modelagem direta precisa ser salvo na memória para uso durante a retropropagação. Isso implica que levantamentos 2D realistas, e provavelmete 3D, exigirão mais memória do que a disponível. Essa limitação computacional

dificulta a aplicação do método proposto para modelos práticos, mas abre caminhos para pesquisas nessa direção.

Agradecimentos

Gostaria de agradecer primeiramente à minha família, em especial a minha esposa Jusciane Silva, pela motivação, suporte e carinho nesse desafio e as minhas crianças, Joana e Júlia, por alegrarem meus dias e se tornarem minhas fontes de inspiração e busca por dias melhores.

Agradeço especialmente ao professor Reynam Pestana, pela oportunidade de tê-lo como orientador e pelo conhecimento adquirido nessa trajetória, além de enaltecer o seu amplo conhecimento e a sua enorme capacidade de estar sempre trazendo e compartilhando com os alunos as inovações tecnológicas emergentes na geofísica. Agradeço também aos demais professores da pós-graduação em geofísica.

Aos colegas da pós-graduação, em especial a Nei Davi Costa Figueiredo, pela amizade, pelas constantes trocas de conhecimento e pelo suporte em situações que a vida nos apresenta.

Agradeço a UFBA, CPGG-UFBA e seus funcionários pela infraestrutura oferecida.

Agradeço à Petrobras, pela oportunidade de realizar este trabalho, em especial a Luis Henrique Amaral.

Enfim, agradeço a todos que estiveram envolvidos diretamente e indiretamente na construção desta dissertação. Muito obrigado!

Apêndice A

Cálculo do gradiente pelo método adjunto

A busca pela melhor estimativa dos parâmetros do modelo a partir de dados medidos geralmente consiste em minimizar um erro funcional como o mostrado na equação 1.19, ou seja, obter a medida do gradiente dada por

$$\frac{\partial J}{\partial \mathbf{m}} = 0. \tag{A.1}$$

Estimar iterativamente esse gradiente da função objetivo por meio da matriz de Fréchet, também dita matriz Jacobiana, requer grande esforço computacional. Como alternativa o método baseado no estado adjunto (*adjoint-state method*) foi introduzido na teoria dos problemas inversos por Chavent (1974), sendo usado para calcular esse gradiente, sem a necessidade do cálculo das tais derivadas de Frechét. Neste apêndice a derivação do método adjunto será baseada seguindo a derivação apresentada por Plessix (2006), através do Lagrangiano associado ao funcional.

A.1 Obtenção do gradiente pelo método adjunto

No método adjunto a tarefa é encontrar o gradiente de um funcional $J(m)$, que depende da variável de estado $u(m)$. J é então definido como um novo funcional h, tal que:

$$J(m) = h(u(m), m) \tag{A.2}$$

O novo funcional h pode ser minimizado através da imposição de uma restrição do tipo:

$$F(u(m), m) = 0 \tag{A.3}$$

onde F é chamado de problema direto ou equação direta.

Utilizando um funcional aumentado ou Lagrangiano associado $\mathcal{L}$, podemos definir

$$\mathcal{L}(\tilde{u}, \tilde{\lambda}, m) = h(\tilde{u}, m) - \tilde{\lambda} F(\tilde{u}, m) \tag{A.4}$$

onde $\tilde{u}$ e $\tilde{\lambda}$ representam um valor qualquer de u e λ, respectivamente. O termo $\tilde{\lambda}$ é também denominado multiplicador de Lagrange ou variável adjunta.

Admitindo que $u = \tilde{u}$ e aplicando a condição dada pela equação A.3, podemos reescrever a equação A.4 como

$$\mathcal{L}(u, \tilde{\lambda}, m) = h(u, m) = J(m) \tag{A.5}$$

e como $\tilde{\lambda}$ não tem dependência com relação a m, o gradiente do funcional J será dado por:

$$\frac{\partial J}{\partial m} = \frac{\partial \mathcal{L}(u, \tilde{\lambda}, m)}{\partial m} = \frac{\partial \mathcal{L}(u, \tilde{\lambda}, m)}{\partial \tilde{u}} \frac{\partial \tilde{u}}{\partial m} + \frac{\partial \mathcal{L}(u, \tilde{\lambda}, m)}{\partial m} \tag{A.6}$$

A fim de evitar o cálculo dispendioso da derivada de Frechét, $\frac{\partial \tilde{u}}{\partial m}$, pode-se escolher um valor de $\tilde{\lambda} = \lambda$ de tal maneira que

$$\frac{\partial \mathcal{L}(u, \lambda, m)}{\partial \tilde{u}} = \frac{\partial h(u, m)}{\partial \tilde{u}} - \left(\frac{\partial F(u, m)}{\partial \tilde{u}} \right)^* \lambda = 0 \tag{A.7}$$

que é chamada de equação adjunta e o * operador adjunto.

Assim para o valor de $\tilde{\lambda} = \lambda$ a equação A.6 resulta em:

$$\frac{\partial J}{\partial m} = \frac{\partial \mathcal{L}(u, \lambda, m)}{\partial m} = \frac{\partial h(u, m)}{\partial m} - \lambda \frac{\partial F(u, m)}{\partial m} \tag{A.8}$$

Portanto $\mathcal{L}$ também pode ser visto como o Lagrangiano associado ao problema de minimização, e seu gradiente equivalente ao de J, avaliado nos pontos $\tilde{u} = u$ e $\tilde{\lambda} = \lambda$, que satisfazem, respectivamente, as equações de estado A.3 e adjunta A.7. Em síntese, para obter o gradiente de J, através do método adjunto, se faz necessário antes resolver as equações A.3 e A.7 e encontrar u e λ correspondentes.

A.1.1 Aplicação a FWI

A técnica dos multiplicadores de Lagrange permite encontrar o máximo ou o mínimo de uma função multivárivel suscetível a uma ou mais restrições (ver equação A.4). No caso da FWI no domínio do tempo, vamos identificar os componentes do problema genérico. O funcional $h(\mathbf{u}, \mathbf{m})$ é dado por:

$$h(\mathbf{u}, \mathbf{m}) = \frac{1}{2} \sum_{s,r} \int_0^T (S_{s,r}\mathbf{u} - \mathbf{d}_{obs})^2 dt \tag{A.9}$$

onde $S_{s,r}\mathbf{u}$ é o dado calculado, restrito por um operador $S_{s,r}$ (s e r, posição da fonte e receptores, respectivamente), com a mesma geometria do dado observado ($\mathbf{d}_{obs}$) e T é o tempo de registro. Cabe lembrar que $\mathbf{u}$ e $\mathbf{d}_{obs}$ são funções de $\mathbf{r}$ e t.

A restrição $F(\mathbf{u}, \mathbf{m}) = 0$, ou problema direto, é dada pela equação da onda acústica:

$$F(\mathbf{u}, \mathbf{m}) = \frac{1}{\upsilon^2} \frac{\partial^2 \mathbf{u}}{\partial t^2} - \nabla^2 \mathbf{u} - \mathbf{s} = 0 \tag{A.10}$$

vale ressaltar que υ é função somente de $\mathbf{r}$, enquanto o termo fonte ($\mathbf{s}$) é função de $\mathbf{r}$ e t. Quanto às condições de contorno, o campo de pressão $\mathbf{u}$ devido à fonte $\mathbf{s}$ satisfaz:

$$\mathbf{u}(t = 0) = 0$$

$$\frac{\partial \mathbf{u}(t = 0)}{\partial t} = 0 \tag{A.11}$$

Com esses elementos podemos escrever o funcional aumentado da equação A.4 como:

$$\mathcal{L}(\mathbf{u}, \lambda, \mathbf{m}) = \frac{1}{2} \sum_{s,r} \int_0^T (S_{s,r}\mathbf{u} - \mathbf{d}_{obs})^2 dt - \sum_{s,r} \underbrace{\int_0^T \int_{\mathbf{r}} \lambda \left[\frac{1}{\upsilon^2} \frac{\partial^2 \mathbf{u}}{\partial t^2} - \nabla^2 \mathbf{u} - \mathbf{s} \right] d\mathbf{r} dt}_{II} \tag{A.12}$$

onde $\lambda = \lambda(\mathbf{r}, t)$ é o multiplicador de Lagrange, também definido como campo adjunto.

O termo identificado como II na equação A.12 pode ser expandido como:

$$\int_0^T \int_{\mathbf{r}} \lambda \left[\frac{1}{\upsilon^2} \frac{\partial^2 \mathbf{u}}{\partial t^2} - \nabla^2 \mathbf{u} - \mathbf{s} \right] d\mathbf{r} dt =$$

$$\underbrace{\int_{\mathbf{r}}\int_0^T \lambda \frac{1}{v^2}\frac{\partial^2 \mathbf{u}}{\partial t^2} dt d\mathbf{r}}_{\mathbf{A}} + \underbrace{\int_0^T \int_{\mathbf{r}} -\lambda \nabla^2 \mathbf{u}\, d\mathbf{r} dt}_{\mathbf{B}} + \underbrace{\int_0^T \int_{\mathbf{r}} -\lambda \mathbf{s}\, d\mathbf{r} dt}_{\mathbf{C}} \tag{A.13}$$

Fazendo a integral por partes do termo A,

$$\int_0^T \lambda \frac{1}{v^2}\frac{\partial^2 \mathbf{u}}{\partial t^2}\, dt = \int_0^T g\, dh = [gh]_0^T - \int_0^T h\, dg \tag{A.14}$$

onde

$$g = \lambda$$

$$dg = \frac{\partial \lambda}{\partial t} dt$$

$$dh = \frac{1}{v^2}\frac{\partial^2 \mathbf{u}}{\partial t^2} dt$$

$$h = \frac{1}{v^2}\frac{\partial \mathbf{u}}{\partial t} \tag{A.15}$$

Logo o resultado da integração do termo A fica

$$\int_0^T \lambda \frac{1}{v^2}\frac{\partial^2 \mathbf{u}}{\partial t^2}\, dt = \left(\lambda \frac{1}{v^2}\frac{\partial \mathbf{u}}{\partial t}\right)|_0^T - \int_0^T \frac{1}{v^2}\frac{\partial \mathbf{u}}{\partial t}\frac{\partial \lambda}{\partial t} dt \tag{A.16}$$

Integrando novamente por partes a equação A.16, com $g = \frac{1}{v^2}\frac{\partial \lambda}{\partial t}$ e $dh = \frac{\partial \mathbf{u}}{\partial t} dt$

$$\int_0^T \lambda \frac{1}{v^2}\frac{\partial^2 \mathbf{u}}{\partial t^2}\, dt = \left(\lambda \frac{1}{v^2}\frac{\partial \mathbf{u}}{\partial t}\right)|_0^T - \left(\mathbf{u}\frac{1}{v^2}\frac{\partial \lambda}{\partial t}\right)|_0^T + \int_0^T \frac{1}{v^2}\mathbf{u}\frac{\partial^2 \lambda}{\partial t^2} dt \tag{A.17}$$

Utilizando as condições iniciais A.11 e as seguintes condições finais:

$$\lambda(t = T) = 0$$

$$\frac{\partial \lambda(t = T)}{\partial t} = 0 \tag{A.18}$$

a equação A.17 resulta em:

$$\int_0^T \lambda \frac{1}{v^2} \frac{\partial^2 \mathbf{u}}{\partial t^2}\, dt = \int_0^T \mathbf{u} \frac{1}{v^2} \frac{\partial^2 \lambda}{\partial t^2} dt \tag{A.19}$$

De maneira análoga, utilizando o mesmo procedimento (integração por partes) o termo **B** na equação A.13 obtemos:

$$\int_{\mathbf{r}} -\lambda \nabla^2 \mathbf{u}\, d\mathbf{r} = \int_{\mathbf{r}} -\mathbf{u} \nabla^2 \lambda\, d\mathbf{r} \tag{A.20}$$

Assim o Lagrangiano da equação A.12 pode ser reescrito como

$$\mathcal{L}(\mathbf{u}, \lambda, \mathbf{m}) = \frac{1}{2} \sum_{s,r} \int_0^T (S_{s,r}\mathbf{u} - \mathbf{d}_{obs})^2 dt - \sum_{s,r} \int_0^T \int_{\mathbf{r}} \left[\mathbf{u} \left(\frac{1}{v^2} \frac{\partial^2 \lambda}{\partial t^2} - \nabla^2 \lambda \right) - \lambda \mathbf{s} \right] d\mathbf{r} dt \tag{A.21}$$

Derivando a equação A.21 em relação a **u** e igualando a zero, encontra-se as equações adjuntas:

$$\frac{1}{v^2} \frac{\partial^2 \lambda}{\partial t^2} - \nabla^2 \lambda = \sum_r S_{s,r}^T (S_{s,r}\mathbf{u} - \mathbf{d}_{obs}) \tag{A.22}$$

A equação A.22 é equivalente à equação da onda acústica A.10, com o resíduo entre o dado calculado e o observado $(S_{s,r}\mathbf{u} - \mathbf{d}_{obs})$ assumindo o papel do termo fonte (**s**). No entanto, ela está sujeita às condições finais A.18 ao invés das iniciais A.11. Isso significa que o cálculo da variável adjunta λ é realizado através da propagação do resíduo de forma reversa no tempo.

Pela formulação do método adjunto, o gradiente de J em um ponto **r** será dado por:

$$\frac{\partial \mathcal{L}(\mathbf{u}, \lambda, \mathbf{m})}{\partial \mathbf{m}}(\mathbf{r}) = \frac{\partial J(\mathbf{m})}{\partial \mathbf{m}}(\mathbf{r}) = \frac{2}{v^3(\mathbf{r})} \sum_{s,r} \int_0^T \lambda \frac{\partial^2 \mathbf{u}}{\partial t^2} dt \tag{A.23}$$

onde o parmâmetro **m** corresponde a velocidade $v(\mathbf{r})$.

Em síntese o cálculo do gradiente de J, utilizando o método adjunto, pode ser descrito nas seguintes etapas:

1. Propagação direta do campo de ondas da fonte, utilizando a equação A.10
2. Cálculo do resíduo entre o dado simulado e o observado $(S_{s,r}\mathbf{u} - \mathbf{d}_{obs})$
3. Propagação reversa do resíduo, utilizando a equação adjunta A.22
4. Correlação cruzada para todos os tempos e todas as fontes utilizando a equação A.23.

Apêndice B

Cálculo do gradiente por diferenciação automática

Baseado no trabalho de Sun et al. (2019), vamos calcular o gradiente de **J** numa arquitetura de RNN usando a diferenciação automática, que consiste num conjunto de técnicas para numericamente avaliar a derivada de funções numéricas especificadas por um programa de computador. Tomemos a solução discreta da função objetivo dada pela equação (1.19):

$$\mathbf{J}(\upsilon(\mathbf{r})) = \frac{1}{2n_s} \sum_{\mathbf{r}_s} \sum_{\mathbf{r}_g} \sum_{t} \delta d_t^2, \tag{B.1}$$

onde $\mathbf{r}_s$ e $\mathbf{r}_g$ são vetores posição da fonte e do receptor, respectivamente; t é o tempo e $\delta d_t = d_t - \tilde{d}_t$, onde d_t são os dados de treinamento (dados observados) e $\tilde{d}_t$ os dados RNN modelados (dados calculados).

O gradiente da função objetivo, como já foi dito anteriormente, é escrito como:

$$\mathrm{g}(\mathbf{r}) = \frac{\partial \mathbf{J}}{\partial \upsilon_n(\mathbf{r})} \tag{B.2}$$

Desconsiderando o índice de iteração n, o gradiente de **J** com relação a RNN é dado por

$$g(\mathbf{r}) = \sum_{t}^{T} \left[\frac{\partial \mathbf{J}}{\partial \tilde{u}(\mathbf{r}, t)} \right] \frac{\partial \tilde{u}(\mathbf{r}, t)}{\partial \upsilon(\mathbf{r})}, \tag{B.3}$$

onde $\tilde{u}(\mathbf{r}, t)$ é o campo de pressão e T o tempo máximo. A derivada parcial $[\partial \mathbf{J} / \partial \tilde{u}(\mathbf{r}, t)]$ pode ser calculada em termos do campo de onda avaliado em incrementos de tempo dentro da RNN usando a regra da cadeia:

$$\left[\frac{\partial \mathbf{J}}{\partial \tilde{u}(\mathbf{r},t)}\right] = \left[\frac{\partial \mathbf{J}}{\partial \tilde{u}(\mathbf{r},t+2\Delta t)}\right]\frac{\partial \tilde{u}(\mathbf{r},t+2\Delta t)}{\partial \tilde{u}(\mathbf{r},t)} + \left[\frac{\partial \mathbf{J}}{\partial \tilde{u}(\mathbf{r},t+\Delta t)}\right]\frac{\partial \tilde{u}(\mathbf{r},t+\Delta t)}{\partial \tilde{u}(\mathbf{r},t)} + \frac{\partial \mathbf{J}}{\partial \tilde{u}(\mathbf{r},t)} \tag{B.4}$$

onde $0 \leq t \leq T$ e as condições iniciais para a retropropagação RNN são assumidas como zeros, i.e, $[\frac{\partial \mathbf{J}}{\partial \tilde{u}(\mathbf{r},t)}]_{t=T+1,T+2} = 0$.

As derivadas parciais de $\tilde{u}(\mathbf{r},t+2\Delta t)$ e $\tilde{u}(\mathbf{r},t+\Delta t)$ com relação $\tilde{u}(\mathbf{r},t)$, e $\tilde{u}(\mathbf{r},t)$ em relação a $v(\mathbf{r})$ são

$$\frac{\partial \tilde{u}(\mathbf{r},t+2\Delta t)}{\partial \tilde{u}(\mathbf{r},t)} = -1, \tag{B.5}$$

$$\frac{\partial \tilde{u}(\mathbf{r},t+\Delta t)}{\partial \tilde{u}(\mathbf{r},t)} = v^2(\mathbf{r})\Delta t^2\nabla^2 + 2, \tag{B.6}$$

$$\frac{\partial \tilde{u}(\mathbf{r},t)}{\partial v(\mathbf{r})} = \frac{2\Delta t^2}{v(\mathbf{r})}\frac{\partial \tilde{u}(\mathbf{r},t-\Delta t)}{\partial t^2} \approx \frac{2\Delta t^2}{v(\mathbf{r})}\frac{\partial \tilde{u}(\mathbf{r},t)}{\partial t^2} \tag{B.7}$$

Substituindo as equações (B.5) e (B.6) na equação (B.4) resulta em:

$$\left[\frac{\partial \mathbf{J}}{\partial \tilde{u}(\mathbf{r},t)}\right] = v^2(\mathbf{r})\Delta t^2\left(\nabla^2\left[\frac{\partial \mathbf{J}}{\partial \tilde{u}(\mathbf{r},t+\Delta t)}\right] - \frac{1}{n_s v^2(\mathbf{r})\Delta t^2}\sum_{\mathbf{r}_s}\sum_{\mathbf{r}_g}\delta d_t\right) + 2\left(\frac{\partial \mathbf{J}}{\partial \tilde{u}(\mathbf{r},t+\Delta t)}\right) - \left(\frac{\partial \mathbf{J}}{\partial \tilde{u}(\mathbf{r},t+2\Delta t)}\right) \tag{B.8}$$

A leitura que pode ser feita na equação (B.8) é a de que ela enfatiza que a derivada parcial da função objetivo em relação ao campo de onda previsto equivale à retropropagação do campo de onda residual no tempo. De forma que esta pode ser rescrita como:

$$\left[\frac{\partial \mathbf{J}}{\partial \tilde{u}(\mathbf{r},t)}\right] = BP\left(-\frac{1}{n_s v^2(\mathbf{r})\Delta t^2}\sum_{\mathbf{r}_s}\sum_{\mathbf{r}_g}\delta d_t\right) \tag{B.9}$$

Substituindo as equações (B.7) e (B.9) na equação (B.3), obtemos que:

$$g(\mathbf{r}) = \sum_{t}^{T} BP\left(-\frac{1}{n_s}\sum_{\mathbf{r}_s}\sum_{\mathbf{r}_g}\delta d_t\right)\frac{2}{v^3(\mathbf{r})}\frac{\partial \tilde{u}^2(\mathbf{r},t)}{\partial t^2} \tag{B.10}$$

Assim, o processo de diferenciação automática usando a regra da cadeia mostra que o gradiente pode ser obtido pela RNN. A interpretação para o resultado obtido na equação B.10 é a de que o gradiente obtido via treinamento RNN equivale à correlação cruzada, no tempo, da derivada parcial de segunda ordem do campo de onda modelado (propagação direta) e os resíduos (propagação reversa), e que o cálculo é particionado em sub-tarefas, de forma a otimizar a operação.

Referências Bibliográficas

Abadi, M., P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin et al. 2016. "Tensorflow: A system for large-scale machine learning". Em *12th USENIX Symposium on Operating Systems Design and Implementation,* 265–283. Savannah, USA.

Baydin, A. G., B. A. Pearlmutter, A. A. Radul e J. M. Siskind. 2018. "Automatic Differentiation in Machine Learning: a Survey". *arXiv preprint arXiv:1502.05767v4.*

Boonyasiriwat, C., P. Valasek, P. Routh, W. Cao, G. T. Schuster e B. Macy. 2009. "An efficient multiscale method for time-domain waveform tomography". *Geophysics* 74 (6): WCC59–WCC68. https://doi.org/10.1190/1.3151869.

Bottou, L. 1991. "Stochastic Gradient Learning in Neural Networks". Em *Proceedings of Neuro-Nîmes 91.* Nimes, France: EC2. http://leon.bottou.org/papers/bottou-91c.

Bourgeois, A., M. Bourget, P. Lailly, M. Poulet, P. Ricarte e R. Versteeg. 1991. "Marmousi, model and data". Em *The Marmousi Experience,* 5–16. Eur. Ass. Expl. Geophys.

Bunks, Carey, Fatimetou M. Saleck, S. Zaleski e G. Chavent. 1995. "Multiscale seismic waveform inversion". *Geophysics* 60, n. 5 (setembro): 1457–1473. https://doi.org/10.1190/1.1443880.

Chavent, G. 1974. "Identification of functional parameter in partial differential equations". Em *Identification of Parameters in Distributed Systems,* 31–48. ASME, NY.

Churchland, P. S., e T. J. Sejnowski. 1992. *The computational brain.* MIT Press.

Claerbout, J. F. 1971. "Toward a unified theory of reflector mapping". *Geophysics* 36 (4): 467–481.

———. 1976. *Fundamentals of Geophysical Data Processing.* McGraw-Hill.

Conceição, M. 2021. "Redes neurais recorrentes de Elman aplicadas à inversão de forma completa da onda acústica utilizando diferenciação automática". Dissertação de mestrado, Universidade Federal da Bahia.

dos Santos, A. W. G. 2013. "Inversão de forma de onda aplicada à análise de velocidades sísmicas utilizando uma abordagem multiescala". Dissertação de mestrado, Universidade Federal da Bahia.

Elman, J. L. 1990. "Finding structure in time". *Cognitive Science* 14 (2): 179–211. ISSN: 0364-0213. https://doi.org/https://doi.org/10.1016/0364-0213(90)90002-E. https://www.sciencedirect.com/science/article/pii/036402139090002E.

Fabien-Ouellet, G., e R. Sarkar. 2020. "Seismic velocity estimation: A deep recurrent neural-network approach." *Geophysic* 85 (1): U21–U29.

Fehler, M., e P. J. Keliher. 2011. *SEAM Phase I: Challenges of Subsalt Imaging in Tertiary Basins, with Emphasis on Deepwater Gulf of Mexico*. Society of Exploration Geophysicists, janeiro. ISBN: 9781560802877. https://doi.org/10.1190/1.9781560802945. https://doi.org/10.1190/1.9781560802945.

Fichtner, A. 2011. *Full Seismic Waveform Modelling and Inversion*. Springer Berlin, Heidelberg.

Gill, P. E., W. Murray e M. H. Wright. 1981. *Practical Optimization*. Academic Press.

Haykin, S. 2007. *Redes Neurais: Princípios e Prática*. Bookman.

Hughes, T. W., I. A. D. Williamson, M. Minkov e S. Fan. 2019. "Wave Physics as an Analog Recurrent Neural Network". *arXiv preprint arXiv:1904.12831v2*.

Jaeger, H. 2001. "The"echo state"approach to analysing and training recurrent neural networks-with an erratum note'". *Bonn, Germany: German National Research Center for Information Technology GMD Technical Report* 148 (janeiro).

Jordan, M. I. 1986. *Serial Order: A Parallel Distributed Processing Approach*. ICS report. Institute for Cognitive Science, University of California, San Diego.

Kingma, D. P., e J. L. Ba. 2014. "Adam: A Method for Stochastic Optimization". *arXiv preprint arXiv:1412.6980*.

Koehne, V. 2014. "FWI multiescala: uma implementação em GPU". Dissertação de mestrado, Universidade Federal da Bahia.

Lailly, P. 1983. "The seismic inverse problem as a sequence of before stack migrations". Em *Conference on inverse scattering: theory and application*, 206–220. Siam Philadelphia, PA.

Lang, K. J., A. H. Waibel e G. E. Hinton. 1990. "A time-delay neural network architecture for isolated word recognition". *Neural Networks* 3 (1): 23–43. ISSN: 0893-6080. https://doi.org/https://doi.org/10.1016/0893-6080(90)90044-L. https://www.sciencedirect.com/science/article/pii/089360809090044L.

Ma, Y., e D. Hale. 2012. "Quasi-Newton full-waveform inversion with a projected Hessian matrix". *Geophysics* 77 (5): R207–R216. https://doi.org/10.1190/geo2011-0519.1.

Maurya, S. P., N. P. Singh e K. H. Singh. 2020. *Seismic Inversion Methods: A Practical Approach*. Springer Cham.

McCulloch, W. S., e W. Pitts. 1943. "A logical calculus of the ideas immanent in nervous activity". *The Bulletin of Mathematical Biophysics* 5 (4): 115–133.

Nocedal, J., e S. Wright. 2006. *Numerical optimization*. Springer Science / Business Media.

Pasalic, Damir, e Ray Mcgarry. 2010. "Convolutional perfectly matched layer for isotropic and anisotropic acoustic wave equations", 2925–2929. Janeiro. https://doi.org/10.1190/1.3513453.

Paszke, A., S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga e Adam Lerer. 2016. "Automatic differentiation in PyTorch". Em *31st Conference on Neural Information Processing Systems (NIPS 2017).* Long Beach, USA.

Plessix, R.-E. 2006. "A review of the adjoint-state method for computing the gradient of a functional with geophysical applications". *Geophysical Journal International* 167 (2): 495–503. https://doi.org/10.1111/j.1365-246x.2006.02978.x.

Qian, N. 1999. "On the momentum term in gradient descent learning algorithms". *Neural networks : the official journal of the International Neural Network Society* 12 (1): 145–151.

Ramón y Cajál, S. 1911. *Histologie Du Systéme Nerveux de L'Homme Et Des Vertebres.* Maloine.

Ren, Y., X. Xu, S. Yang, L. Nie e Y. Chen. 2020. "A Physics-Based Neural-Network Way to Perform Seismic Full Waveform Inversion". *IEEE Access* 8:112266–112277. https://doi.org/10.1109/ACCESS.2020.2997921.

Richardson, A. 2017. "1D FWI using a neural network". https://github.com/ar4/nnfwi_1d.

———. 2018. "Seismic full-waveform inversion using deep learning tools and techniques". *arXiv preprint arXiv:1801.07232.*

———. 2021. "Deepwave", outubro. https://doi.org/10.5281/zenodo.3829886. https://doi.org/10.5281/zenodo.3829886.

Richter, M. 2020. *Inverse Problems.* Birkhäuser Cham.

Robbins, H., e S. Monro. 1951. "A Stochastic Approximation Method". *The Annals of Mathematical Statistics* 22 (3): 400–407. ISSN: 00034851, acesso em 30 de maio de 2022. http://www.jstor.org/stable/2236626.

Rosenblatt, F. 1958. "The perceptron: a probabilistic model for information storage and organization in the brain." *Psychological review* 65 (6): 386–408.

Rumelhart, D. E., G. E. Hinton e R. J. Williams. 1986. "Learning representations by back-propagating errors". *Nature* 323 (6088): 533–536.

Shalev-Shwartz, S., e S. Ben-David. 2014. *Understanding Machine Learning: From Theory to Algorithms.* USA: Cambridge University Press. ISBN: 1107057132.

Sun, J., Z. Niu, K. A. H. Innanen, J. Li e D. Trad. 2019. "A theory-guided deep-learning formulation and optimization of seismic waveform inversion". *Geophysics* 85 (2): R87–R99. https://doi.org/10.1190/geo2019-0138.1.

Tarantola, A. 1984. "Inversion of seismic reflection data in the acoustic approximation". *Geophysics* 49:1259–1266.

Tieleman, T., e G. Hinton. 2012. "Lecture 6.5-rmsprop: Divide the Gradient by a Running Average of Its Recent Magnitude". *COURSERA: Neural Networks for Machine Learning* 4:26–31.

Virieux, J., e S. Operto. 2009. "An overview of full-waveform inversion in exploration geophysics". *Geophysics* 74 (6): WCC1–WCC26.

Yang, F., e J. Ma. 2021. "Revisit Geophysical Imaging in A New View of Physics-informed Generative Adversarial Learning". *arXiv preprint arXiv:2109.11452v1.*

Yoon, D., Z. Yeeh e J. Byun. 2020. "Seismic data reconstruction using deep bidirectional long short-term memory with skip connections". *IEEE Geoscience and Remote Sensing Letters* 18 (7): 1298–1302.

Zeiler, M. D. 2012. "ADADELTA: An Adaptive Learning Rate Method". *arXiv preprint arXiv:1212.5701v1.*

Zhou, B., L. Gao e Y.-H. Dai. 2006. "Gradient Methods with Adaptive Step-Sizes". *Computational Optimization and Applications* 35:69–86. https://doi.org/10.1007/s10589-006-6446-0.

Zhou, H. 2014. *Practical Seismic Data Analysis.* Cambridge University Press.

Zhu, Weiqiang, Kailai Xu, Eric F Darve, Biondo Biondi e Gregory C. Beroza. 2021. "Integrating Deep Neural Networks with Full-waveform Inversion: Reparametrization, Regularization, and Uncertainty Quantification". *Geophysics.*