



AUTOCODIFICADORES CONVOLUCIONAIS PARA ATENUAÇÃO DE RUÍDOS EM DADOS SÍSMICOS

Carlos Salvador Neto

Dissertação de Mestrado apresentada ao Programa de Pós-graduação em Engenharia Civil, COPPE, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Mestre em Engenharia Civil.

Orientador: Alexandre Gonçalves Evsukoff

Rio de Janeiro
Novembro de 2020

AUTOCODIFICADORES CONVOLUCIONAIS PARA ATENUAÇÃO DE
RUÍDOS EM DADOS SÍSMICOS

Carlos Salvador Neto

DISSERTAÇÃO SUBMETIDA AO CORPO DOCENTE DO INSTITUTO
ALBERTO LUIZ COIMBRA DE PÓS-GRADUAÇÃO E PESQUISA DE
ENGENHARIA DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO
PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU
DE MESTRE EM CIÊNCIAS EM ENGENHARIA CIVIL.

Orientador: Alexandre Gonçalves Evsukoff

Aprovada por: Prof. Alexandre Gonçalves Evsukoff
Prof. Luiz Landau
Prof. Sergio Lima Netto
Prof. Felix Thadeu Teixeira Gonçalves

RIO DE JANEIRO, RJ – BRASIL
NOVEMBRO DE 2020

Salvador Neto, Carlos

Autocodificadores convolucionais para atenuação de ruídos em dados sísmicos/Carlos Salvador Neto. – Rio de Janeiro: UFRJ/COPPE, 2020.

XV, 109 p.: il.; 29,7cm.

Orientador: Alexandre Gonçalves Evsukoff

Dissertação (mestrado) – UFRJ/COPPE/Programa de Engenharia Civil, 2020.

Referências Bibliográficas: p. 96 – 106.

1. Redes Neurais Profundas. 2. Autocodificadores Convolucionais. 3. Processamento Sísmico. I. Gonçalves Evsukoff, Alexandre. II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia Civil. III. Título.

Agradecimentos

Inicialmente minha gratidão a Deus e a todos que tornaram possível a produção deste trabalho. Agradeço, igualmente, ao Laboratório de Métodos Computacionais (LAMCE/Lab2M) pelo auxílio prestado, principalmente na pessoa do Bruno de Souza Silva pelo fundamental suporte em Geofísica, direcionamento deste trabalho e criação dos dados sintéticos, a Candy Shirley Rosa Contreras pelo processamento dos dados sísmicos reais e aos colegas de trabalho do projeto Petrobrás nas pessoas de Manuel Vargas, Carlos Eduardo Menezes dos Anjos, Julio de Castro Vargas Fernandes e aos demais colegas de Mestrado, sem os quais todo o ensinamento aprendido e demonstrado na presente dissertação não teria sido possível. Agradeço também ao LNCC pela utilização da infraestrutura do supercomputador Santos Dumont para o treinamento dos modelos de aprendizado de máquina, ainda, agradeço a minha família e noiva por todo apoio e suporte nesses anos de Mestrado, por fim ao meu orientador e aos meus professores da UFRJ.

Resumo da Dissertação apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.)

AUTOCODIFICADORES CONVOLUCIONAIS PARA ATENUAÇÃO DE RUÍDOS EM DADOS SÍSMICOS

Carlos Salvador Neto

Novembro/2020

Orientador: Alexandre Gonçalves Evsukoff

Programa: Engenharia Civil

O método sísmico se destaca por se mostrar essencial para as indústrias petrolíferas na identificação e caracterização de reservatórios de hidro-carbonetos, sendo capaz de fornecer informações importantes a respeito da estrutura geológica, conduzindo assim, a um quadro detalhado da geologia de subsuperfície. Dados sísmicos são observações físicas, medições ou estimativas sobre fontes sísmicas, ondas sísmicas e seus meios de propagação. O objetivo de adquirir e processar dados sísmicos é aprender algo sobre o Interior da terra. Para isso, é preciso inicialmente estabelecer relações específicas entre os objetivos pretendidos e os parâmetros mensuráveis. O primeiro passo é realizar a aquisição de dados projetados para o problema, em seguida os dados são processados para identificar e aprimorar o sinal desejado, e, por fim, são realizadas interpretações dos dados com base nos dados processados. Este estudo tem como foco principal uma das primeiras etapas do processamento sísmico e tem como objetivo melhorar a relação sinal-ruído dos dados sísmicos, desta forma, o trabalho propõe uma abordagem baseada em modelos de aprendizado profundo para atenuar diretamente ruídos como *swell*, ruídos lineares e ruídos aleatórios em dados pré-empilhamento. A rede aprende como detectar diretamente o ruído e, em seguida, obtém dados atenuados, removendo os ruídos do conjunto de dados brutos corrompidos. A avaliação dos resultados deste método será baseada com utilização de funções de avaliação como a relação sinal ruído ou erro médio quadrático e através de uma análise visual da qualidade imagem sísmica a fim de verificar a recuperação dos sinais sísmicos danificados por ruídos. Os testes foram realizados utilizando redes totalmente conectadas e os resultados da relação sinal-ruído e o erro quadrático médio indicam que a abordagem de aprendizado profundo tem a capacidade de atenuar ruídos sem danificar os sinais primários.

Abstract of Dissertation presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

CONVOLUTIONAL AUTOENCODERS FOR SEISMIC DATA DENOISE

Carlos Salvador Neto

November/2020

Advisor: Alexandre Gonçalves Evsukoff

Department: Civil Engineering

The seismic method stands out for being essential for the oil industries in the identification and characterization of hydro-carbide reservoirs, being able to provide important information regarding the geological structure, thus leading to a detailed picture of the subsurface geology. Seismic data are physical observations, measurements or estimates of seismic sources, seismic waves and their means of propagation. The objective of acquiring and processing seismic data is to learn something about the Earth's Interior and to understand certain aspects of the Earth, it is necessary initially to establish some specific relationships between the intended objectives and the measurable parameters. The first step is to perform the data acquisition projected for the problem, then the data is processed to identify and improve the desired signal, and finally, interpretations of the data are performed based on the processed data. This work has as main focus one of the first stages of seismic processing, and aims to improve the signal-to-noise ratio of seismic data. For this purpose, the work proposes an approach based on deep learning models to directly attenuate noise such as Swell, linear and random noises in pre-stack data. The network learns how to directly detect noise and then obtains attenuated data, removing noise from the corrupted raw data set. The evaluation of the results of this method will be based on the use of evaluation functions such as the signal-to-noise ratio or mean quadratic error and through a visual analysis of the seismic image quality in order to verify the recovery of seismic signals damaged by noise. The tests were performed using fully connected networks and the results of the signal-to-noise ratio and the mean square error indicate that the deep learning approach has the ability to attenuate noise without damaging the primary signals.

Sumário

Lista de Figuras	ix
Lista de Tabelas	xv
1 Introdução	1
1.1 Motivação	2
1.2 Proposta e Objetivo	3
1.3 Contribuição da Dissertação	4
1.4 Organização da Dissertação	5
2 Processamento Sísmico	7
2.1 Introdução	7
2.2 Ruídos na sísmica marinha	11
2.3 Métodos de atenuação de ruído	15
2.4 Filtragem	16
2.4.1 Filtragem no tempo	17
2.4.2 Filtragem no tempo e espaço	22
2.5 <i>Muting</i>	25
2.6 Resumo do capítulo	26
3 Revisão Bibliográfica	27
3.1 Redes Neurais Profundas	27
3.1.1 Redes Neurais Convolucionais (CNN)	28
3.1.2 Evolução das Redes Neurais Convolucionais	30
3.1.3 Redes Autocodificadores	33
3.2 Aprendizado profundo para atenuação de ruído em dados sísmicos	35
3.3 Resumo do capítulo	37
4 Materiais e Métodos	38
4.1 Dados	38
4.1.1 Formato SEG-Y	38
4.1.2 Dados Sintéticos	39

4.1.3	Dados Reais	43
4.1.4	Pré-processamento nos Dados Reais	46
4.2	Modelos	53
4.2.1	CNN com três camadas	55
4.2.2	<i>U-Net</i>	55
4.2.3	Comparação entre os modelos	57
4.3	Procedimentos experimentais	58
4.3.1	Pré-processamento	59
4.3.2	Divisão dos Dados	60
4.3.3	Treinamento	60
4.4	Resumo do capítulo	61
5	Resultados e Discussões	62
5.1	Estatísticas de Validação	62
5.2	Análise Quantitativa	63
5.2.1	Ruídos Aleatórios	63
5.2.2	Ruídos Lineares	64
5.2.3	Ruídos <i>Swell</i>	64
5.3	Análise qualitativa entre sísmicas de mesma aquisição	66
5.3.1	Ruídos Aleatórios	67
5.3.2	Ruídos Lineares	69
5.3.3	Ruído <i>Swell</i>	73
5.3.4	Dado sísmico por faixas de frequência	79
5.4	Análise qualitativa entre sísmicas de diferentes aquisições	81
5.4.1	Ruído <i>Swell</i>	81
5.4.2	Ruídos Lineares	86
5.5	Análise quantitativa e qualitativa entre modelos <i>U-Net</i> e CNN com três camadas entre sísmicas de mesma aquisição	88
5.6	Resumo do capítulo	92
6	Conclusões e trabalhos futuros	93
	Referências Bibliográficas	96
A	Anexo	107
A.1	Ferramentas	107
A.1.1	Python	107
A.1.2	Anaconda	108
A.1.3	Tensorflow	109

Lista de Figuras

1.1	Estágios básicos de um projeto de reflexão sísmica convencional, consistindo em aquisição, processamento e interpretação de dados e etapas principais do processamento de dados, juntamente com a recorrência do controle de qualidade (CQ) no processamento. Adaptado de: [1]	2
1.2	Dado sísmico pré-empilhamento com e sem a presença de ruído	4
2.1	Representação da aquisição sísmica marítima (a) e terrestre (b). Adaptado de: [2], [3]	8
2.2	Equipamento geral utilizado em aquisições sísmicas marítimas (A) 2D e (B) 3D Adaptado de: [1]	9
2.3	Cabo rebocado por um navio registrando sinais sísmicos gerados por uma fonte de ar comprimido Adaptado de: [1]	10
2.4	Trajetos de raio de disparos e exibição esquemática gravados em seis (6) receptores Adaptado de: [1]	10
2.5	Diferentes tipos de ruído coerentes e aleatórios observados especificamente em dados sísmicos marinhos. Nas caixas amarelas os tipos de ruídos coerentes abrangem o ruído gerado por componentes técnicos do sistema de pesquisa, enquanto as caixas azuis indicam o ruído do ambiente geológico. Adaptado de: [1]	12
2.6	Ruidos coerentes. (a) Ruído operacional, causado pela movimento do navio (b) Efeito múltiplas. (c) Ruído <i>swell</i> . (d) Efeito bolha, causado pela bomba de ar comprimido (<i>airgun</i>). Adaptado de: [1]	14
2.7	(a) Ruído de cabo mecânico (coerente). (b) Ruído de pássaro. (c) Ruído de boia (coerente). Adaptado de: [1]	14

2.8	Volume de dados sísmicos	17
2.9	Ilustração esquemática de passa-banda com diferentes projetos de filtro no domínio da frequência. (A) passa-baixa, (B) passa-alta, (C) passa-banda e (D) filtros de entalhe. f_1 e f_2 representam frequências de corte Adaptado de: [1]	18
2.10	(A) Um espectro de amplitude em forma de vagão no domínio da frequência e (B) sua contraparte no domínio do tempo, a função Sinc. Adaptado de: [1]	19
2.11	Fluxo de trabalho para o processo de filtragem de passagem de banda nos domínios (A) de frequência e (B) de tempo. (C) A filtragem é realizada utilizando a multiplicação (\times) dos espectros de amplitude do operador sísmico e de filtro bruto no domínio da frequência (painel superior) ou por uma convolução ($*$) dos próprios dados sísmicos com o operador de filtro no domínio do tempo (painel inferior) Adaptado de: [1]	20
2.12	Representação esquemática de um projeto de operador de filtro passa-banda.	21
2.13	Ilustração esquemática da modificação do espectro de amplitude no domínio da frequência. (A) Um esquema espectro de amplitude de um tiro bruto. (B) Filtro <i>Butterworth</i> . (C) Espectro de amplitude esquemático do tiro filtrado	21
2.14	(A) Ilustração esquemática dos eventos comuns em um tiro marinho (parte superior) e exibições correspondentes dos mesmos eventos em um tiro marinho real (parte inferior). (B) Os espectros F-K esquemáticos (parte superior) e reais (parte inferior) do exemplo captado. (P), reflexão primária; (R), onda refratada; (D), onda direta; (B), ruído de pássaro; (M), reflexões múltiplas; (S), eventos de imersão negativos (dispersos)	23
2.15	Painéis negativos e positivos de um espectro F-K e aparência de domínio F-K de eventos comuns encontrados em tiro marinho. Adaptado de: [1]	24
2.16	Polígono definido para discriminar as zonas de sinal e ruído no domínio F-K. (A) Polígono de proteção (B) a saída do filtro F-K	24
2.17	(A) Exemplo de tiro sísmico sem aplicação de <i>mute</i> no topo. (B) Exemplo de tiro sísmico com aplicação de <i>mute</i> no topo	25
3.1	Exemplo de estrutura da uma imagem 2D	29
3.2	Exemplo de filtro convolucional	29

3.3	Exemplo da aplicação de convolucional com uma matriz de entrada 4x4, filtro convolucional de 2x2, função de ativação e execução do redimensionamento espacial com <i>Max Pooling</i> 2x2, deslocamento 2.	30
3.4	Arquitetura <i>AlexNet</i>	31
3.5	Arquitetura <i>VGG16</i>	31
3.6	Bloco <i>Inception</i>	32
3.7	Arquitetura <i>Inception</i>	32
3.8	Bloco <i>ResNet</i>	33
3.9	Arquitetura <i>ResNet</i>	33
3.10	Um exemplo de autocodificador. A imagem de entrada é codificada para uma representação compactada e decodificada.	33
3.11	Arquitetura <i>autoencoder</i> com três camadas	35
3.12	Estrutura <i>convolucional autoencoder</i> : cada caixa azul representa a camada de convolução e a camada cinza, o código de vetor latente. As camadas de <i>pooling</i> foram ignoradas	35
4.1	Os dados sísmicos são registrados em linhas de amostras - amostras ao mesmo tempo em canais consecutivos. A demultiplexação envolve a classificação dos dados em colunas de amostras - todas as amostras em um canal seguidos por outros canais seguintes.	39
4.2	Estrutura de um dado sísmico no formato SEG-Y	39
4.3	Modelo de Velocidade - M1	40
4.4	Tiros gerados a partir do Modelo de Velocidade M1	40
4.5	Modelo de Velocidade - M2	40
4.6	Tiros gerados a partir do Modelo de Velocidade M2	41
4.7	Tiros com ruído gaussiano de 10dB gerados a partir do Modelo de Velocidade M1	41
4.8	Tiros com ruído linear gerados a partir do Modelo de Velocidade M1	42
4.9	Tiros com ruído linear gerados a partir do Modelo de Velocidade M2	42
4.10	Tiros com ruído <i>swell</i> retirados do dado sísmico pré-empilhamento da bacia do Espírito Santo e adicionado ao dado sísmico sintético gerado pelo Modelo de Velocidade M2	43
4.11	Localização geográfica das linhas sísmicas da bacia de Santos - campo de produção Estrela do Mar	44
4.12	Exemplo de tiros do dado sísmico pré-empilhamento da bacia de Santos (linha 541)	44
4.13	Localização geográfica das linhas sísmicas da bacia do Espírito Santo	45
4.14	Exemplo de tiros do dado sísmico pré-empilhamento da bacia do Espírito Santo	45

4.15	Tiros do dados sísmico pré-empilhamento da bacia do Espírito Santo em seu estado bruto	47
4.16	Tiros do dados sísmico pré-empilhamento da bacia do Espírito Santo em seu estado bruto e com filtragens com filtro passa-banda com faixas de frequência de 5-20 Hz e 30-80 Hz	47
4.17	Tiros dos dados sísmico pré-empilhamento da bacia do Espírito Santo filtrada técnica <i>Butterworth</i>	48
4.18	Exemplo de um tiro e seu respectivo espectro F-K do dado sísmico pré-empilhamento localizado na bacia do Espírito Santo. Em (A) encontra-se o dado em sua forma bruta e seu respectivo espectro F-K. Em (B), o resultado do tiro após filtragem utilizando técnica F-K	49
4.19	Exemplo de tiros dos dados sísmicos pré-empilhamento da bacia do Espírito Santo filtrada com técnica F-K	50
4.20	Exemplo de um tiro e seu respectivo espectro F-K do dado sísmico pré-empilhamento localizado na bacia do Espírito Santo. Em (A) encontra-se o dado em sua forma bruta e seu respectivo espectro F-K. Em (B) o resultado do tiro após filtragem utilizando técnica F-K e aplicação do <i>muting</i> na coluna de água	51
4.21	Exemplo de tiros dos dados sísmicos pré-empilhamento da bacia do Espírito Santo filtrados com técnica F-K e aplicação do <i>muting</i> na coluna de água	52
4.22	Arquitetura CNN com três camadas	55
4.23	Arquitetura <i>U-Net</i>	56
5.1	Resultado tiro 115 com ruído aleatório 10dB	67
5.2	Resultado tiro 115 com ruído aleatório 10dB	68
5.3	Resultado tiro 115 com ruído aleatório 10dB	68
5.4	Espectro de frequência ruído aleatório 10, 15 e 20dB	69
5.5	Resultado tiro 36 com ruído linear M1	70
5.6	Resultado tiro 36 com ruído linear M1	70
5.7	Resultado tiro 36 com ruído linear M1	71
5.8	Resultado tiro 36 com ruído linear M2	71
5.9	Resultado tiro 36 com ruído linear M2	72
5.10	Resultado tiro 36 com ruído linear M2	72
5.11	Espectro de frequência ruído linear M1 e M2	73
5.12	Resultado tiro 36 com ruído <i>swell</i> M2	74
5.13	Resultado tiro 36 com ruído <i>swell</i> M2	74
5.14	Resultado tiro 36 com ruído <i>swell</i> M2	75
5.15	Resultado tiro 6170 com ruído <i>swell</i> com filtro passa-banda	76

5.16	Resultado tiro 6170 com ruído <i>swell</i> com filtro passa-banda	76
5.17	Resultado tiro 6170 com ruído <i>swell</i> com filtro passa-banda	77
5.18	Espectro F-K tiro 6170, ruído <i>swell</i> com filtro passa-banda	78
5.19	Espectro de frequência tiro 6170, ruído <i>swell</i> com filtro passa-banda .	78
5.20	(a) Tiro de entrada, (b) Tiro predito (c) Diferença entre (a) e (b), representando o <i>swell</i> detectado	80
5.21	Resultado treinamento dado sísmico Espírito Santo e predição em dado sísmico da bacia de Santos	82
5.22	Resultado de treinamento com dado sintético M2 com ruído <i>swell</i> e predição em dado real da bacia do Espírito Santo com ruído <i>swell</i> . .	83
5.23	Resultado de treinamento com dado sintético M2 com ruído <i>swell</i> e predição em dado real da bacia do Espírito Santo com ruído <i>swell</i> . .	83
5.24	Resultado de treinamento om dado real da bacia do Espírito Santo (filtro passa-banda) com ruído <i>swell</i> e predição em dado M2 com ruído <i>swell</i>	84
5.25	Resultado de treinamento om dado real (filtro passa-banda) da bacia do Espírito Santo com ruído <i>swell</i> e predição em dado M2 com ruído <i>swell</i>	84
5.26	Resultado de treinamento om dado real da bacia do Espírito Santo (filtro passa-banda) com ruído <i>swell</i> e predição em dado M2 com ruído <i>swell</i>	85
5.27	Resultado de treinamento com dado real da bacia do Espírito Santo (filtro F-K e <i>mute</i> no topo) com ruído <i>swell</i> e predição em dado M2 com ruído <i>swell</i>	85
5.28	Resultado de treinamento om dado real da bacia do Espírito Santo (filtro F-K e <i>mute</i> no topo) com ruído <i>swell</i> e predição em dado M2 com ruído <i>swell</i>	86
5.29	Resultado de treinamento com dado real da bacia do Espírito Santo (filtro F-K e <i>mute</i> no topo) com ruído <i>swell</i> e predição em dado M2 com ruído <i>swell</i>	86
5.30	Resultado de treinamento dado sísmico modelo M1 e predição dado sísmico modelo M2	87
5.31	Resultado de treinamento dado sísmico modelo M2 e predição dado sísmico modelo M1	87
5.32	Resultado tiro 6170 referente ao modelo CNN com três camadas, com ruído <i>swell</i> e com filtro passa-banda	89
5.33	Resultado tiro 6170 referente ao modelo CNN com três camadas, com ruído <i>swell</i> e com filtro passa-banda	90

5.34	Resultado tiro 6170 referente ao modelo CNN com três camadas com ruído <i>swell</i> com filtro passa-banda	90
5.35	Resultado tiro 6170 referente ao modelo <i>U-Net</i> com ruído <i>swell</i> com filtro passa-banda	91
5.36	Resultado tiro 6170 com ruído <i>swell</i> e filtro passa-banda	91
5.37	Resultado tiro 6170 referente ao modelo <i>U-Net</i> com ruído <i>swell</i> com filtro passa-banda	92

Lista de Tabelas

4.1	Dados Sísmicos pré-empilhamento reais e sintéticos utilizados neste trabalho	53
4.2	Comparação de parâmetros treináveis entre os modelos CNN com três camadas e <i>U-Net</i>	58
4.3	Horas de treinamento dos modelos e dos dados	61
5.1	Resultados dos modelos propostos para ruído aleatório com intensidades de 10, 15 e 20dB - PSNR e MSE	64
5.2	Resultados dos Modelos propostos para ruídos lineares nos dados sísmicos M1 e M2 - PSNR e MSE	65
5.3	Resultados dos modelos propostos para dado real com filtragem passa-banda - PSNR e MSE	66
5.4	Análise qualitativa - Dados sísmicos utilizados para treinamento e teste	81
5.5	Resultados dos Modelos <i>U-Net</i> e CNN com três camadas	88

Capítulo 1

Introdução

O petróleo constitui-se, na atualidade, como matéria-prima de reconhecido valor econômico, assumindo posição de destaque tanto nas atividades que esse recurso torna viável, na forma de combustíveis automotivo ou na queima em usinas termoelétricas, quanto no uso para fabricação de produtos como: plásticos, tintas, borrachas sintéticas, lubrificantes, solventes. Entretanto, a exploração de petróleo traz à tona uma série de desafios técnicos e práticos aos matemáticos, físicos, geofísicos, geólogos e engenheiros. A busca por reservas de petróleo economicamente viáveis envolve a aplicação de métodos de levantamento geofísico, como: o método sísmico, o método gravitacional, o método elétrico e o método magnético. Esses métodos visam investigar a correta estrutura da subsuperfície terrestre e suas propriedades.

Dentre as técnicas citadas destaca-se o método sísmico, por se mostrar essencial para as indústrias petrolíferas na identificação e caracterização de reservatórios de hidro-carbonetos, sendo capaz de fornecer informações importantes a respeito da estrutura geológica com a utilização de dados sísmicos, conduzindo, assim, a um quadro detalhado da geologia de subsuperfície.

Os dados sísmicos são observações físicas, medições ou estimativas sobre fontes sísmicas, ondas sísmicas e seus meios de propagação. A forma dos dados sísmicos varia e pode incluir gráficos analógicos, séries temporais, mapas, texto ou até ideias em alguns casos. O objetivo de adquirir e processar dados sísmicos é aprender algo sobre o Interior da Terra [4]. Para entender certos aspectos da Terra, é preciso inicialmente estabelecer algumas relações específicas entre os objetivos pretendidos e os parâmetros mensuráveis. Sendo assim, o primeiro passo é realizar a aquisição de dados projetados para o problema, em seguida, os dados são processados para identificar e aprimorar o sinal desejado, e, por fim, são realizadas interpretações dos dados com base nos dados anteriores processados, como apresentado na figura 1.1. Yilmaz [5] define os principais passos do processamento sísmico como: (i) deconvolução, (ii) empilhamento e (iii) migração (figura 1.1). É importante ressaltar que

os dados sísmicos são denominados como pós-empilhamento após aplicação desses três passos citados, e denominados como pré-empilhamento no estágio anterior à aplicação desses passos.

Outras etapas de processamento podem ser consideradas como métodos que contribuem para preparar entradas adequadas para estas três etapas principais de processamento, denominadas estas como pré-processamento.



Figura 1.1: Estágios básicos de um projeto de reflexão sísmica convencional, consistindo em aquisição, processamento e interpretação de dados e etapas principais do processamento de dados, juntamente com a recorrência do controle de qualidade (CQ) no processamento. Adaptado de: [1]

1.1 Motivação

O processamento sísmico é uma etapa do processo de exploração no setor de óleo e gás. O objetivo, nesta etapa, é o tratamento de dados sísmicos adquiridos por meio de um levantamento sísmico no qual são gerados dados sísmicos. Nestes dados são encontradas informações relevantes como o sinal e componentes indesejáveis, como ruídos. Para mitigar esse problema, aplica-se um importante procedimento conhecido como pré-processamento cujo objetivo comum em uma etapa de processamento, em particular, é melhorar a relação sinal ruído.

Os registros sísmicos pré-empilhamento adquiridos são inevitavelmente contaminados por ruídos e sinais diferentes provenientes de outras fontes, como ondas do mar, vento, ruído instrumental, entre outros. Muitos desses ruídos podem comprometer esses registros gerando consequências relevantes na exploração de óleo e gás. Desse modo, a atenuação do ruído tem um papel fundamental no processamento e interpretação dos dados sísmicos. Tipicamente, os métodos de atenuação de ruído são baseados em técnicas de filtragem, com as quais se pressupõe um conhecimento prévio sobre as características do ruído. Uma das etapas usualmente realizadas no

processamento sísmico para atenuação de ruídos é a análise do espectro de *Fourier*. No entanto, essa abordagem não é efetiva quando o ruído e o sinal sísmico ocupam a mesma faixa de frequência. Além disso, a parametrização da filtragem não é intuitiva, pois tipicamente variam com o tempo e podem alterar fortemente a forma do campo de onda de tal maneira que prejudique as etapas subsequentes [5].

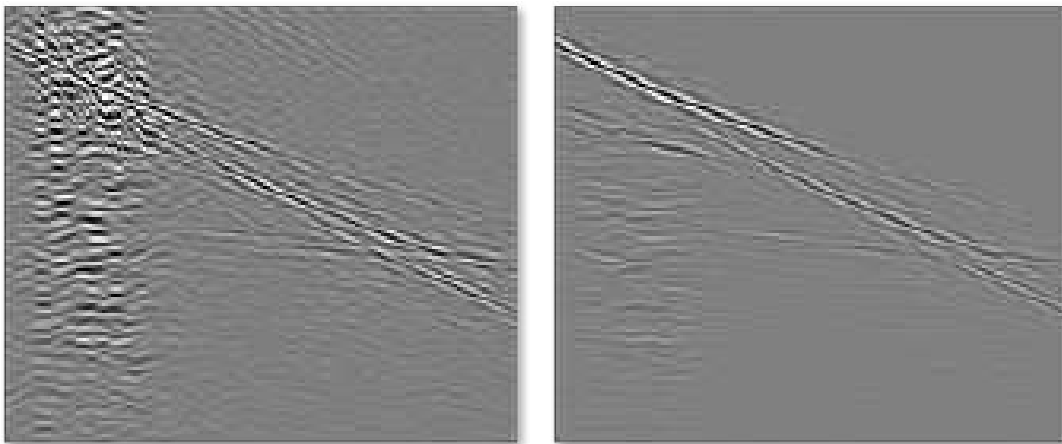
Por conseguinte, técnicas de aprendizado de máquinas desempenharam um papel importante na pesquisa de sistemas reconhecimento de padrões. Recentemente, métodos de aprendizado profundo têm sido amplamente aplicados dentro de diversos campos de pesquisa com abrangência desde processamento de imagens [6] em tarefas como classificação [7], localização e detecção de objetos [8], segmentação semântica [9] e segmentação de instâncias [10] até o processamento de sinais de áudio para reconhecimento de fala [11] e outros problemas de processamento de linguagem natural como tradução automática [12], análise de sentimentos [13], entre outras. Outrossim, uma das arquiteturas de aprendizado profundo que possui expressivo destaque na literatura são as redes neurais convolucionais (CNN) [14] [15] [16]. Destaca-se, portanto, o surgimento desse tipo de arquitetura que possibilitou a extração de características sem a necessidade do especialista para fazer a escolha dos métodos de extração de característica e o ajuste de seus parâmetros. Sendo assim, essa arquitetura de aprendizado profundo torna-se exequível uma vez que os parâmetros existentes nas redes neurais profundas são ajustados de maneira automática pelo algoritmo de retropropagação do erro [17]. Na área de geociência, esses métodos também são pesquisados de modo significativo na literatura, na qual é possível encontrar diversos nichos de pesquisa, desde aplicações em dados de perfil de poço até imagens sísmicas. Aplicações como essas introduzem novos desafios aos campos de pesquisa, especialmente, na área de aprendizado de máquina, em função da complexidade e do volume de dados a serem analisados inerentes a cada problema. Do mesmo modo, técnicas de aprendizado profundo conhecidas como Autocodificadores para atenuação de ruídos têm sido utilizadas na remoção de ruídos em diferentes domínios de dados, como imagens e sinais de áudio. Este tipo de método é capaz de aprender, simultaneamente, uma representação compacta ou esparsa dos dados, segundo requerido, e uma função não linear que mapeia esta representação com o propósito de gerar a saída desejada, isto é, o dado sem a presença de ruído.

1.2 Proposta e Objetivo

Este trabalho tem como foco principal uma das primeiras etapas do processamento sísmico e tem como objetivo melhorar a relação sinal-ruído dos dados sísmicos, sendo assim, o trabalho propõe uma abordagem baseada em modelos de aprendizado profundo para atenuar diretamente ruídos como *swell*, ruídos lineares e ruídos aleatórios

em dados pré-empilhamento. A rede aprende como detectar diretamente o ruído e, em seguida, obtém dados atenuados, removendo os ruídos do conjunto de dados brutos corrompidos. A avaliação dos resultados deste método será baseada com utilização de funções de avaliação como a relação sinal ruído ou erro médio quadrático e através de uma análise visual da qualidade imagem sísmica a fim de verificar a recuperação dos sinais sísmicos danificados por ruídos.

Os testes foram realizados utilizando redes totalmente conectadas e os resultados da relação sinal-ruído e o erro quadrático médio indicam que a abordagem de aprendizado profundo tem a capacidade de atenuar ruídos sem danificar os sinais primários. Neste trabalho foram utilizados dados sísmicos pré-empilhamento sintéticos e reais, e para treinamento dos modelos foi necessário a utilização dos respectivos dados com e sem ruídos, como mostrado na figura 1.2. Para a geração dos dados sintéticos foi utilizado a técnica de modelagem sísmica a partir do método de diferenças finitas [18], onde as reflexões de campos de ondas são geradas através de uma fonte artificial. Os dados sintéticos com ruídos foram obtidos pela adição de ruído com distribuições conhecidas. A hipótese subjacente de ruído aditivo é coerente com o tipo de ruído encontrado em casos reais. Já para os dados reais, foi obtida uma sísmica marinha em sua forma bruta majoritariamente com ruídos *swell*, e para a obtenção da respectiva linha sísmica sem ruídos foi necessário o seu processamento sísmico por meio de técnicas de filtragem.



(a) Sísmica com a presença de ruído.

(b) Sísmica sem a presença de ruído

Figura 1.2: Dado sísmico pré-empilhamento com e sem a presença de ruído

1.3 Contribuição da Dissertação

A presente dissertação contribuirá para o estudo da atenuação de ruídos em dados sísmicos pré-empilhamento reais e sintéticos. Esta atividade é executada por

um profissional da área de geofísica onde normalmente são empregadas técnicas de filtragens de frequência a fim de atenuar os ruídos presentes. Este procedimento é manual e muito dispendioso uma vez que a medida que os filtros são aplicados é necessário realizar uma revisão da qualidade nos resultados intermediários para averiguar se sinais primários foram danificados. Em caso de danificação do sinal primário, o geofísico normalmente desfaz o processo para aplicar novamente o filtro (ou outro filtro) com parâmetros diferentes. Em um levantamento sísmico de vasta extensão, grande quantidade de dados sísmicos são gerados e com isso um grande esforço é empregado pelo geofísico ocasionando um elevado tempo de trabalho que tem por objetivo. demandado no processo [1]. A presente pesquisa tem por objetivo acelerar o processamento de atenuação de ruídos automatizando parte deste processo através da utilização de um modelo convolucional para aprender o processo de atenuação de ruídos sísmicos. O modelo será treinado a fim de aprender a atenuar os ruídos sísmicos de um levantamento específico e para isto foi preciso utilizar dados sísmicos já processados por um geofísico em um levantamento específico. Uma vez o modelo treinado, pode-se aplicá-lo ao restante dos dados sísmicos não processados com o propósito de atenuar os ruídos. Foram implementados modelos de aprendizado profundo com arquitetura do tipo autocodificador convolucional com a utilização da linguagem *Python* [19] e a biblioteca *Tensorflow* [20]. Devido ao uso de grande quantidade de dados sísmicos pré-empilhamento, diferentes tipos de testes realizados com ruídos distintos e emprego de redes convolucionais profundas, o processo de treinamento foi desenvolvido para a utilização de múltiplas GPUs e a infraestrutura utilizada para tal atividade foi um servidor Nvidia DGX2 com 8 GPUs V100 que compõe o supercomputador Santos Dumont do Laboratório Nacional de Computação Científica (LNCC) [21]. O ajuste de parâmetros dos modelos utilizados consumiram treze (13) horas de processamento para o modelo mais simples e vinte e cinco (25) horas para o modelo mais complexo. O códigos desenvolvidos para esta dissertação estão disponibilizado no *github* através do endereço https://github.com/carlossneto/seismic_denoising.

1.4 Organização da Dissertação

O estudo proposto é um trabalho multidisciplinar que envolve temas nas áreas de inteligência computacional e geociência, com isso não será possível tratar todos os assuntos com o mesmo nível de profundidade. Pressupõe-se que o leitor tenha o embasamento em inteligência computacional, mais especificamente em redes neurais convolucionais uma vez que o enfoque deste trabalho será dado com esta visão [22]. Esta dissertação encontra-se estruturada em capítulos que trazem os contornos essenciais do estudo proposto. O segundo capítulo veicula introdução à sísmica no

qual são contemplados conceitos sobre levantamentos sísmicos, ruídos marinhos e seus impactos gerados em análises geológicas e o como é realizado o processamento deste ruído. No capítulo três é apresentada uma revisão bibliográfica de métodos computacionais para atenuação de ruídos sísmicos pré-empilhamento, além de ser introduzido conceito de redes neurais profundas e suas aplicações para atenuação de ruído. O quarto capítulo aponta a metodologia e os dados utilizados bem como as transformações necessárias para utilização dos mesmos e os modelos testados para o desenvolvimento deste trabalho. O capítulo cinco desdobra-se em apresentação e discussão dos resultados obtidos. Por fim, o capítulo seis expressa as conclusões do estudo, além de indicar trabalhos futuros, a seguir a bibliografia e os apêndices.

Capítulo 2

Processamento Sísmico

2.1 Introdução

O experimento sísmico de coleta de dados é denominado aquisição ou levantamento sísmico. Durante o levantamento, uma fonte controlada de onda sísmica é utilizada repetidamente. Cada acionamento ou disparo dessa fonte controlada representa um tiro e, no decorrer da aquisição, vários tiros são disparados e o resultado é capturado e registrado por um conjunto de receptores [23].

Neste estudo, aborda-se o levantamento sísmico terrestre e marítimo. Em um levantamento sísmico terrestre (figura 2.1(a)) é comum a utilização de explosivos ou caminhões vibradores como fontes sísmicas. Quando acionados ocorre uma vibração vertical que é registrada pelos receptores instalados no solo. Já em uma aquisição sísmica marítima (figura 2.1(b)) são utilizados dispositivos mecânicos conhecidos como bombas de ar comprimido que armazenam e liberam rapidamente o ar. Uma vez acionados, a energia liberada é convertida em ondas sonoras gerando o sinal sísmico que é registrado por um conjunto de receptores conectado a um cabo, que por sua vez, é rebocado por um navio.



(a) Levantamento sísmico marítimo. 1- Fonte Sísmica, 2- Representação de uma reflexão da estrutura geológica, 3 - *Streamer*, 4 - Navio Sísmico.
 (b) Levantamento sísmico terrestre. 1 - Cami-
 vibrator, 2- Representação de uma reflexão da
 estrutura geológica, 3 - *Streamer*, 4 - Navio Sí-
 smico.

Figura 2.1: Representação da aquisição sísmica marítima (a) e terrestre (b).
 Adaptado de: [2], [3]

Uma aquisição sísmica marítima emprega-se uma configuração específica de vários receptores (hidrofonos) montados em um cabo receptor denominado *Streamer*. A configuração varia de acordo com o objetivo do levantamento: pode ser um único cabo (para aquisição 2D, figura 2.2(A)), vários cabos paralelos (para aquisição 3D, figura 2.2(B) e 4D) ou receptores no fundo do mar (para aquisição 4D) etc. Ao longo dos cabos são instalados niveladores de profundidade, conhecidos como pássaros e ao final do cabo é instalada uma boia de cauda equipada com equipamentos como farol, radar e um *transponder*.

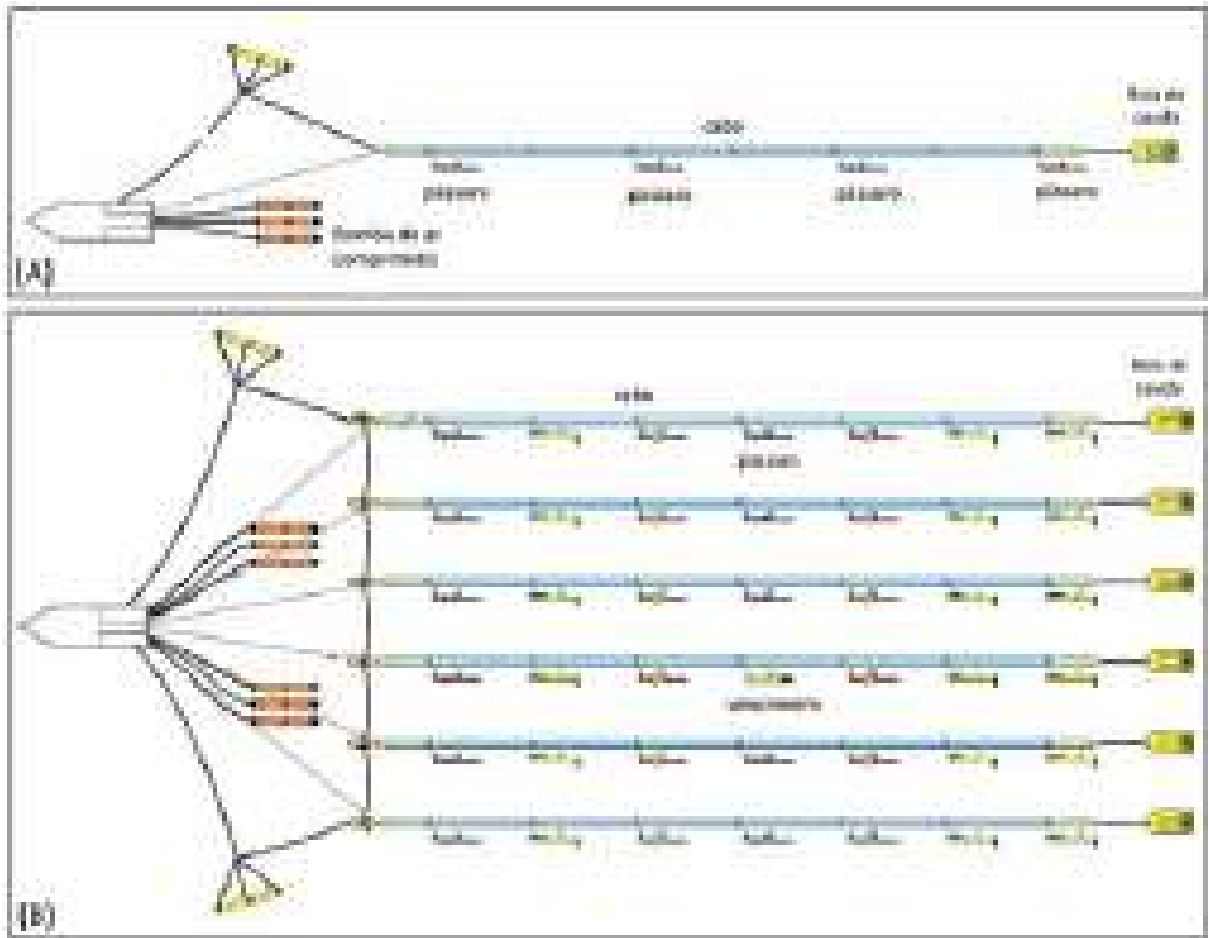


Figura 2.2: Equipamento geral utilizado em aquisições sísmicas marítimas (A) 2D e (B) 3D

Adaptado de: [1]

A figura 2.3 apresenta disparos de uma fonte e os sinais sísmicos gerados registrados por um conjunto de receptores conectados a um cabo que, por sua vez, é rebocado por um navio. Cada registro captado por um receptor representa um traço sísmico. Denomina-se de tiro um grupo específico de traços sísmicos originados do mesmo disparo da fonte, mas registrados em diferentes receptores. O número de traços envolvidos em uma coleta de tiros é igual ao número total de canais de gravação e a distância entre cada canal de gravação, já a localização correspondente do disparo da fonte é denominada *offset*. Ao final de uma aquisição sísmica, o conjunto de tiros gerados são agrupados para criação de um sismograma.

É possível observar, por conseguinte, que a figura 2.4 apresenta o trajetos de raio de seis (6) disparos sucessivos e a exibição esquemática desses disparos gravados em seis (6) receptores 2.4(C). Nota-se que em 2.4(B) são gerados seis (6) tiros contendo seis (6) traços. Em 2.4(A) os seis (6) tiros são agrupados em um sismograma.



Figura 2.3: Cabo rebocado por um navio registrando sinais sísmicos gerados por uma fonte de ar comprimido
Adaptado de: [1]

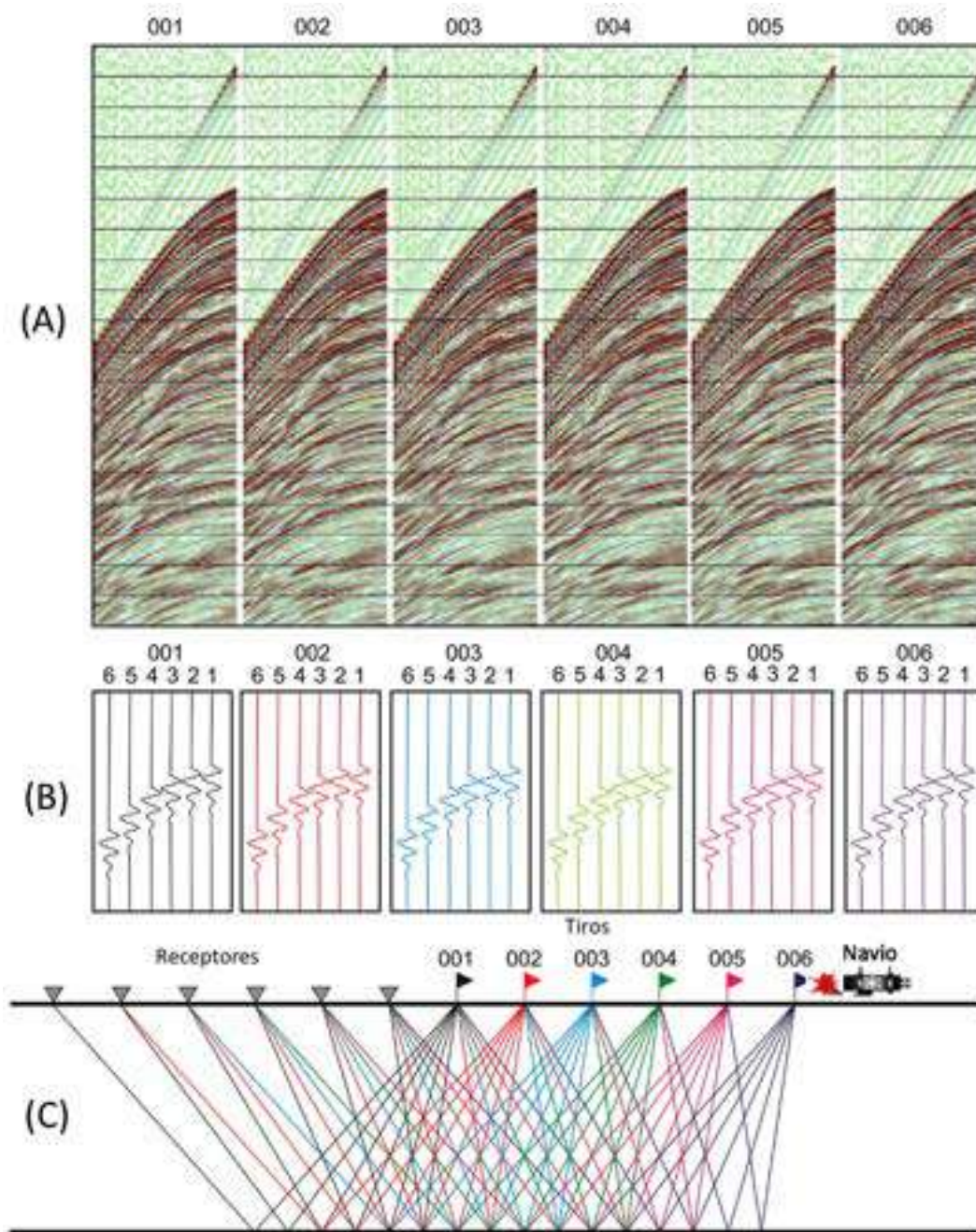


Figura 2.4: Trajetos de raio de disparos e exibição esquemática gravados em seis (6) receptores
Adaptado de: [1]

2.2 Ruídos na sismica marinha

Em um evento sísmico podem ser encontrados reflexões ou ruídos. As reflexões nos tiros gravados são reconhecidas pelo tempo de viagem hiperbólico [5]. Já o ruído pode ser definido como todo tipo de evento ou amplitude aparecendo nos dados sísmicos que não sejam reflexões genuínas. Às vezes, o ruído gravado é gerado pelos componentes do sistema sísmico, por exemplo: uma bolha gerada pela bomba de ar comprimido; ruído de pássaros. Outras vezes os ruídos surgem devido às especificações do ambiente de trabalho, como mamíferos marinhos, hélices, tráfego marítimo. O primeiro tipo de ruído é denominado ruído gerado pela fonte, enquanto o segundo tipo é conhecido como ruído ambiente. O ruído nos dados sísmicos pode ser classificado como ruído coerente, que tem consistência por rastreamento ou ruído aleatório, que não tem uma consistência sistemática de um traço para outro (Figura 2.5). A maioria dos componentes de ruído coerentes é gerada pelos componentes de aquisição; por exemplo, o ruído operacional é produzido pela embarcação de pesquisa, assim como o ruído dos pássaros surge dos movimentos dos niveladores de profundidade montados em intervalos regulares ao longo do cabo.

Os ruídos marinhos encontrados no sinal sísmico podem ser caracterizados da seguinte maneira:

- O ruído operacional é originado diretamente do movimento hidrodinâmico do navio sísmico, seus motores, maquinaria e hélices. Suas características como forma, amplitude e frequência dependem principalmente das dimensões, velocidade de reboque e forma de construção do navio.
- Ondas guiadas (*guided waves*) são múltiplas ondas refletidas que viajam em camadas separadas rodeadas por contrastes de alta impedância e, portanto, caracterizam-se como um fenômeno comum de propagação de ondas em meios estratificados [24]. Segundo Sheriff e colaboradores [25], as ondas guiadas são ondas próximas da superfície que se encontram presas. A natureza dispersiva destas ondas faz delas fáceis de serem reconhecidas em tiros gravados. Ondas guiadas também se constituem em chegadas antecipadas [5].
- Ruído difuso (*side-scattered noise*) comumente ocorre no fundo da água, quando não é plano: topografia suave. Irregularidades de variação de tamanho atuam como pontos de dispersão [5].
- Ruído do cabo (*cable noise*) é linear e baixo em amplitude em frequência. Este, primariamente, aparece em tiros gravados como chegadas tardias [5].
- Onda aérea é uma onda acústica que se propaga no do ar, a 330 m/s aproximadamente, e que pode ser gerada e registrada durante um levantamento sísmico.

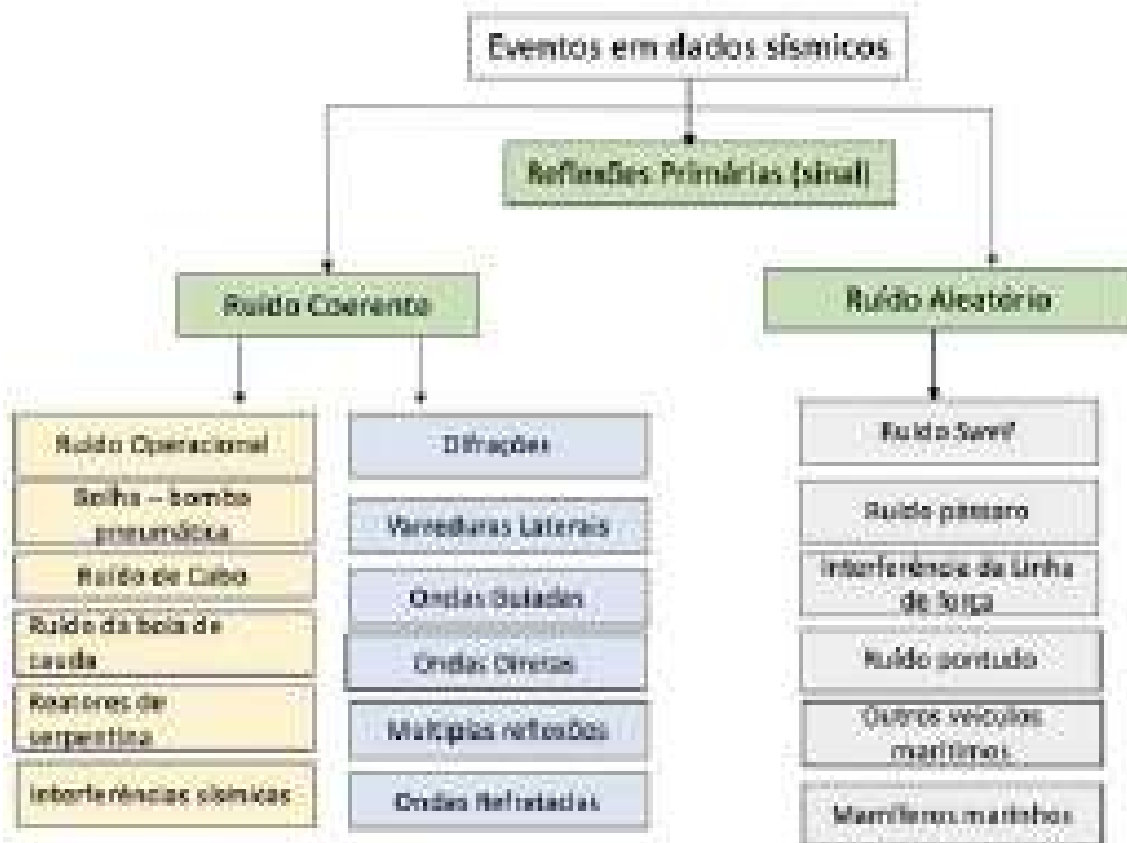


Figura 2.5: Diferentes tipos de ruído coerentes e aleatórios observados especificamente em dados sísmicos marinhos. Nas caixas amarelas os tipos de ruídos coerentes abrangem o ruído gerado por componentes técnicos do sistema de pesquisa, enquanto as caixas azuis indicam o ruído do ambiente geológico. Adaptado de: [1]

As ondas aéreas são um tipo de ruído coerente [26].

- Ruído aleatório é oriundo de várias fontes, no caso de aquisição marinha, o movimento da onda na água provoca a vibração do cabo, enquanto o ruído elétrico do instrumento de gravação pode causar ruído ambiental [5]).
- Ruído de ondas (*swell noise*) é o tipo de ruído mais dominante entre dados sísmicos marinhos brutos, sendo gerado pelo movimento das ondas do mar. Quanto à aquisição sísmica, quando realizada em condição de mar revolto, o ruído torna-se mais evidente. As principais características do ruído de ondulação são sua grande amplitude e seu conteúdo de baixa frequência, geralmente variando de dois (2) até dez (10) Hz, podendo chegar até 15Hz [27]. É possível que ocorra indução de atrasos ou suspensão temporária de aquisição de dados em pesquisas marítimas [28].
- Ruído bolha (*bubble noise*) é ocasionado pelos disparos da bomba de ar comprimido. Quando esta é disparada a uma certa profundidade na coluna d'água, descarrega alta pressão de sua câmara de ar para a água a fim de gerar o sinal sísmico primário. Neste contexto, ocorrem (I) interações entre a pressão interna da bolha de ar gerada por essa descarga repentina de ar; (II) o resultado da pressão ambiente em expansão periódica e (III) o colapso da bolha de ar que sobe em direção à superfície do mar. Cada uma dessas três expansões, assim como o colapso individual produz um novo sinal com polaridade oposta até atingir a superfície do mar, denominada ruído de bolhas.
- Ruído de pássaro (*bird noise*) são dispositivos conhecidos como pássaros montados na flâmula em intervalos regulares para manter a profundidade (para pesquisas 2D) e posições laterais (para pesquisas em 3D) da serpentina. Os ângulos dos pássaros são modificados automaticamente pelo *software* de controle sempre que a profundidade do *streamer* muda durante a aquisição, para manter a serpentina no nível de profundidade desejado. O movimento das barbatanas e o efeito da turbulência da água ao redor dos dispositivos durante a gravação pode induzir ruído de baixa frequência (aproximadamente trinta (30) Hz).
- Múltiplas reflexões ocorrem quando o sinal sísmico viaja pelo mesmo caminho do raio mais do que uma vez e eles produzem chegadas de reflexão que se repetem nos registros de tiro. O fundo do mar é um bom refletor e possui um coeficiente de reflexão relativamente alto, refletindo de volta uma parte importante da entrada de energia sísmica em direção à superfície do mar registrada.

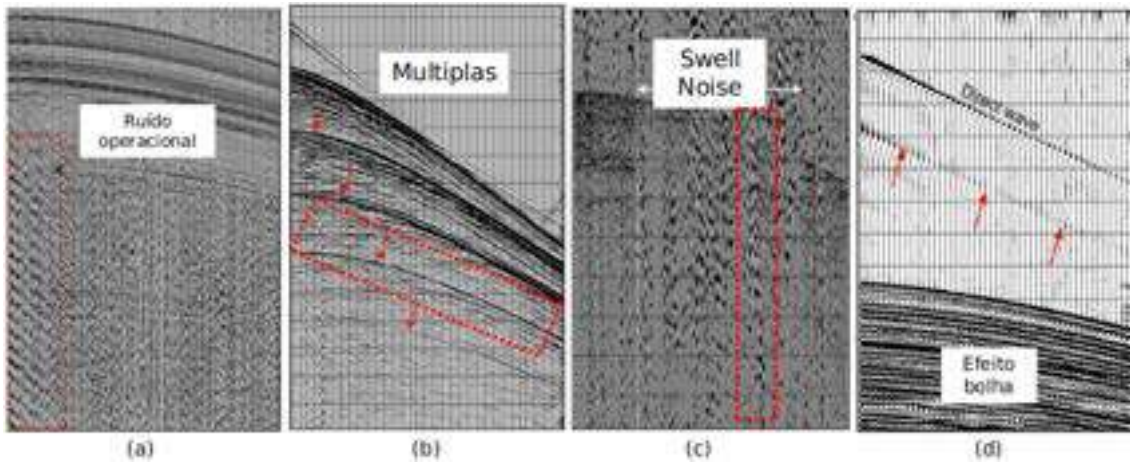


Figura 2.6: Ruídos coerentes. (a) Ruído operacional, causado pela movimento do navio (b) Efeito múltiplas. (c) Ruído *swell*. (d) Efeito bolha, causado pela bomba de ar comprimido (*airgun*). Adaptado de: [1]

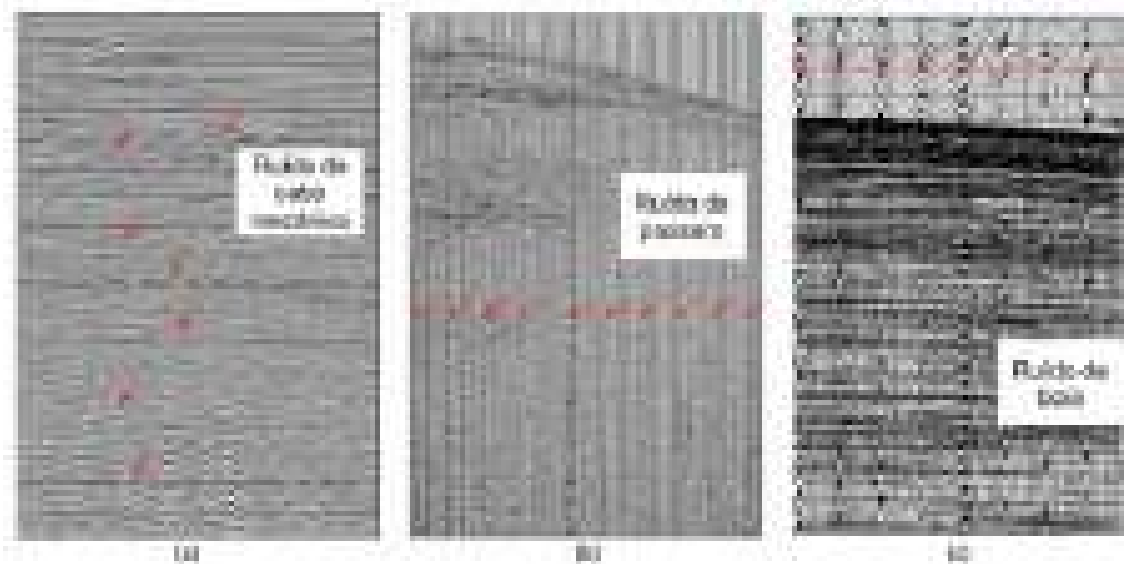


Figura 2.7: (a) Ruído de cabo mecânico (coerente). (b) Ruído de pássaro. (c) Ruído de boia (coerente). Adaptado de: [1]

2.3 Métodos de atenuação de ruído

Alguns exemplos clássicos de atenuação de ruído podem ser encontrados em Yilmaz [5] como filtragem passa-banda e F-K nos quais os dados sísmicos são transformados para domínio de *Fourier* e atenuam porções indesejáveis do sinal, ou seja, ruído, com base na suposição de que o sinal e o ruído sejam separados nesse novo domínio. Alguns exemplos clássicos de atenuação de ruído podem ser encontrados como deconvolução f-x [29], previsão de espaço de frequência ou f-x e previsão de espaço-tempo ou t-x [30], filtragem de *Cadzow* [31] ou a análise de espectro singular [32]. Em outros estudos encontramos métodos de atenuação de ruídos mais recentes. Bonar e Sacchi [33] apresentam uma técnica para atenuação e reconstrução de sinais com o algoritmo de médias não locais. Oropéza e Sacchi [32] apresenta uma análise de espectro como uma ferramenta para atenuação de ruídos e reconstrução simultânea de dados sísmicos. Por sua vez, Tian e colaboradores [34] utilizam a técnica de filtragem de pico de frequência e tempo com uma melhoria chamada traço hiperbólico de excentricidade variável, este método leva a correlação espacial entre os canais adjacentes em consideração, e melhora a linearidade dos sinais efetivos através de um processo de reamostragem. Outras técnicas em pauta na literatura são as com promoção de esparsidade. De acordo com Zhu e colaboradores [35], técnicas com o intuito de promover a esparsidade dos dados sísmicos a partir da transformação para outros domínios, como transformada *wavelet* [36], *S-transform* [37], transformada *Curvelet* [38], [39], transformada Radon [40], [41] têm a vantagem de possibilitar a eliminação de ruídos mesmo que ocupem a mesma faixa de frequência do sinal. Para o tratamento de ruídos do tipo *swell* os métodos convencionais utilizam recursos de sinal para atenuar o ruído sísmico como número de onda, frequência e transformação de domínio. Uma abordagem direta é aplicar a filtragem de f-x para a faixa de frequência na qual o ruído está presente e substituir os traços ruidosos com suas versões filtradas por f-x para essas frequências. No entanto, como a amplitude do ruído pode ser alta em comparação com o sinal, os resíduos, muitas vezes, ainda são inaceitáveis. Schonewille e colaboradores [42] introduzem uma melhoria aplicando a filtragem f-x iterativamente. Bekara e Nan Der Baan [43] desenvolveram um método automático para explorar diferenças nas propriedades estatísticas entre ruído *swell* e as amplitudes do sinal a fim de estabelecer um critério de detecção. Por fim, Sternfels e colaboradores [44] modelam o sinal coerente via sua matriz de trajetória de baixa patente e o ruído irregular como um componente dos dados de entrada. Com isso, seu método pode efetivamente separar o sinal e o ruído *swell* na entrada corrompida. Algumas abordagens de atenuação de ruídos referem-se ao uso de técnicas baseadas em predição, empregando modelos de aprendizado de máquina treinados em diferentes conjuntos de dados. Chen e colaboradores [45] propõem um

framework híbrido baseado em uma aproximação analítica e na aproximação baseada em aprendizado, com aplicações em seções sísmicas 2-D. Chen [46] apresenta a construção da representação dos dados utilizando o algoritmo sequencial *generalized K-means (SGK)* para aplicações em dados 2-D e 3-D. Já Siahisar e colaboradores [47] realizaram uma comparação de métodos de representação esparsa dos dados conhecido como *dictionary learning* com outros métodos clássicos de atenuação de ruído como *wavelet*, *curvelet* ATV e NTV, utilizando dados sísmicos 2-D. O autor também apresenta aplicações de codificação esparsa para atenuação de ruído aleatório e ainda para atenuação de ruído de múltiplas no dado sísmico corrigido pela técnica de correção normal-move-out (correção NMO).

2.4 Filtragem

O objetivo do processamento sísmico é o tratamento dos dados obtidos num levantamento. Segundo Yilmaz [5], os três estágios principais no processamento de dados sísmicos são deconvolução, empilhamento e migração (figura 1.1), no qual cada etapa visa melhorar a resolução sísmica. O termo resolução no contexto de sísmica representa a capacidade de distinguir recursos separados; a distância mínima entre dois recursos para que os dois possam ser definidos separadamente e não como um. Esta resolução se aplica tanto no sentido vertical como horizontal de um objeto, que pode ser interpretado a partir de dados sísmicos.

Como exemplo, este estudo apresenta a Figura 2.8 que representa o volume de dados sísmicos nas coordenadas de processamento - ponto médio, deslocamento e tempo. A deconvolução atua nos dados ao longo do eixo do tempo e aumenta a resolução temporal. Empilhamento comprime o volume de dados na direção de deslocamento e produz o plano da seção empilhada (a face frontal do prisma). A migração move os eventos de imersão para suas verdadeiras posições de subsuperfície e reduz as difrações e, assim, aumenta a resolução lateral.

O processamento de sinais em geofísica é geralmente alcançado por meio da realização de vários processos matemáticos aplicados ao sinal sísmico, que, geralmente, apresenta-se na forma de séries temporais. Considerando o ambiente no qual o sinal é observado, é possível afirmar que os dados tornam-se uma função de tempo ou uma função de espaço. Isso significa dizer que os dados são definidos nos domínios de tempo ou de distância, respectivamente. Sendo assim, os dados podem ser transformados a partir do domínio do tempo ou da distância por análise espectral. Um traço sísmico pode ser representado por uma série de *Fourier*, desta forma, estes sinais são compostos pelo somatório de vários senoides com diferentes fases, amplitude e frequência características. Com isso, é possível decompor os traços em seus diferentes componentes sinusoidais de diferentes frequências usando análise espectral, e



Figura 2.8: Volume de dados sísmicos

remover os componentes de amplitude indesejados de uma banda de frequência especificada com a aplicação de um operador de filtro adequado [1]. Vale ressaltar que o processo matemático que decompõe um sinal no domínio do tempo em componentes de frequência é a Transformada de *Fourier*. Com este procedimento, é possível analisar conteúdos de frequência da série temporal representada por uma série de valores de amplitude em função do tempo no domínio do tempo como traços sísmicos. Os traços sísmicos normalmente consistem em bilhões de amostras de tempo discretas em milhões de traços, que tornam computacionalmente cara a aplicação da transformada de *Fourier* para grandes volumes de dados como conjuntos de dados sísmicos, contudo, a transformação discreta de *Fourier* pode ser calculada de forma mais rápida através de um algoritmo específico conhecido como transformada rápida de *Fourier* (FFT) [48].

2.4.1 Filtragem no tempo

O procedimento comumente utilizado para realização do atenuação de ruídos em sinais sísmicos é a filtragem de frequência. A sua principal motivação diz respeito ao fato de que as amplitudes de sinal e ruído aparecem em diferentes componentes de frequência dos dados sísmicos. Se as frequências de ruído estiverem próximas às frequências do sinal, os sinais de reflexão também podem ser removidos junto com o ruído por filtragem de frequência. Os filtros de frequência são normalmente classificados de acordo com as características de suas regiões de banda passante. Da literatura neste campo, é possível identificar projeto de filtros diferentes para modificar adequadamente o espectro de amplitude da entrada. A saber:

- Os filtros passa-baixa (Figura 2.9A) removem as frequências superiores a um limite de valor de frequência especificado. Interruptor de baixa frequência (f_1) é zero.
- Os filtros passa-alta (Figura 2.9B) removem a banda de frequência inferior a um limite especificado do valor de frequência (f_1). Corte de alta frequência (f_2) é a frequência de *Nyquist*.
- Os filtros passa-banda (Figura 2.9C) mantêm a faixa de frequência especificada entre uma faixa e um valor de corte de alta frequência (f_1 e f_2 , respectivamente).
- Os filtros de entalhe (Figura 2.9D) removem apenas um valor de frequência especificado ou um valor muito estreito de banda de frequência, normalmente, usado para filtrar componentes específicos de ruído de monofrequência, como interferência na linha de energia de cinquenta (50) ou sessenta (60) Hz.

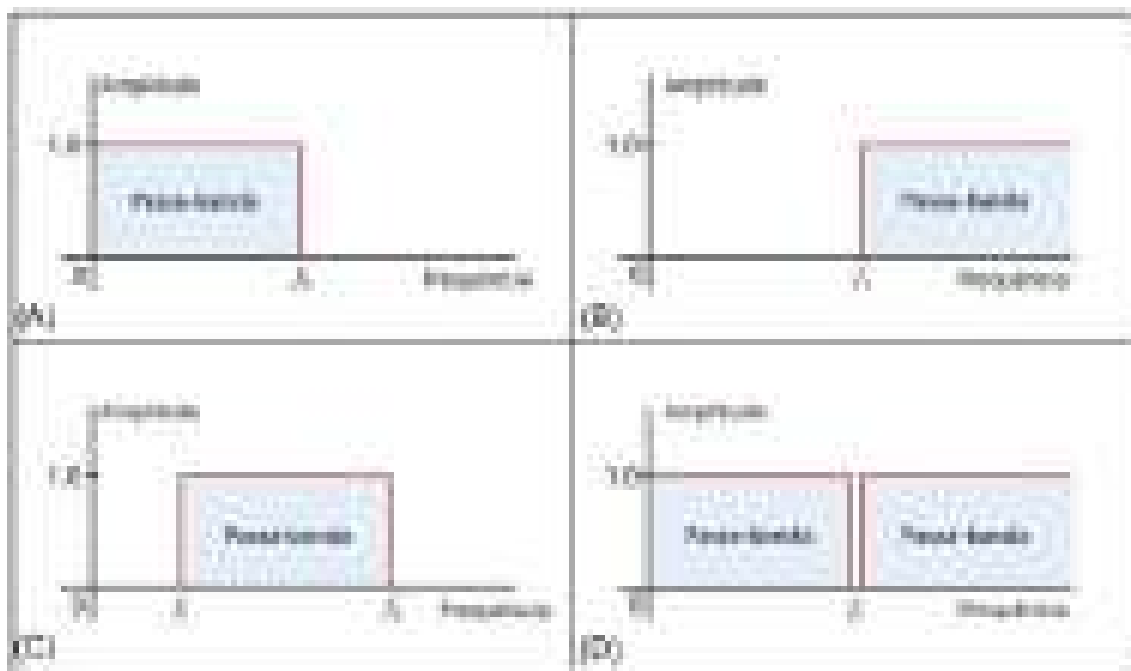


Figura 2.9: Ilustração esquemática de passa-banda com diferentes projetos de filtro no domínio da frequência. (A) passa-baixa, (B) passa-alta, (C) passa-banda e (D) filtros de entalhe. f_1 e f_2 representam frequências de corte
Adaptado de: [1]

Segundo Dondurur [1], os sinais no domínio do tempo com banda limitada são utilizadas, na prática, para filtrar os dados sísmicos. Estes são conhecidos como operadores de filtro no domínio do tempo e da amplitude. Amostras deste operador são denominadas coeficientes de filtro. Vale ressaltar que este filtro é esquematicamente definido como uma janela retangular no domínio da frequência (figura 2.10A) e o seu objetivo principal é manter as amplitudes dentro desta janela sem distorção de

amplitude, enquanto suprime as amplitudes que podem ser encontradas fora dessa faixa de frequência. Sua equação pode ser definida da seguinte maneira:

$$A(f) = \begin{cases} 1 & f_1 < f < f_2 \\ 0 & \text{outros} \end{cases} \quad (2.1)$$

onde f representa a frequência, enquanto f_1 e f_2 são as frequências de corte. Sendo assim, este é um espectro ideal capaz de ser utilizado para filtrar dados sísmicos quando, simplesmente, forem multiplicados pelo espectro de dados do pesquisador, pois nunca afetará as amplitudes na banda passante e, ainda, removerá, completamente, as amplitudes fora da banda passante.

A sua função é representada por (figura 2.10B):

$$A(r) = \text{sinc}(t) = \frac{\text{seno}(t)}{t} \quad (2.2)$$



Figura 2.10: (A) Um espectro de amplitude em forma de vagão no domínio da frequência e (B) sua contraparte no domínio do tempo, a função Sinc.

Adaptado de: [1]

Por conseguinte, utilizando o operador de filtro representado pela figura 2.10, a filtragem dos dados sísmicos pode ser feita no domínio do tempo ou da frequência. No domínio da frequência, os espectros de amplitude do traço sísmico de entrada e o operador do filtro são multiplicados (Figura 2.11A), em contrapartida, no domínio do tempo, um processo de convolução [49] é realizado e os coeficientes do operador de filtro são convoluídos por amplitudes de traços sísmicos (Figura 2.11B). Ambas as aplicações produzem os mesmos resultados, contudo, a aplicação no domínio do tempo possui custo computacional mais baixo, uma vez que processo de convolução é mais econômico do que computando a transformação de *Fourier* direta e inversa, antes e depois da filtragem [5].

É preciso considerar que o controle de qualidade do resultado de uma filtragem de frequência, geralmente, é feito por meio da análise visual dos tiros filtrados e por uma comparação visual dos espectros de amplitude dos tiros brutos e filtrados.

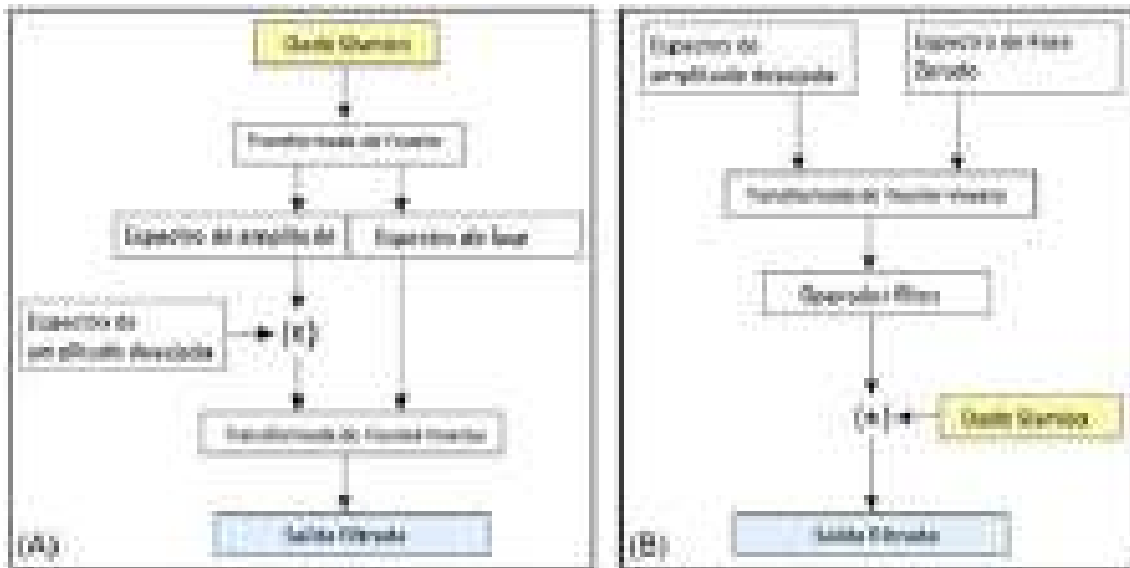


Figura 2.11: Fluxo de trabalho para o processo de filtragem de passagem de banda nos domínios (A) de frequência e (B) de tempo. (C) A filtragem é realizada utilizando a multiplicação (\times) dos espectros de amplitude do operador sísmico e de filtro bruto no domínio da frequência (painel superior) ou por uma convolução ($*$) dos próprios dados sísmicos com o operador de filtro no domínio do tempo (painel inferior)

Adaptado de: [1]

Segundo Yilmaz [5], uma aplicação de um filtro passa-banda bem-sucedida deve remover a maioria das amplitudes de ruído indesejadas dos componentes, tanto de alta quando de baixa frequência.

Filtro passa-banda do tipo *Butterworth*

O filtro passa-banda do tipo *Butterworth* é esquematicamente definido como uma janela retangular no domínio da frequência e o seu objetivo principal é manter as amplitudes dentro desta janela sem distorção de amplitude, enquanto suprime as amplitudes fora dessa faixa de frequência. Conforme Dondurur [1] e Yilmaz [5], um projeto de filtro passa-banda pode ser obtido executando as etapas representadas na figura 2.12. Primeiramente, é necessário definir as frequências de corte no domínio da frequência e, em seguida, aplicar a transformação inversa de *Fourier* para obter o operador no domínio do tempo (figura 2.12A). No domínio do tempo, é preciso truncar o operador nas duas extremidades uma vez que suas amplitudes possuem intervalos de tempo de mais e menos infinito. Este corte resulta em um espectro de amplitude com fortes ondulações no domínio da frequência e este fenômeno é conhecido como efeito *Gibbs* [50]. No domínio da frequência é preciso definir as inclinações de corte, o que resulta em uma região de banda passante em forma de trapézio (figura 2.12B). A última etapa é suavizar as bordas do trapézio, a fim de

rejeitar as bandas nos lados de alta e baixa frequência no domínio da frequência. Isso fornecerá um operador de filtro mais restrito no domínio do tempo. Com a transformação inversa de *Fourier* torna-se possível obter o operador de filtro final no domínio do tempo (figura 2.12C).

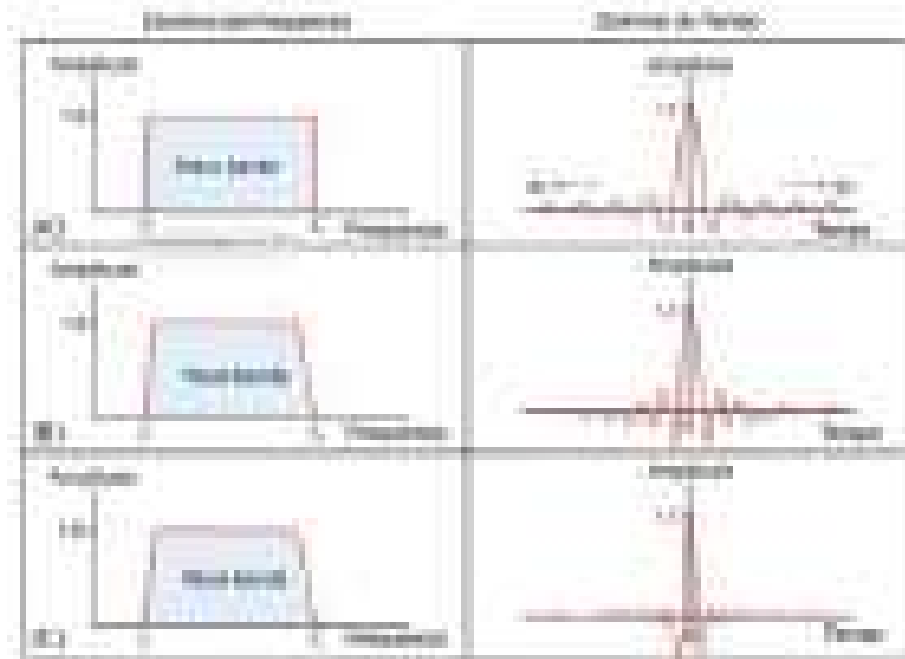


Figura 2.12: Representação esquemática de um projeto de operador de filtro passa-banda.

O filtro passa-banda do tipo *Butterworth* requer duas frequências para valores de corte de baixa e alta frequência e dois valores de inclinação para a definição da banda de transição do trapézio (Figura 2.13 C). O espectro do operador de filtro mostrado na Figura 2.13 A, ainda, é multiplicado pelo espectro do dado bruto no domínio da frequência, ou suas representações no domínio do tempo do operador podem ser convoluídas com os dados brutos de entrada no intuito de obter o resultado do filtro passa-banda.

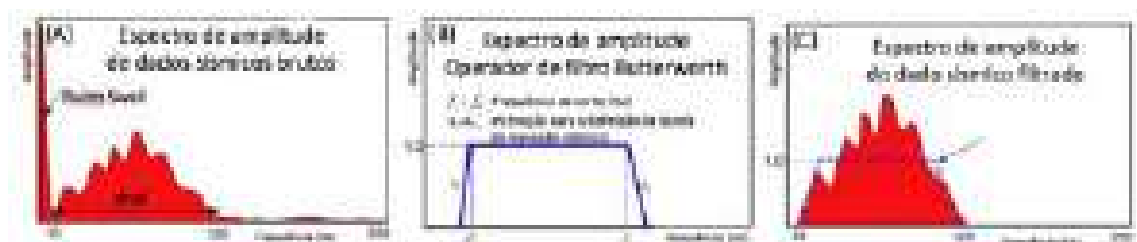


Figura 2.13: Ilustração esquemática da modificação do espectro de amplitude no domínio da frequência. (A) Um esquema espectro de amplitude de um tiro bruto. (B) Filtro *Butterworth*. (C) Espectro de amplitude esquemático do tiro filtrado

2.4.2 Filtragem no tempo e espaço

A filtragem por imersão utiliza uma transformada 2D de *Fourier* [51] para remover tipos de ruído correlacionados que propagam-se em velocidades diferentes. Em uma transformação 1D *Fourier*, o sinal de entrada é apenas uma função do tempo. No entanto, na prática, os dados sísmicos dependem do tempo e do espaço (deslocamento ou ponto médio comum). Sendo assim é possível aplicar uma transformada *Fourier* 2D aos dados sísmicos. O domínio correspondente da transformada de *Fourier* da dimensão de tempo é a frequência e para a dimensão de espaço, o domínio correspondente é o número de onda. O sinal observado no domínio do tempo-distância (t-x) é transferido para o domínio da frequência-número de onda (F-K), portanto, o espectro obtido também é conhecido como espectro F-K.

O espectro F-K é uma ferramenta utilizada para discriminar os eventos correlatos com diferentes inclinações presentes nos dados sísmicos no domínio do tempo. Este espectro é composto por dois painéis: um é negativo e o outro é um painel positivo, representados por números de onda negativos e positivos em relação à respectiva frequência. Se a inclinação estiver próxima do *offset* (ou *offset* curto), isso significa que o evento tem uma queda positiva e suas amplitudes surgem no painel positivo do número de onda do espectro F-K. Em contrapartida, se a imersão estiver longe do *offset* (ou *offset* longo), ocorrerá a situação oposta, pois, aparecerá no painel negativo (figura 2.15). À medida que a inclinação de um evento aumenta no domínio do tempo, as amplitudes desse evento no domínio da frequência se aproximam do eixo horizontal do número de onda. Na figura 2.14(A) são apresentados os ruídos no domínio do tempo e na figura 2.14(B) os mesmos ruídos são apresentados no domínio da frequência. Nota-se, assim, no domínio da frequência, as reflexões primárias (R) em forma de leque no painel positivo ao eixo de frequência no espectro F-K; o ruído de pássaro horizontal (B) em ambos os painéis; o ruído linear da boia de cauda (T) no painel negativo no espectro F-K e, por fim, a onda direta, indicada por D, que aparece como um evento linear no painel positivo.

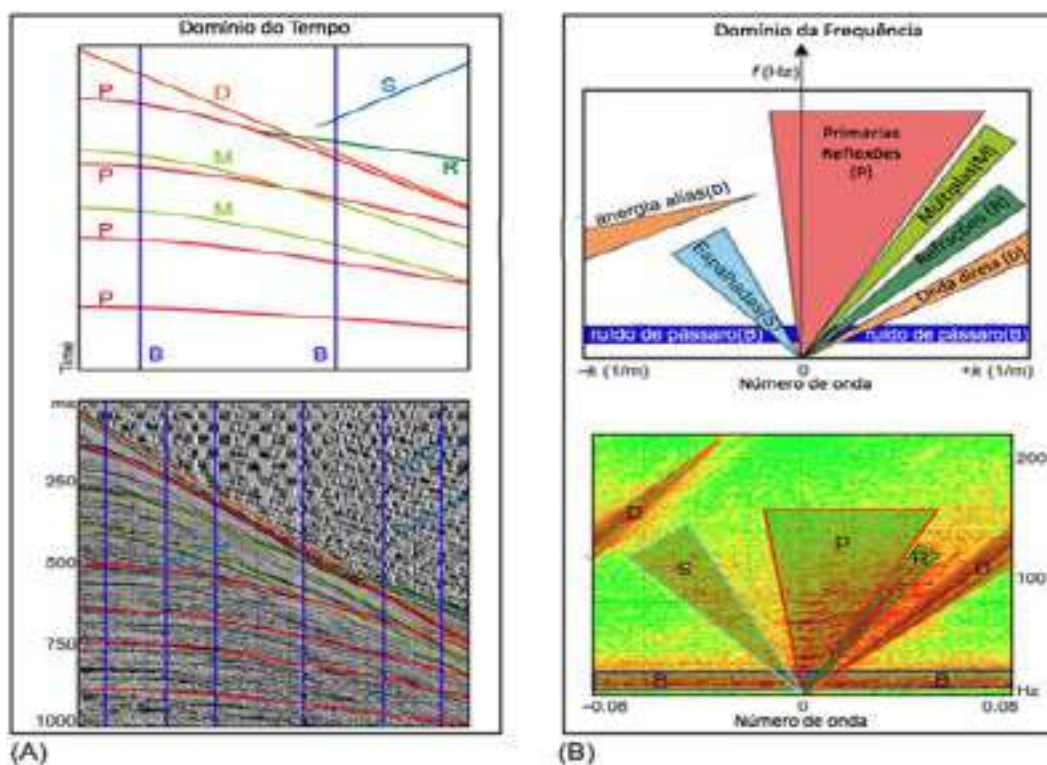


Figura 2.14: (A) Ilustração esquemática dos eventos comuns em um tiro marinho (parte superior) e exibições correspondentes dos mesmos eventos em um tiro marinho real (parte inferior). (B) Os espectros F-K esquemáticos (parte superior) e reais (parte inferior) do exemplo captado. (P), reflexão primária; (R), onda refratada; (D), onda direta; (B), ruído de pássaro; (M), reflexões múltiplas; (S), eventos de imersão negativos (dispersos)

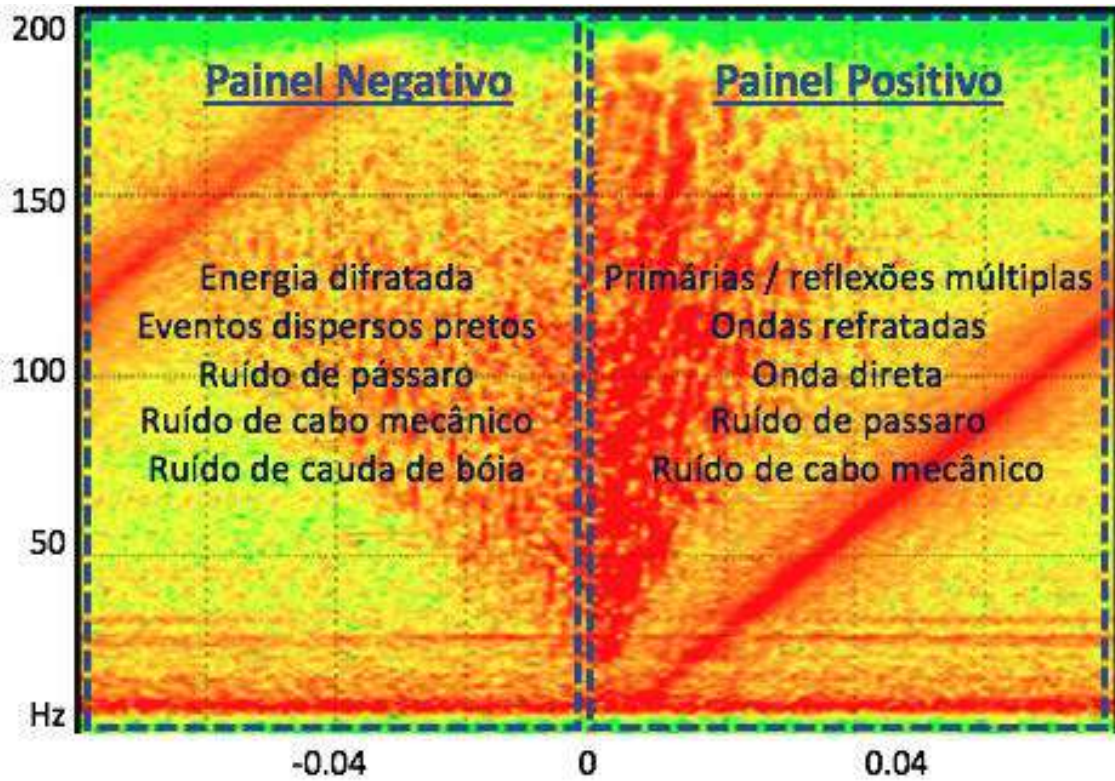


Figura 2.15: Painéis negativos e positivos de um espectro F-K e aparência de domínio F-K de eventos comuns encontrados em tiro marinho. Adaptado de: [1]

Em uma aplicação da filtragem F-K, é preciso manter as amplitudes das reflexões primárias e descartar o restante dos eventos. Para isso, é possível estabelecer fronteiras entre as regiões de interesse e descarte em um espectro F-K através do desenho de um polígono. Este pode desempenhar as funções de proteção ou remoção de uma determinada área para discriminar as zonas de sinal e ruído no domínio F-K (Figura 2.16).

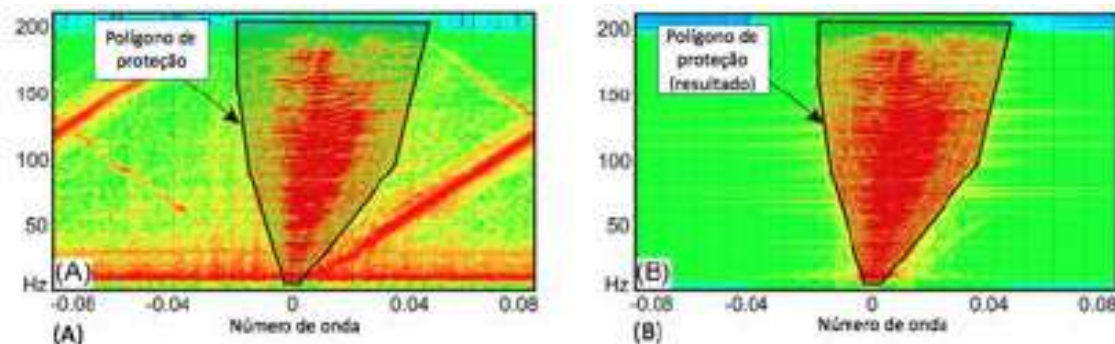


Figura 2.16: Polígono definido para discriminar as zonas de sinal e ruído no domínio F-K. (A) Polígono de proteção (B) a saída do filtro F-K

A identificação e separação bem-sucedida de sinais de reflexão verdadeiros e ruídos indesejados é um problema que vem sendo estudado no processamento de dados

sísmicos. Na literatura, é possível verificar que tal processo afeta consideravelmente a fidelidade das imagens sísmicas subsequentes ([52];[53]) e inversão geofísica, como inversão de amplitude-variação-com-deslocamento ([54];[55]), inversão de formas de onda ([56];[57];[58]) e interpretação geológica ([59]). Os dados sísmicos são inevitavelmente afetados por diferentes tipos de ruídos, em situações como a existência de ruído nos dados sísmicos pré-empilhamento. Desse modo, as informações de amplitude são afetadas causando resultados de inversão não confiáveis. Já para dados sísmicos pós-empilhamento, a existência de ruído afeta a capacidade de interpretação, que vincula diretamente a modelagem de reservatórios subterrâneos.

2.5 *Muting*

O *Muting* é usado para remover as amplitudes de algumas áreas barulhentas nos dados sísmicos. A diferença da eliminação de traços encontra-se nas zonas de bloqueio que podem, não necessariamente, compor toda a amostra de um traço sísmico específico. Essas áreas específicas são as zonas de ruído coerente, como as amplitudes na coluna de água (figura 2.17), ondas diretas, refrações ou áreas com taxa de sinal ruído, anormalmente, baixa no tiro. Isso significa afirmar que as amplitudes nessas áreas indesejadas são, simplesmente, multiplicadas por zero. No pré-processamento deste estudo foi executado *muting* na coluna de água.

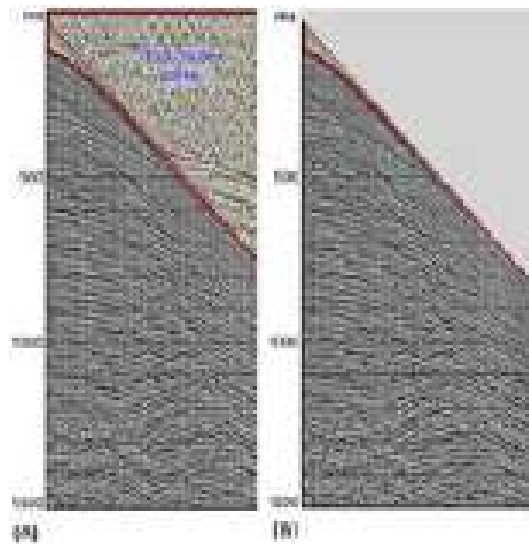


Figura 2.17: (A) Exemplo de tiro sísmico sem aplicação de *mute* no topo. (B) Exemplo de tiro sísmico com aplicação de *mute* no topo

2.6 Resumo do capítulo

Neste capítulo foi apresentada uma introdução geral à sísmica na qual contemplou-se conceitos sobre levantamento sísmico, os tipos de ruídos marinhos que podem ser gerados a partir de uma aquisição sísmica marinha e seu impacto na incerteza das imagens sísmicas dos processos subsequentes, inversão geofísica e consequentemente interpretação geológica. Também foram apresentados conceitos sobre como esses ruídos são filtrados em dados sísmicos pré-empilhamento. O próximo capítulo abordará conceito de redes neurais profundas e contemplará, ainda, uma revisão bibliográfica da aplicação dessas redes para atenuação de ruído.

Capítulo 3

Revisão Bibliográfica

Neste capítulo será introduzido conceito de aprendizado profundo com enfoque em redes autocodificadores convolucionais e suas aplicações para atenuação de ruídos em dados sísmicos pré-empilhamento.

3.1 Redes Neurais Profundas

O aprendizado profundo permite que modelos de redes neurais compostos por várias camadas de processamento aprendam representações de dados com vários níveis de abstração. Esses métodos atualizaram o estado da arte em reconhecimento de fala, classificação de imagens, detecção de objetos, assim como outros domínios. Uma arquitetura de aprendizado profundo é composta por uma pilha multicamada de módulos sujeitos a aprendizado, sendo que muitos são capazes de calcular mapeamentos não lineares de entrada e saída. Com a composição de várias camadas não lineares, um sistema pode implementar funções complexas de suas entradas. O aprendizado supervisionado é um tipo de aprendizado de máquina no qual, durante o seu treinamento, o algoritmo recebe uma entrada e produz uma saída na forma de um vetor de pontuações. A função objetivo mede o erro (ou distância) entre as pontuações de saída e o padrão desejado de pontuações. Em seguida, o algoritmo modifica seus parâmetros internos ajustáveis para reduzir esse erro. Esses parâmetros ajustáveis, chamados de pesos, são números reais e definem a função de entrada e de saída do modelo. Numa rede neural profunda pode haver centenas de milhões desses pesos ajustáveis e centenas de milhões de exemplos rotulados com os quais treinar a máquina. Para ajustar o vetor de peso, o algoritmo de aprendizado calcula um vetor de gradiente do procedimento de retropropagação [17], utilizando-se o cálculo da derivada (gradiente) de uma função objetivo em relação aos pesos de uma pilha de módulos de várias camadas. A derivada da função objetivo é calculada em relação aos parâmetros do modelo e a equação de retropropagação é aplicada repetidamente para propagar gradientes por todas as camadas, começando pela saída

no topo (onde a rede produz sua previsão) até o fundo (onde a entrada externa é alimentada), a fim de reajustar todos os pesos em relação ao erro da função objetivo.

3.1.1 Redes Neurais Convolucionais (CNN)

As redes convolucionais [14] são projetadas para processar dados na forma de tensores. Uma imagem colorida, por exemplo, é composta por três matrizes 2D, contendo intensidades de pixel nos três canais de cores. Há também modalidades de dados na forma de múltiplas matrizes: 1D para sinais e sequências, incluindo idioma; 3D para imagens ou espectrogramas de áudio e 4D para imagens de vídeo ou volumétricas.

Uma arquitetura convolucional típica é uma estruturada como uma série de camadas. As primeiras etapas são compostas por dois tipos de camadas e função de ativação: camadas convolucionais e camadas de *pooling*. Uma camada convolucional é composta por um conjunto de núcleos convolucionais, no qual cada neurônio atua como um núcleo. O núcleo convolucional funciona dividindo a imagem em pequenas fatias, comumente conhecidas como campos receptivos. A divisão de uma imagem em pequenos blocos ajuda a extrair mapas de recursos. O núcleo convolve com as imagens usando um conjunto específico de pesos chamados bancos de filtros, multiplicando seus elementos com os elementos correspondentes do campo receptivo (figura 3.2). Em seguida, uma função não linear, por exemplo a função *ReLU*, é aplicada ao resultado anterior dessa soma ponderada local. Todas as unidades em um mapa de recursos compartilham o mesmo banco de filtros. Devido à capacidade de compartilhamento de peso da operação convolucional, diferentes conjuntos de recursos dentro de uma imagem podem ser extraídos deslizando o núcleo com o mesmo conjunto de pesos na imagem [22]. Esta operação pode ser representada por:

$$o = f(o_{l-1} * W_l) + b_l \quad (3.1)$$

onde o_l representa o tensor de saída da camada l , o_{l-1} representa o tensor de saída da camada anterior ou, quando $l = 1$, a entrada, $*$ representa o operador de convolução, W_l representa o tensor de pesos dos filtros da camada l , b_l representa o vetor de bias dessa mesma camada e f representa a função de ativação.

Em uma tarefa de classificação, camadas convolucionais mais altas de representação amplificam aspectos da entrada para discriminação e suprimem variações irrelevantes. Utilizando-se uma imagem como entrada de um modelo, esta tem a forma de uma matriz de valores de pixel e os recursos aprendidos na primeira camada de representação, geralmente, referem-se à presença ou à ausência de arestas e orientações, do mesmo modo, a locais específicos na imagem. A segunda camada, tipicamente, projeta motivos ao ver arranjos particulares de arestas, independentemente de pequenas variações nas posições das arestas. A terceira camada pode

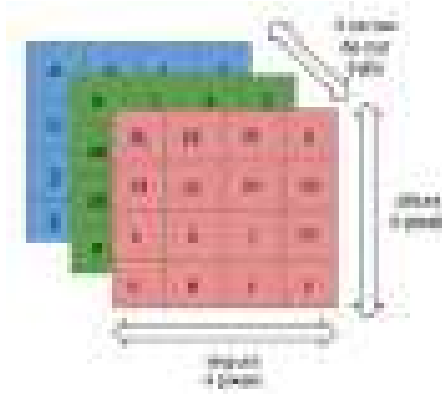


Figura 3.1: Exemplo de estrutura da uma imagem 2D



Figura 3.2: Exemplo de filtro convolucional

montar motivos em combinações maiores que correspondem a pares de objetos familiares, e as camadas subsequentes detectam objetos como combinações desses pares. O aspecto principal do aprendizado profundo é que essas camadas de recursos não são projetadas por engenheiros humanos: elas são aprendidas com dados, utilizando-se um procedimento de aprendizado de uso geral.

Cada filtro convolucional está ligado a uma função de ativação com finalidade de decisão, ajudando no aprendizado de padrões complexos. Na literatura, destacam-se funções de ativação, como sigmóide, tanh, maxout, SWISH, ReLU e variantes de ReLU, como leaky ReLU, ELU, e PReLU. Vale considerar que, nos estudos empíricos, a função ReLU e suas variantes são preferidas dentre as demais, uma vez que sua eficiência é constatada na superação do problema de desaparecimento do gradiente ([60] e [61]).

Embora o papel da camada convolucional seja detectar conjunções locais de recursos da camada anterior, o papel da camada de *pooling* é mesclar recursos semanticamente semelhantes em um. Sua função é reduzir progressivamente o tamanho espacial da representação para reduzir a quantidade de parâmetros e computação na rede. Também opera independentemente em cada fatia de profundidade da entrada e a redimensiona espacialmente. Uma unidade de *pooling* típica calcula o máximo de uma janela local de unidades em um mapa de recursos, ou em alguns mapas de recursos (figura 3.3).

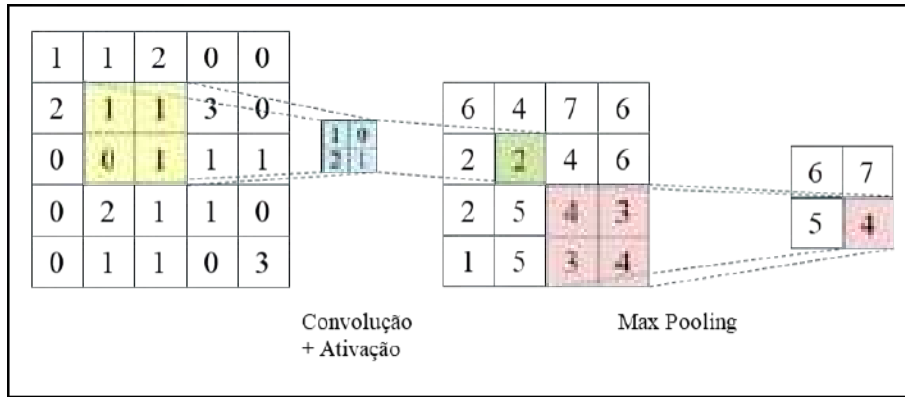


Figura 3.3: Exemplo da aplicação de convolucional com uma matriz de entrada 4x4, filtro convolucional de 2x2, função de ativação e execução do redimensionamento espacial com *Max Pooling* 2x2, deslocamento 2.

3.1.2 Evolução das Redes Neurais Convolucionais

As arquiteturas de CNN passaram por evoluções a fim de torná-las escaláveis em problemas heterogêneos, complexos e multiclasse. As inovações incluem diferentes aspectos, como modificação de unidades de processamento, estratégias de otimização de parâmetros e hiperparâmetros, padrões de design e conectividade de camadas etc. Em 1998, LeCuN propôs a rede LeNet [62] iniciando a história das CNNs profundas. Esta rede estava limitada a tarefas de reconhecimento de dígitos manuais. Sua arquitetura foi constituída por cinco camadas alternadas de convolucional e *pooling*, seguidas por duas camadas totalmente conectadas.

Em 2012, Krizhevsky propôs a rede AlexNet [15] que venceu a competição 2014-ILSVRC [63]. Desde então, inovações significativas na CNN foram propostas em grande parte atribuídas à reestruturação das unidades de processamento e ao projeto de novos blocos. A rede AlexNet é considerada a primeira arquitetura CNN profunda que apresentou resultados inovadores para tarefas de classificação e de reconhecimento de imagens. A sua profundidade foi estendida de cinco (5) (LeNet) para oito (8) camadas tornando a CNN aplicável a diversas categorias de imagens. A função de ativação não saturante ReLU foi empregada para melhorar a taxa de convergência. Outros ajustes realizados foram a utilização de filtros de grande porte (11x11 e 5x5) *pixels* nas camadas iniciais, em comparação com as redes propostas anteriormente. A técnica de regularização conhecida como dropout [64] foi utilizada para redução do sobreajuste e melhoria da generalização. A arquitetura da rede AlexNet é mostrada na figura 3.4.

Em 2014, Simonyan [65] propôs a arquitetura VGG que mostrou bons resultados tanto para classificação de imagens quanto para problemas de localização e ficou em 2º lugar na competição 2014-ILSVRC. A rede foi desenvolvida com dezenove (19) camadas de profundidade em comparação com AlexNet e substituiu os filtros

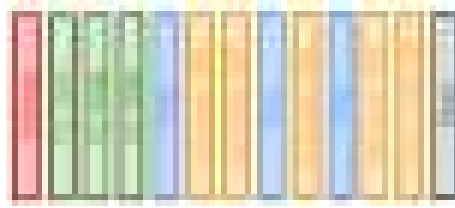


Figura 3.4: Arquitetura *AlexNet*

11x11 *pixels* e 5x5 *pixels* por uma pilha de camadas de filtros 3x3. Demonstrou experimentalmente que a colocação simultânea de filtros de tamanho pequeno (3x3) *pixels* poderia induzir o efeito do filtro de tamanhos maiores (5x5 e 7x7) *pixels*. O uso de filtros de tamanho pequeno também fornece um benefício adicional de baixa complexidade computacional, reduzindo o número de parâmetros. A rede também utiliza convoluções 1x1 *pixel* entre as camadas convolucionais possibilitando o aprendizado de uma combinação linear dos mapas de características resultantes. A rede possui 138 milhões de parâmetros, tornando-a computacionalmente custosa. A arquitetura da rede VGG é mostrada na figura 3.5.

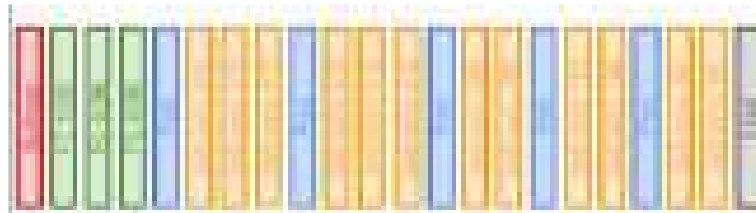


Figura 3.5: Arquitetura *VGG16*

Neste mesmo período, Szegedy e colaboradores apresentaram a rede Inception-V1 [66] (também conhecida como GoogleNet): a vencedora da competição 2014-ILSVRC. A rede Inception-V1 introduziu o novo conceito de bloco inicial na CNN por meio do qual incorpora transformações convolucionais em variadas escalas utilizando a ideia de divisão, transformação e fusão. Nessa rede, as camadas convolucionais convencionais são substituídas por pequenos blocos semelhantes que encapsulam filtros de tamanhos diferentes (1x1, 3x3 e 5x5) *pixels* para capturar informações espaciais em diferentes escalas, incluindo nível de grão fino e grosso. A rede adiciona uma camada de filtro convolucional 1x1 *pixel*, antes de empregar *kernels* de tamanho grande. Além disso, essa rede utiliza conexões esparsas (nem todos os mapas de recursos de saída estão conectados a todos os mapas de recursos de entrada) para superar o problema de informações redundantes e custo reduzido pela omissão de mapas de recursos que não eram relevantes. Destaca-se ainda a redução da densidade da conexão por meio do agrupamento médio global na última camada, em vez de usar uma camada totalmente conectada. Esses ajustes de parâmetro causaram uma diminuição significativa no número de parâmetros de 138 milhões para 4 mi-

lhões de parâmetros. Vale ressaltar que outros fatores regulatórios aplicados foram a normalização do lote e o uso de *RmsProp* como um otimizador [67]. A arquitetura do bloco *Inception* é mostrada na figura 3.6 e a arquitetura da rede é apresentada na figura 3.7.

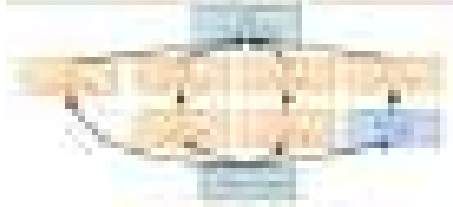


Figura 3.6: Bloco *Inception*



Figura 3.7: Arquitetura *Inception*

Em 2015, He e colaboradores apresentaram a rede *ResNet* [68], que foi a vencedora da competição 2015-ILSVRC. A rede *ResNet* foi desenvolvida com 50/101/152 camadas de profundidade e introduziu o conceito de aprendizado residual nas CNNs, resolvendo assim os problemas enfrentados durante o treinamento de redes profundas. Destaca-se como característica distinta dessa rede, a estrutura de aprendizagem residual baseada em referência que introduziu conexões de atalho dentro das camadas a fim de permitir a conectividade entre camadas; no entanto, essas conexões são independentes de dados e livres de parâmetros. As camadas representam funções não residuais, as informações residuais são sempre transmitidas e os links de identidade nunca são fechados. Estes links residuais (conexões de atalho) aceleram a convergência de redes profundas, possibilitando, assim, à *ResNet* a capacidade de evitar problemas de diminuição de gradiente. A arquitetura do bloco residual é mostrada na figura 3.8 e arquitetura da rede é apresentada na figura 3.9.



Figura 3.8: Bloco *ResNet*

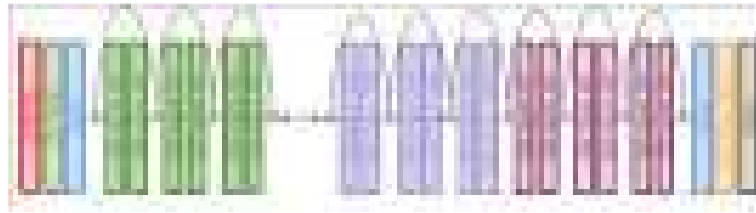


Figura 3.9: Arquitetura *ResNet*

3.1.3 Redes Autocodificadores

Os autocodificadores geralmente são implementados por redes neurais e são treinados para tentar copiar sua entrada em sua saída. Internamente, os autocodificadores possuem uma camada oculta que descreve um código usado para representar a entrada. A rede é composta por duas partes: uma função codificadora $h = f(x)$ e um decodificador que produz uma reconstrução $r = g(h)$.

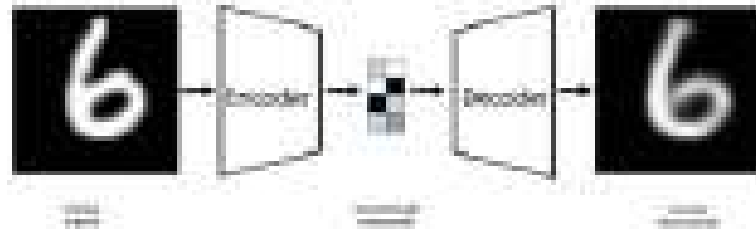


Figura 3.10: Um exemplo de autocodificador. A imagem de entrada é codificada para uma representação compactada e decodificada.

Zhai e colaboradores [69] afirmam que o conceito de autocodificador foi originalmente proposto por LeCun em sua tese de doutorado [70]. Tradicionalmente, os autocodificadores eram usados para redução de dimensionalidade ou extração de características [71], [72] e [73], contudo, posteriormente, estes foram empregados para aprender modelos generativos [74] de dados [75].

Em 2008, Vincent [76], [77] propôs os autocodificadores para *denoising* (DAEs). Essa rede adiciona intencionalmente ruídos nos dados de treinamento e treina os autocodificadores com dados corrompidos. Por meio do processo de treinamento, o autocodificador *denoising* pode recuperar a versão sem ruído dos dados de treinamento, o que implica uma robustez aprimorada. O autocodificador que recebe

um ponto de dados corrompido como entrada é treinado para prever o ponto de dados original e não corrompido como sua saída. Esse tipo de rede tem por objetivo desfazer uma corrupção na entrada, ao invés de simplesmente, copiar sua entrada.

Em 2011, Rifai e colaboradores [78] propuseram os autoencodificadores contrativos (CAEs) um termo de penalidade selecionado é adicionado à função de custo padrão na etapa de reconstrução. Esse termo de penalidade é empregado para penalizar a sensibilidade dos recursos em relação às entradas. Dessa maneira, o mapeamento do vetor de entrada para a representação converge com maior probabilidade. Os autocodificadores também podem ser utilizados com camadas convolucionais em vez de camadas totalmente conectadas (observe que o exemplo da figura 3.10 é uma imagem). Neste caso, utiliza-se imagens 2D em vez de vetores 1D, sendo assim, imagem de entrada é reduzida para fornecer uma representação latente de dimensões menores e forçar o autocodificador a aprender uma versão compactada das imagens. Vale ressaltar que uma camada CNN é composta por camadas convolucionais e *pooling*, sendo que uma camada convolucional consiste em vários nós convolucionais cujas entradas são mapas de características bidimensionais e os parâmetros de aprendizado são os elementos das matrizes de filtro. Assim como uma rede autocodificador simples, o tamanho da camada de entrada também é o mesmo das camadas de saída, porém a estrutura do codificador é composta por camadas de convolução enquanto a estrutura do decodificador é composta pelas camadas convolucionais transpostas.

No universo de autocodificadores, arquiteturas de redes neurais profundas são formadas pela concatenação os modelos básicos, hierarquicamente. Um autocodificador empilhado herda os benefícios de uma rede profunda com maior poder expressivo. A primeira camada oculta do autocodificador empilhado tende a aprender o recurso de primeira ordem a partir dos dados de entrada. A camada oculta sucessiva gera o recurso de alto nível correspondente ao padrão na aparência dos de níveis anteriores. Uma arquitetura profunda é formada pela composição de múltiplos níveis de representações, da mesma forma, o autocodificador básico pode ser concatenado para criar o autocodificador empilhado em modelo profundo, sendo composto por uma camada visível, várias camadas ocultas e uma camada de saída. A saída de cada camada anterior é conectada à entrada da camada sucessiva.

A figura 3.11 apresenta um exemplo de um autocodificador simples com camadas totalmente conectadas enquanto a figura 3.12 exemplifica uma estrutura de um autocodificador convolucional.

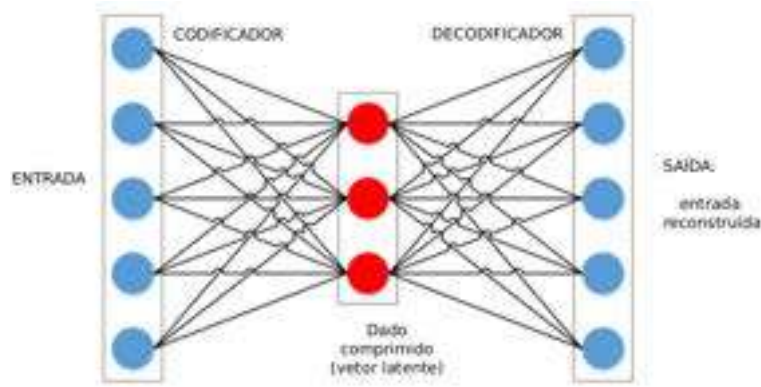


Figura 3.11: Arquitetura *autoencoder* com três camadas

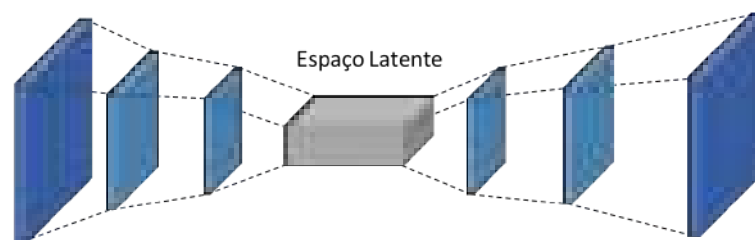


Figura 3.12: Estrutura *convolucional autoencoder*: cada caixa azul representa a camada de convolução e a camada cinza, o código de vetor latente. As camadas de *pooling* foram ignoradas

3.2 Aprendizado profundo para atenuação de ruído em dados sísmicos

Na área de geociência existem diversos nichos de pesquisa, desde aplicações em dados de perfil de poço até imagens sísmicas. Quanto ao aspecto das aplicações, é possível afirmar que essas introduzem novos desafios ao campo da investigação, uma vez que à complexidade e o volume de dados a serem analisados inerentes a cada problema demandam novas pesquisas na área de aprendizado de máquina. Um problema de interesse ligado ao processo de geofísica de exploração é a interpretação de dados sísmicos. Na literatura, existem trabalhos diversos que buscam automatizar o processo de interpretação empregando técnicas de inteligência artificial, entre eles destacam-se [79], [80], [81] nos quais são propostos modelos para segmentação de estruturas tais como falhas, regiões de sal, entre outras. A reconstrução do modelo de velocidade a partir do processamento de dados sísmicos gera, segundo esses autores, um impacto muito grande no processo de localização e imageamento de corpos alvo.

Recentemente, o método mais atrativo para reconstrução do modelo de velocidade com maior acurácia e resolução tem sido o *full waveform inversion (FWI)*. Considerando as pesquisas realizadas, foram propostas diversas abordagens utilizando-se os métodos de aprendizado de máquina, entre elas destacam-se: métodos para acele-

rar a convergência do *FWI* [82] ou seja, um algoritmo objetiva obter os parâmetros do meio a partir do processo de inversão; métodos nos quais o modelo de inteligência artificial é treinado para realizar o processo de inversão *end-to-end*, isto é, usando o dado sísmico como entrada, o modelo diretamente deriva o operador de inversão e produz, na sua saída, o modelo de velocidade correspondente [83] [84].

A atenuação de ruído para imagens usando a tecnologia de aprendizado de máquina alcançou grande sucesso no campo da visão computacional. O aprendizado profundo na tarefa de atenuação de ruídos para imagens foi desenvolvido nas últimas décadas ([85]; [86] e [87]) e o resultado pode ser atestado em trabalhos de pesquisa que indicam [88] um salto gigantesco nesse campo. As redes neurais convolucionais (CNNs) aprendendo representações dimensionais inferiores dos recursos da imagem durante o treinamento. Utilizando CNNs, redes residuais profundas (*ResNet*) [68] e normalização de lotes [89], [90] propõem modelos de atenuação de ruído com CNN e superam os métodos tradicionais não baseados na CNN. As técnicas baseadas na CNN para atenuação de ruído têm sido ampla e continuamente desenvolvidas em muitas variantes: modelo Noise2Noise [91] foi introduzido para a tarefa de atenuação de ruído sem fornecer informações sobre a verdade do solo. O CBDNet [92] consiste em duas sub-redes (estimativa de ruído e *denoising* não cego) e atinge resultados de última geração em termos de funções de avaliação quantitativas e qualidade visual. Da mesma forma, FFDNet [93], RED30 [94], BM3D-Net [95] e CS-DIP [96] também alcançaram desempenhos proeminentes.

Mandelli e colaboradores [97] propõem o uso de uma arquitetura de *autoencoder* convolucional conhecida pelo nome de *U-Net* [98], projetada originalmente para segmentação de imagens, posteriormente, empregada em diferentes tarefas de processamento de imagens [99]. Esta arquitetura é treinada empregando como entrada janelas do dado sísmico com adição do ruído e, como alvo para o treinamento supervisionado, é utilizado o mesmo conjunto de janelas sem a presença do ruído.

Outro exemplo similar à abordagem anterior é apresentado por Jin e colaboradores [100] que empregam como base a arquitetura *autoencoder* convolucional, com adição de camadas do tipo residual [68] permitindo a implementação de modelos mais profundos. Essa camada do tipo residual auxilia o processo de convergência do modelo, reduzindo o problema de desvanecimento ou explosão do gradiente. Seguindo esta mesma ideia, Zhang e colaboradores [101] empregaram este tipo de camadas numa rede neural convolucional a fim de realizar o processo de atenuação, mas os pares entrada-saída usados no treinamento do modelo consistem no dado ruidoso e o resíduo entre este dado e o dado original, respectivamente, e isto é conhecido como *residual learning*. O mesmo autor, em outro trabalho, utiliza técnica de aprendizado com o mesmo objetivo de atenuação de ruído [93], utiliza camadas de convolução dilatadas [102] as quais aumentam o campo receptivo da rede sem

aumentar o número de parâmetros do modelo.

Uma outra alternativa, visando aumentar a qualidade dos resultados dos modelos de aprendizado profundo, é o uso de funções custo específicas para o problema em questão. A seleção desta função não é uma tarefa trivial e, na literatura, existem estudos em diversas opções empregadas no processamento de imagens [103] [104]. Outro caminho é a utilização de um tipo de arquitetura conhecida por Rede Generativa Adversária *GAN*. Goodfellow [74] propôs um novo paradigma de treinamento, no qual duas redes neurais, chamadas de gerador e discriminador, participariam de um jogo para dois jogadores com soma zero e máxima, permitindo às redes aprender a distribuição de dados. Alwon [105] emprega uma arquitetura deste tipo para realizar o processo de atenuação de ruído em dados sísmicos. Esta arquitetura é composta por DCGAN [106] como discriminador, rede *U-Net* [98] como gerador e função custo *wasserstein* [107]. Por fim, e não menos considerável, Isola e colaboradores [108] propõem uma arquitetura GAN, composta por um gerador e discriminador, sendo o gerador baseado em uma rede com camadas totalmente conectadas e uma rede neural convolucional (CNN). Ambas são treinadas usando o mesmo mecanismo de par entrada saída apresentado por Mandelli [97].

3.3 Resumo do capítulo

Neste capítulo foram introduzidos conceitos de redes neurais profundas e uma revisão bibliográfica de aplicações para atenuação de ruídos em dados sísmicos pré-empilhamento. O próximo capítulo traz os dados sísmicos sintéticos e reais utilizados neste trabalho, bem como as transformações realizadas nos mesmos, os modelos convolucionais utilizados neste estudo e o detalhamento de todos os experimentos executados nesta pesquisa.

Capítulo 4

Materiais e Métodos

Neste capítulo serão apresentados as arquiteturas convolucionais e os dados sintéticos e reais utilizados neste trabalho, bem como as transformações necessárias nos dados sísmicos para treinamento das redes, como adição e extração de ruídos. Em seguida, serão explicados os experimentos realizados e como os resultados foram avaliados.

4.1 Dados

Neste trabalho foram utilizados os dados sísmicos pré-empilhamento sintéticos e reais para treinamento e testes em atenuação de ruído sísmico. Para maior compreensão, vale ressaltar que nos dados sintéticos foram adicionados ruídos aleatórios e lineares, enquanto que nos dados reais já estava presentes de forma predominante a presença do ruído *Swell*.

4.1.1 Formato SEG-Y

Os dados registrados em campo (aquisição), dito bruto, não passam de uma sequência de amostras associadas a cada receptor, sendo assim, para que esses dados possam ser trabalhados e interpretados é necessário atribuir informação a cada traço, localizando-o espacialmente e definindo parâmetros utilizados em sua aquisição.

De forma geral, estes dados são gravados em um modo multiplexado utilizando-se um determinado tipo de formato. Os dados são desmultiplexados conforme descrito na Figura 4.1. Matematicamente, a desmultiplexação é vista como a transposição de uma matriz grande, para que as colunas da matriz resultante possam ser lidas como traços sísmicos registrados em diferentes deslocamentos com um ponto de disparo comum. A partir desse estágio, os dados são convertidos para um formato conveniente usado durante todo o processamento, formato este determinado pelo tipo de sistema de processamento e pela empresa que responsável pela pesquisa sísmica. Contudo, o formato SEG-Y [109], estabelecido pela Sociedade de Geofísicos

de Exploração [5], foi universalmente adotado pela indústria sísmica para troca de dados.

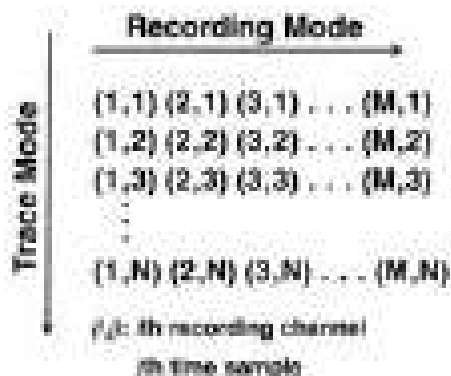


Figura 4.1: Os dados sísmicos são registrados em linhas de amostras - amostras ao mesmo tempo em canais consecutivos. A demultiplexação envolve a classificação dos dados em colunas de amostras - todas as amostras em um canal seguidos por outros canais seguintes.

De acordo com a literatura, este padrão foi formulado em 1973 para o registro de dados sísmicos de reflexão e, posteriormente, publicado em 1975. Um arquivo SEG-Y (figura 4.2) pode armazenar uma quantidade arbitrária de traços sísmicos. No início do arquivo há espaço reservado para armazenar informações gerais do dado sísmico. Depois segue-se uma sequência de registros de traços. O registro de cada traço é composto por um cabeçalho e pelo dado binário propriamente dito. O cabeçalho de cada traço ocupa 240 bytes, enquanto que o dado binário ocupa 4 bytes por amostra registrada.

Para leitura e manipulação dos arquivos SEG-Y, neste trabalho, foram utilizados o software Seismic Unix e a biblioteca segyio [110] com linguagem Python [19]. O Seismic Un*x (SU) é um pacote de programas de linha de comando para o processamento de dados sísmicos 2D desenvolvido como software livre [111]. Esse pacote foi inspirado no funcionamento usual do sistema operacional Unix.

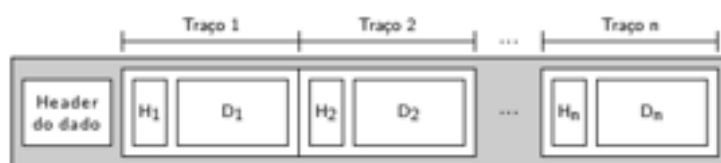


Figura 4.2: Estrutura de um dado sísmico no formato SEG-Y

4.1.2 Dados Sintéticos

Considerando os objetivos a serem alcançados nesta pesquisa, foram gerados dois dados sísmicos pré-empilhamento sintéticos, obtidos pela técnica de modelagem sísmica.

mica a partir do método de diferenças finitas [18]. O primeiro dado sísmico foi gerado com 71 tiros, 701 receptores, profundidade de 6000 km e foi criado a partir do modelo de velocidade M1 [112] (figura 4.3). Já o segundo dado sísmico foi gerado com 160 tiros, 801 receptores, profundidade de 7000 km e foi criado a partir do modelo de velocidade M2 [113] (figura 4.5).

Na figura 4.4 são apresentados exemplos de tiros do dado sísmico gerados a partir do modelo de velocidade M1 (figura 4.3) e na figura 4.6 são apresentados exemplos de tiros do dado sísmico gerado a partir do modelo de velocidade M2 (figura 4.5).

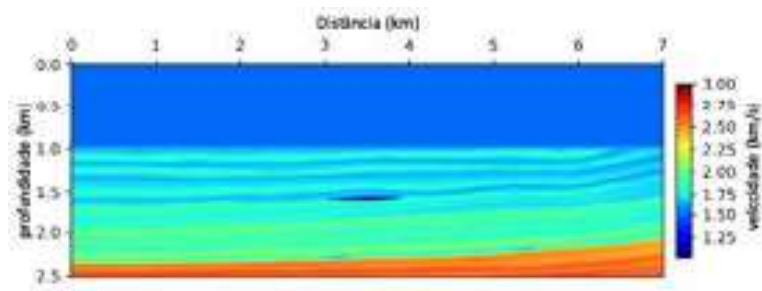


Figura 4.3: Modelo de Velocidade - M1

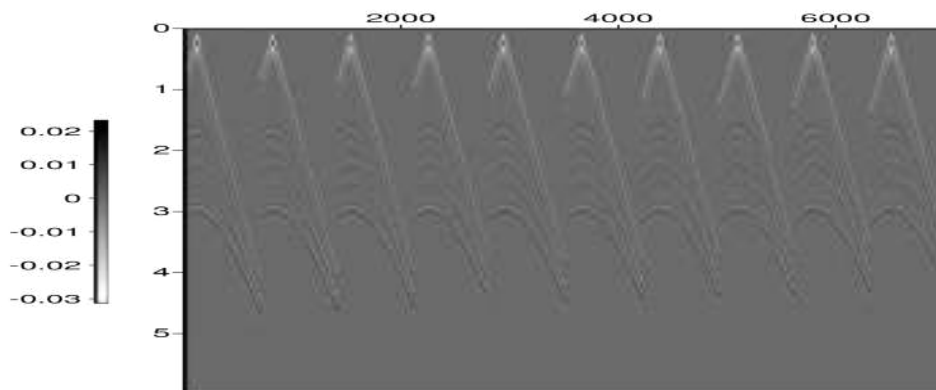


Figura 4.4: Tiros gerados a partir do Modelo de Velocidade M1

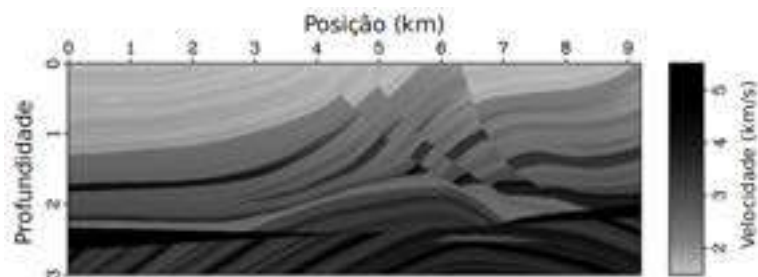


Figura 4.5: Modelo de Velocidade - M2

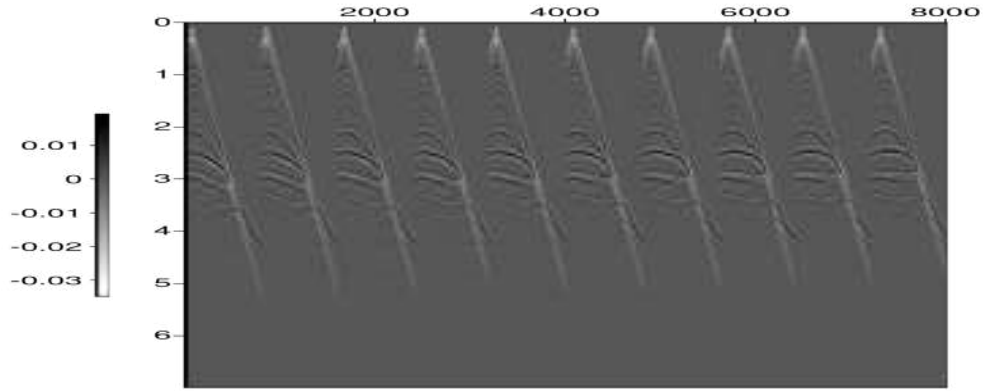


Figura 4.6: Tiros gerados a partir do Modelo de Velocidade M2

Ruído aleatório

Foram gerados três dados sísmicos a partir do modelo de velocidade M1, com adição de ruídos gaussianos de intensidades 10, 15 e 20 decibéis. O *software* Seismic Un*x (SU) foi utilizado para geração destes dados.

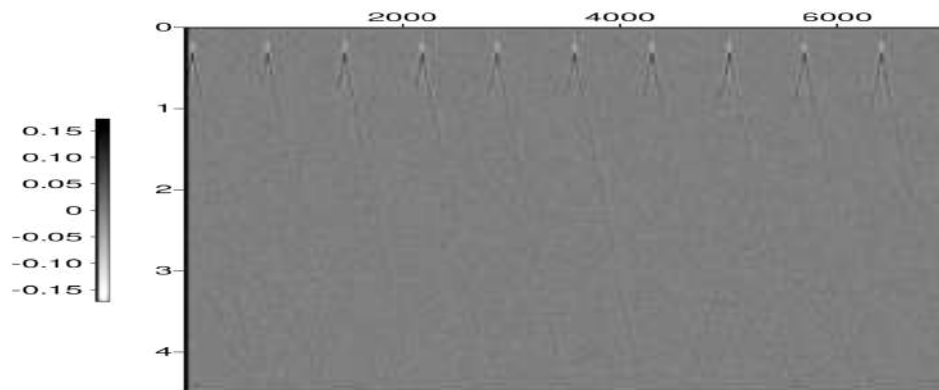


Figura 4.7: Tiros com ruído gaussiano de 10dB gerados a partir do Modelo de Velocidade M1

Ruídos lineares

Foram gerados dois dados sísmicos com ruídos lineares representados por eventos com desvios e inclinações aleatórias. Utilizou-se os modelos de velocidades M1 e M2 para geração de cada dado sísmico.

A figura 4.8 traz exemplos de tiros com ruídos lineares gerados a partir dos modelos de velocidade M1 e a figura 4.9 apresenta exemplos de tiros com ruídos lineares gerados a partir dos modelos de velocidade M2.

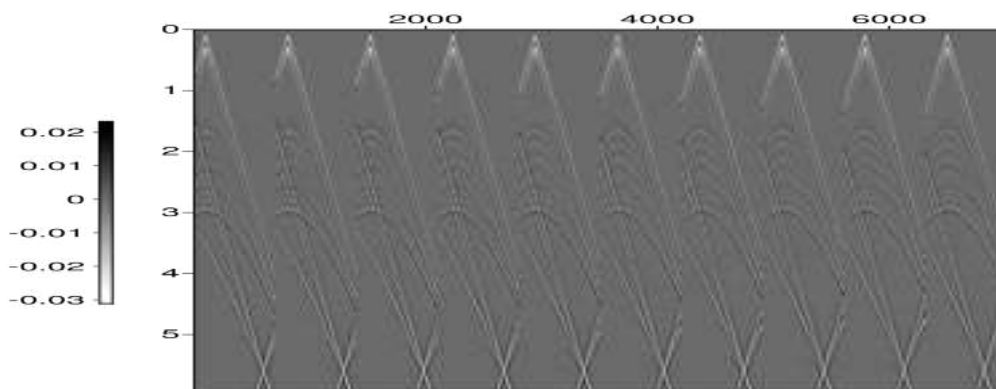


Figura 4.8: Tiros com ruído linear gerados a partir do Modelo de Velocidade M1

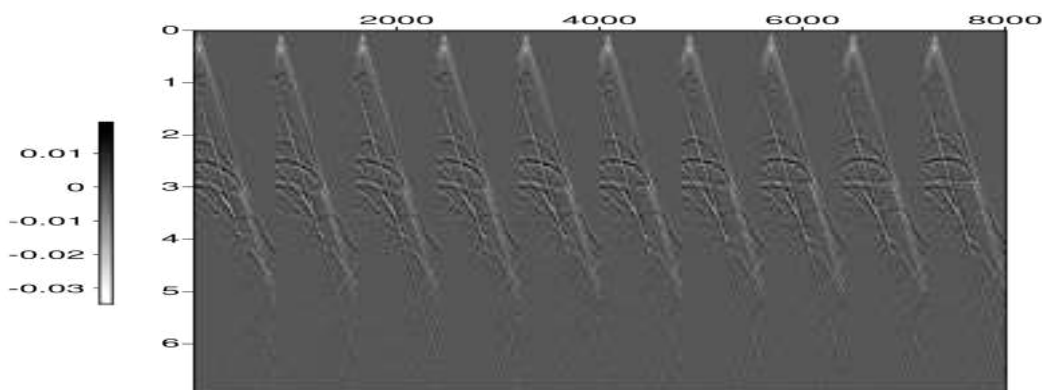


Figura 4.9: Tiros com ruído linear gerados a partir do Modelo de Velocidade M2

Ruído *Swell*

No intuito de gerar este dado sísmico, foi extraído o ruído *swell* do dado sísmico real pré-empilhamento da bacia do Espírito Santo (seção 4.1.3) e adicionados ao dado sintético gerado a partir do modelo de velocidade M2 (figura 4.5). Observa-se na figura 4.10 exemplos de tiros o dado sintético com ruído *swell* real.

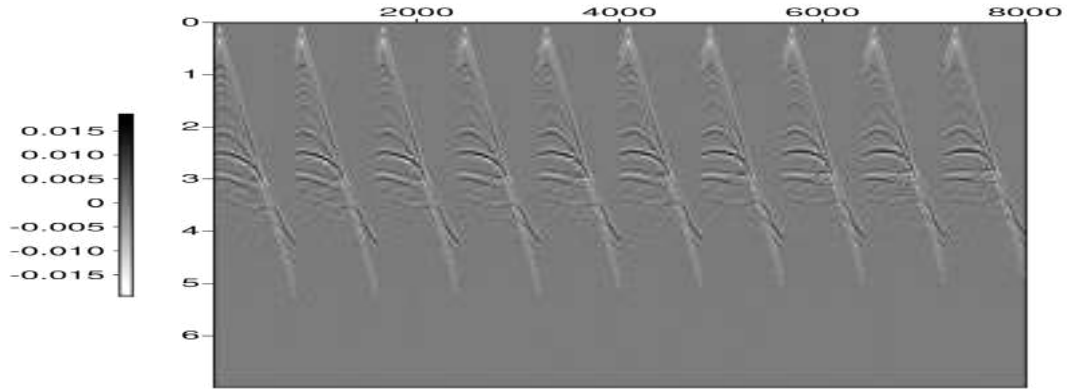


Figura 4.10: Tiros com ruído *swell* retirados do dado sísmico pré-empilhamento da bacia do Espírito Santo e adicionado ao dado sísmico sintético gerado pelo Modelo de Velocidade M2

4.1.3 Dados Reais

Diante do contexto e das práticas apresentadas, a pesquisa avançou com a utilização de dois dados sísmicos 2-D pré-empilhamento oriundos dos levantamentos da bacia de Santos e da bacia do Espírito Santo.

Bacia de Santos

O dado sísmico pré-empilhamento citado encontra-se localizado na Bacia de Santos pertencente ao levantamento 0231 SANTOS 5A, no campo de produção Estrela do Mar (Figura 4.11). Possui 1335 tiros com 150 receptores. A geometria de aquisição desse levantamento foi realizada com espaçamento entre receptores de 26.667 m, espaçamento entre as fontes de 26.667 m e *Offset* mínimo de 335.33 m. A figura 4.12 apresenta exemplo de tiros do dado sísmico 541 da bacia de Santos.

Bacia do Espírito Santo

A linha sísmicas 2-D pré-empilhamento 0259-1411 localizada na bacia do Espírito Santo, pertencentes ao levantamento 0259_2D_SPP_2Q_1999, como é possível observar no mapa da Figura 4.13. A geometria de aquisição desse levantamento foi realizada com espaçamento entre receptores de 12.5 m, espaçamento entre as fontes de 37.5 m e *Offset* mínimo de 252 m. A figura 4.14 apresenta exemplo de tiros do dado sísmico da bacia do Espírito Santo.

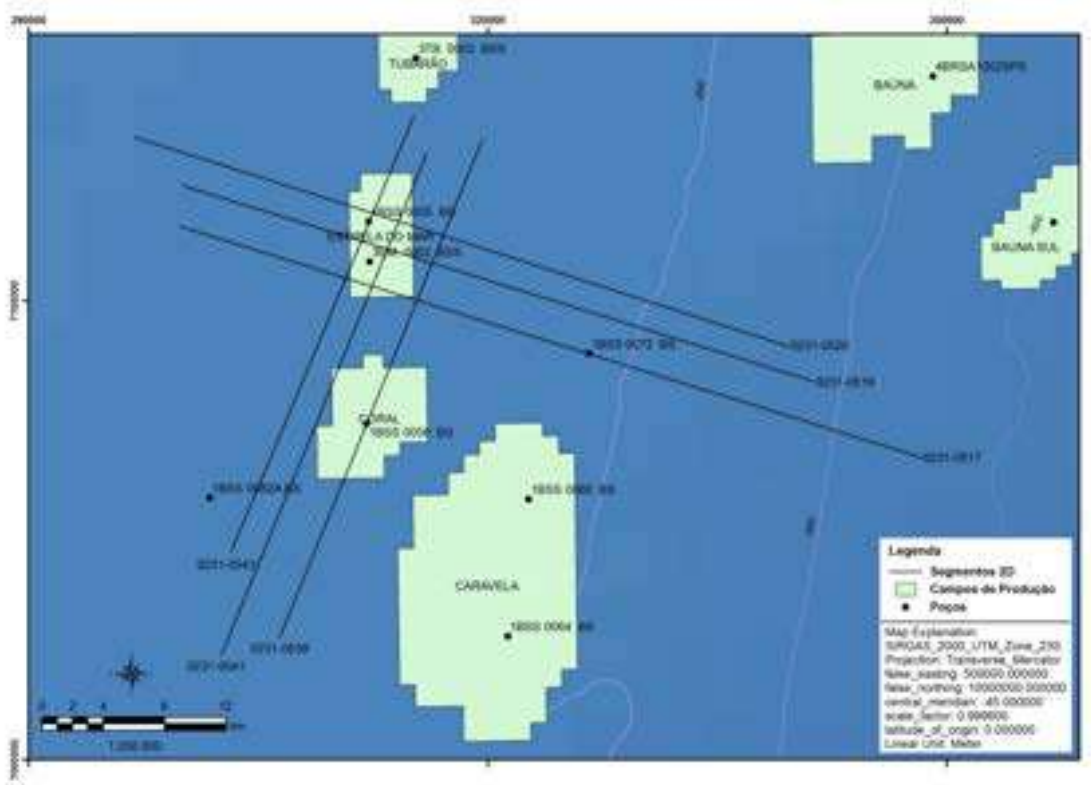


Figura 4.11: Localização geográfica das linhas sísmicas da bacia de Santos - campo de produção Estrela do Mar

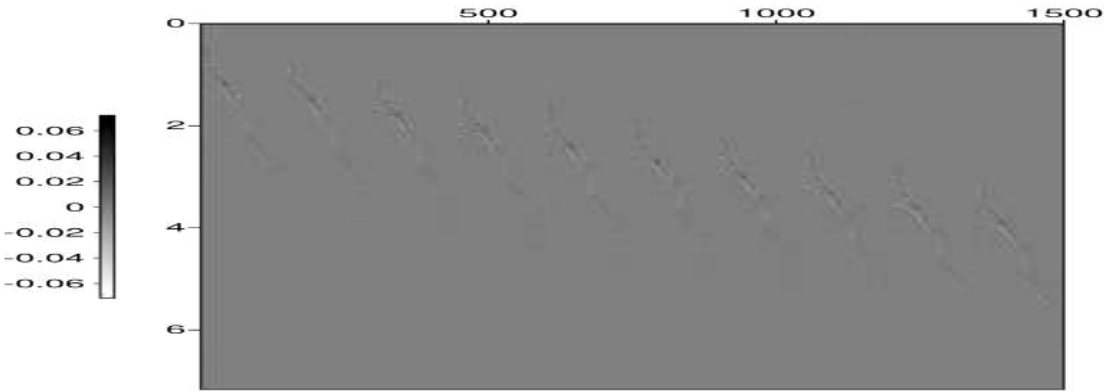


Figura 4.12: Exemplo de tiros do dado sísmico pré-empilhamento da bacia de Santos (linha 541)

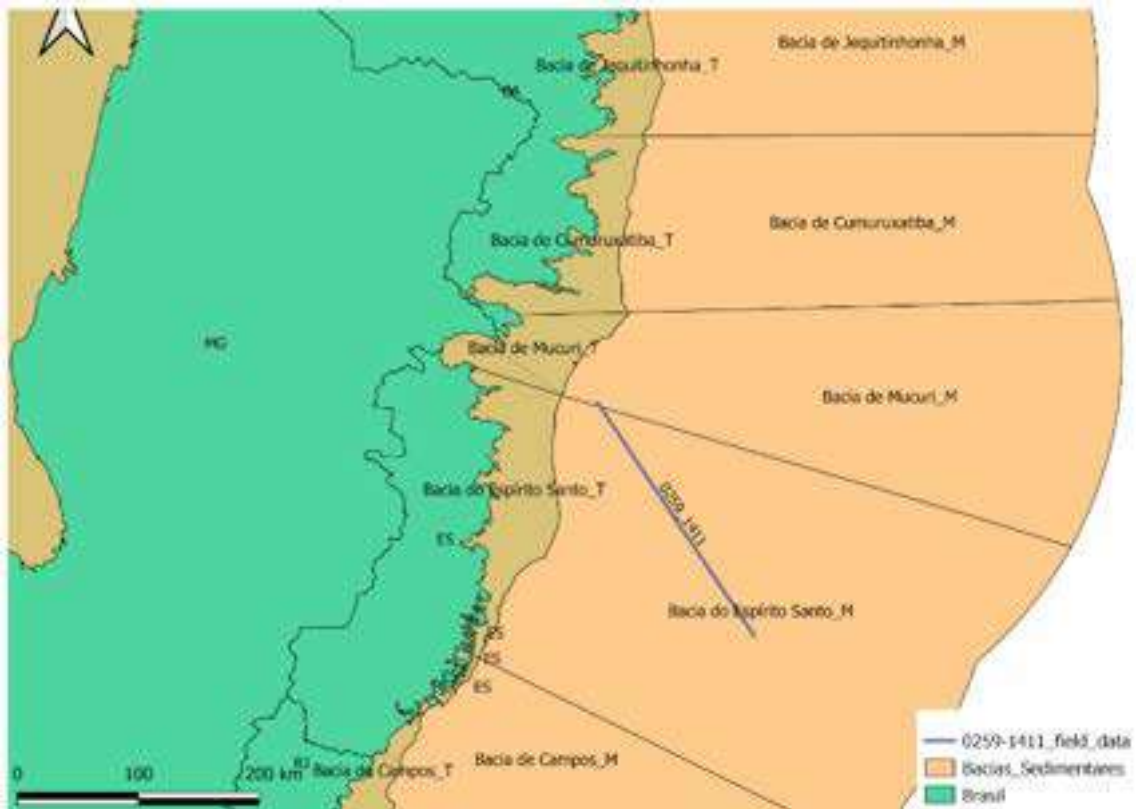


Figura 4.13: Localização geográfica das linhas sísmicas da bacia do Espírito Santo

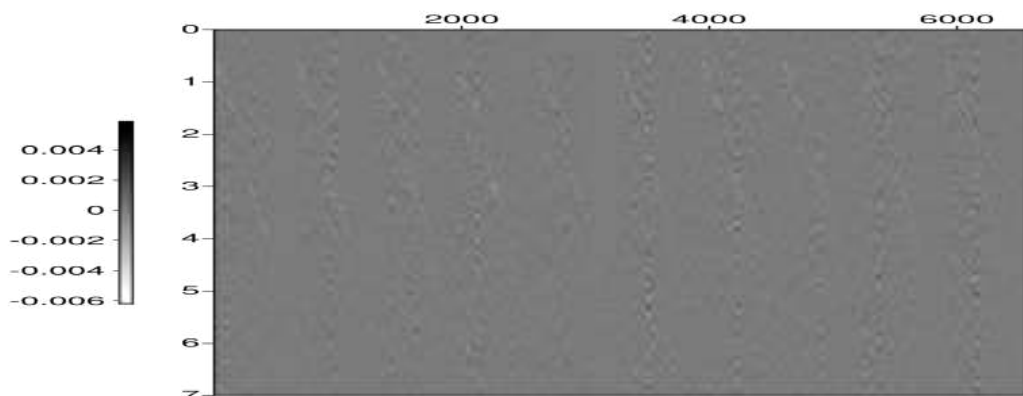


Figura 4.14: Exemplo de tiros do dado sísmico pré-empilhamento da bacia do Espírito Santo

4.1.4 Pré-processamento nos Dados Reais

A partir da disponibilização dos dados sísmicos pré-empilhamento reais em sua forma bruta, foi-se necessário realizar um procedimento conhecido como o pré-processamento para obtenção dos dados sísmicos reais sem a presença de ruídos. Isso se fez necessário para treinamento das redes convolucionais, uma vez que é preciso fornecer a rede um par de tiro sísmico (com a presença de ruído e sem a presença de ruído). Já no caso dos dados sintéticos, foram gerados sem a presença de ruídos, fazendo-se necessário a adição de ruídos para geração do par sísmico a ser utilizados no processo de treinamento das redes convolucionais. O procedimento de pré-processamento sísmico dos dados reais normalmente foi realizado por um geofísico com a utilização de técnicas de filtragem de frequência.

Em um pré-processamento, geralmente, é necessário remover ambas amplitudes de alta e de baixa frequência e essa filtragem pode ser alcançada através da aplicação de um filtro passa-banda projetado para manter os componentes de amplitude dentro de uma banda de frequência enquanto remove as amplitudes fora desta banda do espectro de amplitude de entrada.

Para realização da atenuação dos ruídos no dado sísmico pré-empilhamento localizado na bacia do Espírito Santo foi projetado um filtro passa-banda do tipo *Butterworth*. Esse operador requer duas frequências para valores de corte de baixa e alta frequência e dois valores de inclinação para a definição da banda de transição de o trapézio (Figura 2.13 C). O espectro do operador de filtro mostrado na Figura 2.13 A é multiplicado pelo espectro do dado bruto no domínio da frequência ou suas representações no domínio do tempo do operador podem ser convoluídas com os dados brutos de entrada para obter o resultado do filtro passa-banda. Os parâmetros de frequência utilizados nesse filtro foram $f_1=5\text{Hz}$, $f_2=100\text{ Hz}$ e os parâmetros de inclinação para a definição da banda de transição foram $s_1=18\text{dB/Oct}$, $s_2=36\text{dB/Oct}$. A ferramenta utilizada para a execução desse processamento foi o *SeisSpace ProMax* [114].

Na figura 4.15 é apresentado o dado sísmico pré-empilhamento da linha do Espírito Santo em seu estado bruto.

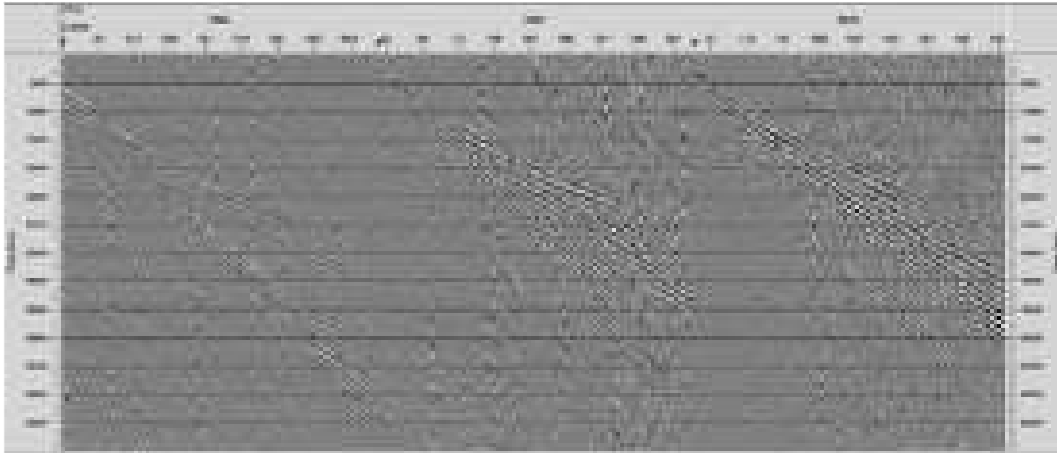


Figura 4.15: Tiros do dados sísmico pré-empilhamento da bacia do Espírito Santo em seu estado bruto

Na figura 4.16 é apresentado exemplo de um tiro pré-empilhamento da linha do Espírito Santo em sua forma bruta e o resultado de filtragens com faixas de frequência de 5-20 Hz e 30-80 Hz. Observa-se grande quantidade ruídos *swell* concentrados na faixa de frequência entre 5-20 Hz.

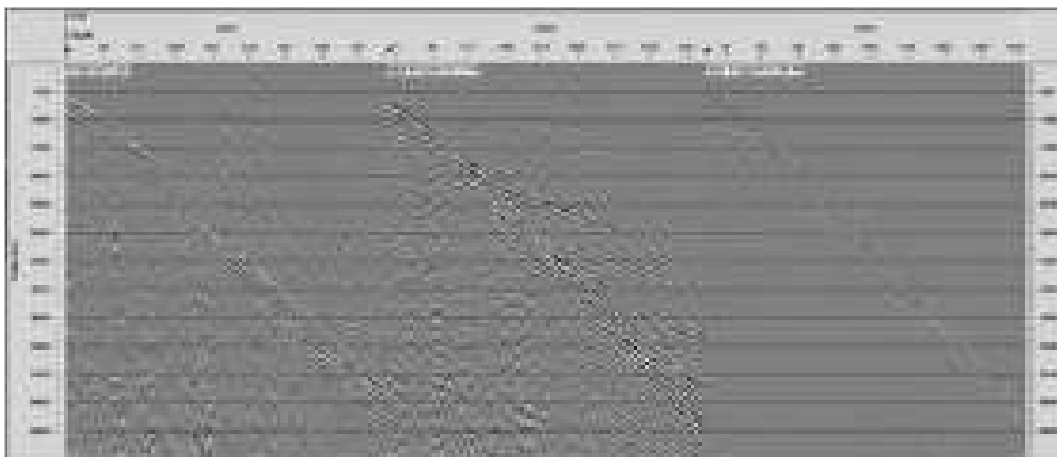


Figura 4.16: Tiros do dados sísmico pré-empilhamento da bacia do Espírito Santo em seu estado bruto e com filtragens com filtro passa-banda com faixas de frequência de 5-20 Hz e 30-80 Hz

Nota-se na figura 4.17 o resultado do processamento do dado sísmico pré-empilhamento da linha do Espírito Santo utilizando o filtro *Butterworth* com faixa de frequência de 5-100 Hz e bandas de transição de 18dB/Oct e s2=36dB/Oct.

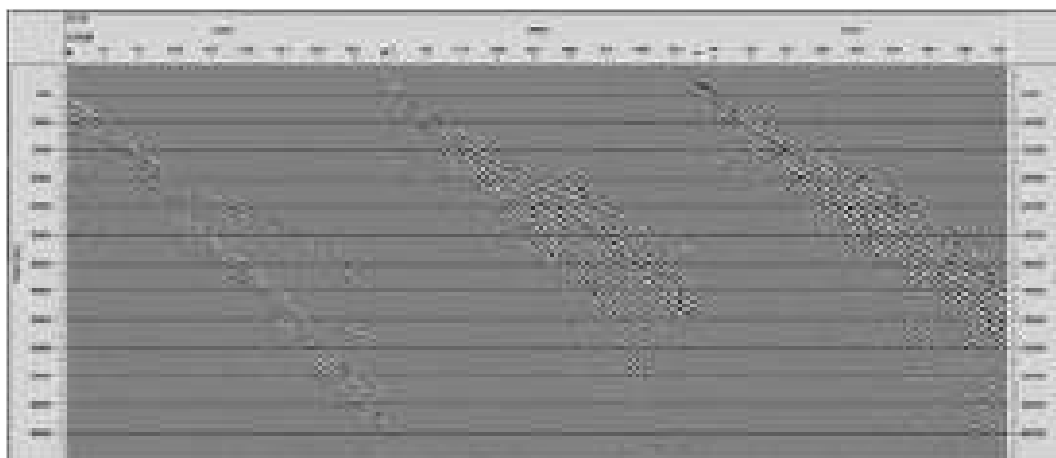


Figura 4.17: Tiros dos dados sísmico pré-empilhamento da bacia do Espírito Santo filtrada técnica *Butterworth*

O filtro de imersão F-K também foi utilizado para atenuação de ruídos no dado sísmico pré-empilhamento localizado na bacia do Espírito Santo. Vale ressaltar que a filtragem por imersão utiliza uma transformada 2D de *Fourier* nos domínios de tempo e espaço para remover tipos de ruído correlacionados que propagam-se com velocidades diferentes. Na figura 4.18 apresenta-se exemplo de um tiro e seu respectivo espectro F-K com alguns eventos significativos sobrepostos do dado sísmico pré-empilhamento localizado na bacia do Espírito Santo. Observa-se em (A) ruído *swell* de baixa frequência circulado por um retângulo pontilhado e um polígono protegendo as amplitudes de reflexão. Em (B) é possível apresentar o resultado após a aplicação do filtro F-K: nota-se que o ruído *swell* de baixa frequência for retirado, conforme indicado através do retângulo pontilhado.

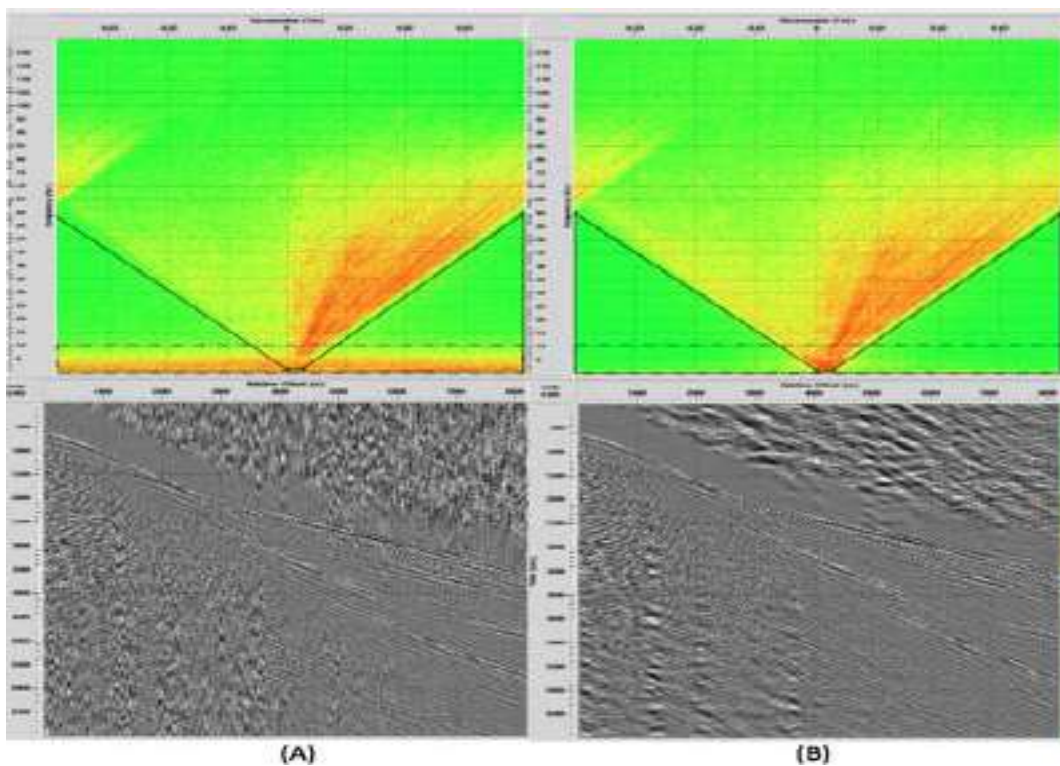


Figura 4.18: Exemplo de um tiro e seu respectivo espectro F-K do dado sísmico pré-empilhamento localizado na bacia do Espírito Santo. Em (A) encontra-se o dado em sua forma bruta e seu respectivo espectro F-K. Em (B), o resultado do tiro após filtragem utilizando técnica F-K

Dando continuidade à pesquisa, observa-se o resultado da filtragem utilizando a técnica F-K no dado sísmico pré-empilhamento da linha do Espírito Santo (figura 4.19).

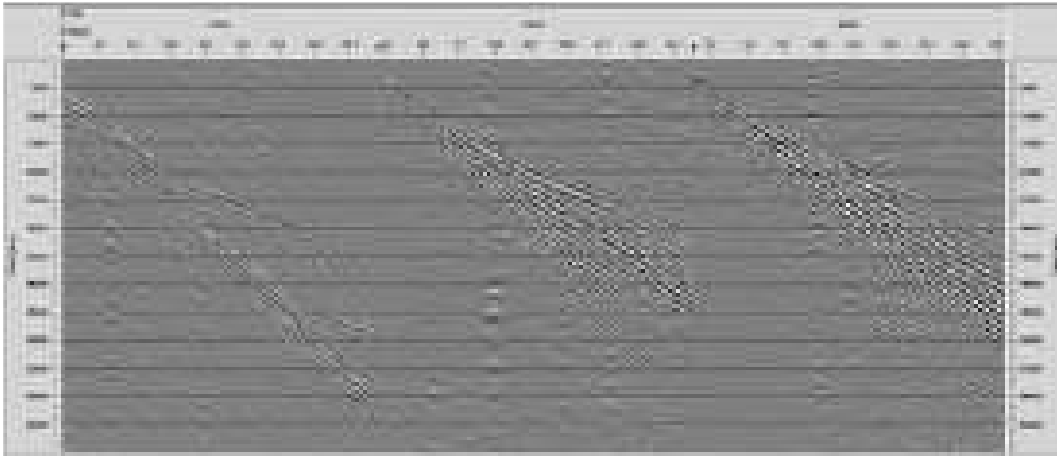


Figura 4.19: Exemplo de tiros dos dados sísmicos pré-empilhamento da bacia do Espírito Santo filtrada com técnica F-K

Na figura 4.20 é apresentado exemplo de um tiro e seu respectivo espectro F-K com alguns eventos significativos sobrepostos do dado sísmico pré-empilhamento localizado na bacia do Espírito Santo. A partir do exposto, observa-se em (A) ruído *swell* de baixa frequência circulado por um retângulo pontilhado e um polígono protegendo as amplitudes de reflexão. Enquanto que, em (B) é apresentado o resultado após a aplicação do filtro F-K e aplicação do *muting* na coluna de água. Nota-se que o ruído *swell* de baixa frequência foi retirado, conforme indicado no retângulo pontilhado e nos sinais anulados, no topo direito da imagem.

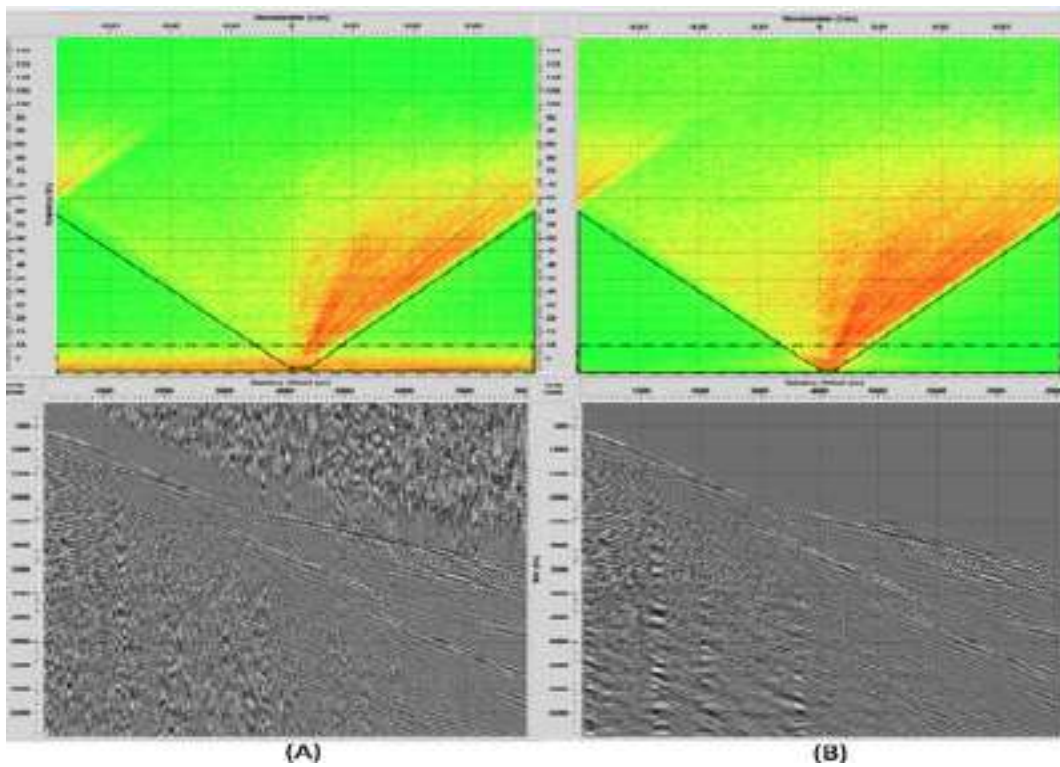


Figura 4.20: Exemplo de um tiro e seu respectivo espectro F-K do dado sísmico pré-empilhamento localizado na bacia do Espírito Santo. Em (A) encontra-se o dado em sua forma bruta e seu respectivo espectro F-K. Em (B) o resultado do tiro após filtragem utilizando técnica F-K e aplicação do *muring* na coluna de água

Uma das etapas do pré-processamento executadas no dado sísmico pré-empilhamento do Espírito Santo foi a eliminação das amplitudes dos traços na coluna de água, este processo é conhecido como *muting*. Na figura 4.21 é apresentado o resultado da filtragem utilizando a técnica F-K e aplicação do *muting* na coluna de água no dado sísmico pré-empilhamento da linha do Espírito Santo.



Figura 4.21: Exemplo de tiros dos dados sísmicos pré-empilhamento da bacia do Espírito Santo filtrados com técnica F-K e aplicação do *muting* na coluna de água

Os resultados apresentados nesta seção foram obtidos por meio da utilização de dados sísmicos pré-empilhamento reais e sintéticos, com seus respectivos ruídos e tipos de filtragem. O resumo pode ser observado na tabela 4.1.

Tabela 4.1: Dados Sísmicos pré-empilhamento reais e sintéticos utilizados neste trabalho

Tipo Dado Sísmico	Modelo Velocidade	Tipo Ruído	Dimensões
Dado Sintético	M1	Aleatório 10dB	tiros: 141
Dado Sintético	M1	Aleatório 15dB	receptores: 701
Dado Sintético	M1	Aleatório 20dB	profundidade: 4500 km
		.	dt: 0.001
Dado Sintético	M1	Linear	tiros: 71
			receptores: 701
			profundidade: 6000 km
			dt: 0.001
Dado Sintético	M2	Linear	tiros: 160
			receptores: 701
Dado Sintético	M2	<i>Swell</i>	profundidade: 6000 km
		.	dt: 0.001
		.	
Tipo Dado Sísmico	Filtragem	Tipo Ruído	Dimensões
Dado Real - Bacia do ES	Passa-banda	<i>Swell</i>	tiros: 1201
			receptores: 656
Dado Real - Bacia do ES	F-K	<i>Swell</i>	profundidade: 1751 km
			dt: 0.004
Dado Real - Bacia do ES	F-K <i>mute</i> no topo	<i>Swell</i>	
			tiros: 1335
			receptores: 150
Dado Real - Bacia Santos	Passa-banda	<i>Swell</i>	profundidade: 3585 km
			dt: 0.002

4.2 Modelos

Considerando a literatura e seus atuais apontamentos empíricos, foram implementados, neste trabalho, dois modelos de aprendizado profundo com arquitetura auto-codificador convolucional. Uma rede convolucional é composta por uma sequência de camadas na qual cada camada transforma um volume de ativações em um outro volume através de uma função diferenciável. Essa estrutura é constituída por um bloco com a sequência de três camadas: Camada Convolucional, Camada de *Pooling* e Camada Totalmente Conectada. Essas camadas são empilhadas para formar uma arquitetura convolucional completa, de modo que torna-se possível a repetição das camadas e o empilhamento dessa sequência. Nesse contexto, esta seção tem por objetivo a extração de características e espera-se que modelos com mais camadas possam extrair características complexas.

Existem bibliotecas disponíveis para implementação de redes neurais convolucionais, dentre elas é possível citar algumas populares como *Tensorflow*[20] [115],

Pytorch[116], *MXNet* [117] e *The Microsoft Cognitive Toolkit*[118]. Por meio destes *frameworks*, as redes são implementadas adicionando-se camada a camada, de modo a construir o bloco convolucional. Diante deste cenário, o *frameworks* selecionado para o desenvolvimento deste trabalho foi o *Tensorflow* com linhagem *Python*[19]. O código de treinamento foi desenvolvido para a utilização paralelismo de GPUs; a infraestrutura utilizada para tal atividade foi um servidor Nvidia DGX2 com 8 GPUs V100 que compõe o supercomputador Santos Dumont do Laboratório Nacional de Computação Científica (LNCC) [21]. Merece destaque o fato de que todos os treinamentos executados utilizaram paralelização com as oito GPUs contidas no servidor devido à grande quantidade de dados de entrada e os vários testes realizados com dados sísmicos pré-empilhamento reais e sintéticos contendo variações diversas de ruídos. Neste trabalho foram testados vários modelos com arquitetura convolucional, contudo, foram selecionados duas arquiteturas e os seus resultados serão apresentados no capítulo seguinte. Os códigos desenvolvidos referente aos modelos testados foram disponibilizados em um repositório público do *github* no endereço https://github.com/carlossneto/seismic_denoising.

No contexto de visão computacional, as entradas das redes convolucionais são usualmente matrizes tridimensionais com altura, largura e profundidade determinadas pela quantidade de canais de cores. O dado de entrada utilizado nos modelos desenvolvidos são dados sísmicos pré-empilhamento que possui a estrutura de uma imagem 2D com tons de cinza, o que significa possuir apenas um canal de cor. Neste estudo, para treinamento dos modelos foram extraídos recortes das imagens de entrada com tamanho de 96 *pixels* de altura por 96 *pixels* de largura, contudo as predições foram realizadas com a imagem inteira.

Ao desenvolver uma rede convolucional é necessário informar alguns parâmetros essenciais ao instanciar uma camada convolucional como número de filtros, dimensão do filtro em *pixels*, deslocamento do filtro em *pixels* e o preenchimento de borda em *pixels*. Quando uma imagem é processada por uma camada convolucional, esta sofre uma perda no seu volume. Para preservar exatamente o tamanho espacial do volume de entrada é necessário utilizar o hiperparâmetro de preenchimento de bordas. O volume de saída de uma camada convolucional é calculada pela equação:

$$(W - F + 2P)/S + 1 \tag{4.1}$$

onde W representa o tamanho do campo receptivo da imagem, F representa o tamanho do filtro, P representa o preenchimento de zeros nas bordas e S representa o deslocamento.

4.2.1 CNN com três camadas

Baseado nos trabalhos dos autores Flamary [119] e Xu e colaboradores [120], os quais obtiveram sucesso tanto na reconstrução de imagens naturais como em astronomia, esta pesquisa propôs um modelo que consiste em uma rede totalmente convolucional de três camadas (figura 4.22). A primeira camada é composta por 32 filtros de dimensão 11×11 *pixels*, deslocamento do filtro igual a um, função de ativação *sigmoide* e sem preenchimento de borda, tornando a dimensão da imagem de entrada reduzida. A segunda camada consiste em 16 filtros de dimensão 5×5 *pixels*, com os mesmos hiperparâmetros da camada anterior. Já a terceira camada consiste em um filtro 1×1 *pixel* com função de ativação linear, cuja objetivo é compor a imagem reconstruída pelo modelo. A implementação da arquitetura foi realizada baseado nas especificações dos trabalhos citados, sem a utilização de um código pré-existente, contudo, houveram modificações no número de camadas convolucionais, hiperparâmetros e função de ativação. A dimensão dos filtros foi reduzida devido a dimensão da imagem de entrada de 96 *pixels* de altura por 96 *pixels* de largura.

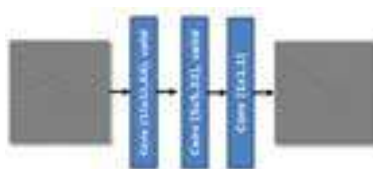


Figura 4.22: Arquitetura CNN com três camadas

4.2.2 *U-Net*

Os experimentos desta pesquisa foram realizados com uma arquitetura chamada *U-Net* [98], conhecida em problemas de segmentação de imagens. Este modelo também foi utilizado por Mandelli [97] para interpolação e *denoising* em dados sísmicos sintéticos pré-empilhamento com ruídos aleatórios. Diante do exposto, a seleção desta arquitetura foi motivada pela grande capacidade dos Autocodificadores Convolucionais em aprender representações compactas de dados e pela forte eficiência computacional na reconstrução de dados corrompidos. O modelo empregado é apresentado na figura 4.23 e é composto de três camadas de codificação, uma camada central e três camadas de decodificação. Em cada uma das camadas do codificador, três operações são executadas de forma sequencial, nomeadamente; convolução, função de ativação não linear e camadas de *pooling*. O número de filtros empregados nas camadas convolucionais é 32, 64, e 128; com filtro de dimensão 3×3 *pixels*, função de ativação tangente hiperbólica e camada de *max-pooling* com hiperparâmetro de dimensão de 2×2 , reduzindo a imagem pela metade a cada camada.

A camada central é composta de um operador de convolução com 256 filtros de dimensão 3×3 *pixels* e função de ativação tangente hiperbólica. Enquanto que, as camadas de decodificação são compostas pela seguinte sequência de operações: a primeira consiste em um operador de deconvolução que realiza um *upsampling* da imagem de entrada com fator de dois. Após essa operação, os mapas de características resultantes são concatenados com a saída da camada respectiva na etapa de codificação (*skip connection*) e, finalmente, é realizada uma operação de convolução com tamanho do dimensão do filtro 3×3 *pixels*.

Este conjunto de operações é executado em cada uma das camadas do decodificador. Embora a arquitetura *U-Net* tenha sido desenvolvida, originalmente, para problemas de segmentação de imagens médicas, houve adaptações quanto ao número dos filtros em relação ao artigo [98], passando de 64, 128, 256 para 32, 64 e 128 a fim de reduzir a quantidade total do número de parâmetros. De igual modo, a função de ativação foi modificada de *RELU* para tangente hiperbólica uma vez que o dado de entrada foi escalonado entre -1 e 1 e a saída da rede tornou-se linear. O desenvolvimento deste modelo nesta pesquisa, foi realizado a partir de implementação da arquitetura original retirada de um repositório do *github*.

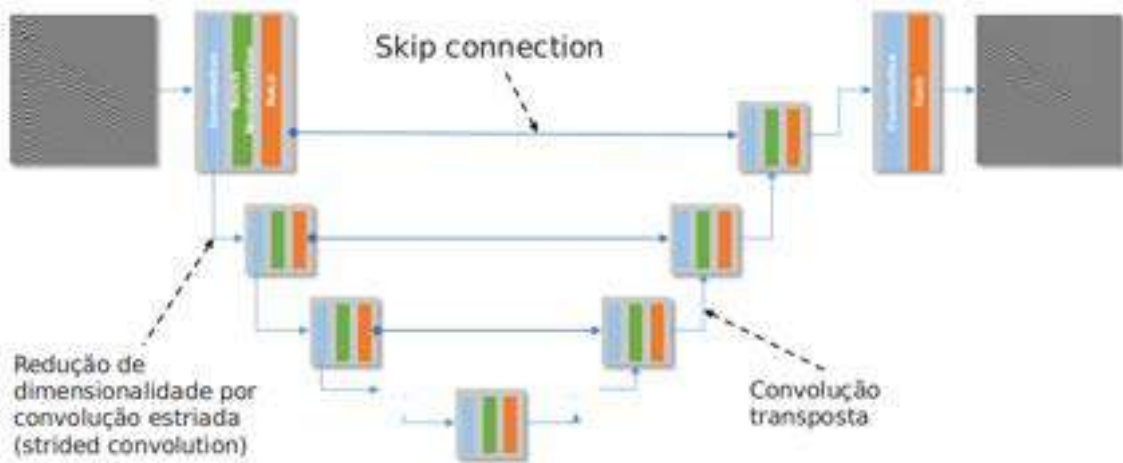


Figura 4.23: Arquitetura *U-Net*

4.2.3 Comparação entre os modelos

Considerando relevante a comparação entre os modelos, é apresentado na tabela 4.2, a seguir, o total de parâmetros treináveis e as camadas convolucionais contidas nos modelos CNN com três camadas *U-Net*. Observa-se a diferença na quantidade de parâmetros treináveis entre os dois modelos, uma vez que o modelo CNN com três camadas possui 59.074 parâmetros enquanto o *U-Net* possui 1.925.025 parâmetros, ou seja, uma diferença de 32 vezes a mais parâmetros. Este número pode ser melhor justificado uma vez que o *U-Net* trata-se de uma arquitetura Autocodificador Convolutacional com 2 (dois) blocos de decodificação, uma camada central e dois (2) blocos de codificação. Cada bloco contém 2 (duas) camadas convolucionais, totalizando 18 camadas convolucionais enquanto o modelo CNN, com três camadas com 59.074, possui apenas três camadas convolucionais.

Tabela 4.2: Comparação de parâmetros treináveis entre os modelos CNN com três camadas e *U-Net*

Modelos	Parâmetros Treináveis	Camadas Convolucionais
CNN com três camadas	59.073	Conv2D: 4x11x11 Conv2D: 32x5x5 Conv2D: 1x1x1
<i>U-Net</i>	1.925.025	Conv2D: 32x3x3 Conv2D: 32x3x3 - Conv2D: 64x3x3 Conv2D: 64x3x3 - Conv2D: 128x3x3 Conv2D: 128x3x3 - Conv2D: 256x3x3 Conv2D: 256x3x3 - Conv2D_t: 128x2x2 Conv2D: 128x3x3 Conv2D: 128x5x5 - Conv2D_t: 64x2x2 Conv2D: 64x3x3 Conv2D: 64x3x3 - Conv2D_t: 32x2x2 Conv2D: 32x3x3 Conv2D: 32x3x3 - Conv2D: 1x1x1

4.3 Procedimentos experimentais

Os experimentos realizados neste trabalho foram divididos em análise quantitativa e qualitativa entre dados de uma mesma aquisição, assim como análise qualitativa entre dados de diferentes aquisições. Já em análise quantitativa, os modelos foram treinados com dados sísmicos pré-empilhamento sintéticos e reais contendo ruídos aleatórios, lineares e *swell*.

Cada dado sísmico pré-empilhamento foi dividido em conjunto de tiros de treinamento e conjunto de tiros de testes. Os modelos CNN com três camadas e *U-Net* foram treinados com os respectivos dados e avaliados através das funções de avaliação PNSR e MSE nos dados; da mesma maneira, treinamento e testes de um mesmo dado sísmico pré-empilhamento.

Quanto a análise qualitativa entre sísmicas de mesma aquisição, foram selecionados tiros do conjunto de testes e plotados: (I) o tiro com ruído, (II) tiro sem ruído, (III) diferença entre o tiro com e sem ruído, (IV) resultado da predição do modelo, (V) diferença entre o tiro com ruído e a predição e a (VI) diferença entre o tiro sem ruído e a predição. Esse procedimento foi realizado com dados sintéticos com ruídos aleatórios e lineares e, ainda, com dados reais com ruído *swell*.

Tratando-se da análise qualitativa entre sísmicas de diferentes aquisições, o objetivo do experimento foi o de verificar a capacidade de generalização dos modelos. A partir da exposição dos modelos (seção 4.2), o treinamento e a predição foram realizados com dados sísmicos pré-empilhamento de diferentes aquisições. Os primeiros testes foram executados com dados contendo ruído *swell*. O modelo foi treinado com dado sísmico pré-empilhamento real da bacia do Espírito Santo e a predição executada com dado real pré-empilhamento da Bacia de Santos. Também foram executados testes de dados sintéticos com a adição do ruído *swell* retirada do dado sísmico pré-empilhamento real da bacia do Espírito Santo. Por outro lado, a predição foi executada com o dado sísmico pré-empilhamento real da bacia do Espírito Santo. Em seguida, o modelo foi treinado com dado sísmico pré-empilhamento real da bacia do Espírito Santo e a predição executada no dado sísmico sintético com adição de ruído *swell*, retirada do dado sísmico real dessa mesma aquisição. Por fim, foram executados testes com ruídos lineares. Nesse contexto, o modelo foi treinado com dados sísmicos sintéticos gerados com Modelo de Velocidade M1 e predição executada com dados sintéticos gerados pelo Modelo de Velocidade M2; o teste inverso também foi realizado.

4.3.1 Pré-processamento

1. Escalonamento: A proposta de pré-processamento para contornar o problema dos diferentes níveis de ruído entre os tiros foi o escalonamento pelo máximo absoluto do dado sísmico, no qual cada tiro é dividido pelo valor máximo do dado sísmico. A formulação utilizada para essa normalização pode ser vista na equação 4.2.

$$ImagemEscalonada = \frac{(ImagemTiros)}{MaxAbs(Sismica)} \quad (4.2)$$

2. Dimensionamento da entrada: Devido a diferentes dimensões das sísmicas apresentadas, foi realizado recorte de partes da imagem com dimensão de 96 x 96 *pixels* dos tiros pré selecionados para treinamento dos modelos.

4.3.2 Divisão dos Dados

Os dados foram divididos em conjunto de treinamento e teste. Os dados sintéticos foram divididos em uma partição de 70% para treinamento e 30% para teste, já os dados reais foram divididos em uma partição de 30% para treinamento e 70% para teste.

4.3.3 Treinamento

Todos os modelos utilizados neste trabalho possuem pesos e bias treináveis, que podem ser representados como um conjunto de matrizes como mostrado a seguir:

$$\theta = \{W_1, W_2, \dots, W_n, b_1, b_2, \dots, b_n\} \quad (4.3)$$

onde n é o número de camadas, $W_i = (i = 1, 2, \dots, n)$ denota os pesos de cada camada e $b_i = (i = 1, 2, \dots, n)$, seus respectivos bias.

Com o objetivo de atingir a melhor solução para o problema, torna-se necessário atualizar esses pesos, sendo isso realizado através da minimização do erro da função custo entre os valores preditos e reais. A função custo selecionada para treinamento dos modelos foi a A L1 e a sua formulação pode ser representada por

$$L1 = \frac{1}{N} \sum_{i=1}^M |y_i - \hat{y}_i| \quad (4.4)$$

onde N é o total de amostras, y_n é o valor real da n -ésima amostra e \hat{y}_n é o valor predito da mesma amostra. Esta função custo foi selecionada em função de sua maior robustez em comparação à função A L2, uma vez que são resistentes a valores discrepantes nos dados. A L1 permite igual ênfase a todas as observações, em contraste à A L2 que eleva ao quadrado os resíduos proporcionando mais peso aos resíduos maiores, ou seja, valores extremos nos quais os valores previstos estão longe das observações reais.

Para os experimentos desta pesquisa, os parâmetros da rede são atualizados usando-se o algoritmo de retro-propagação proposto por Lecun ([121], [122]) e a função custo é otimizada utilizando o algoritmo Adam [123] com taxa de aprendizado de 0.001. Por fim, os modelos foram treinados com 800 épocas.

A tabela 4.3 apresenta o tempo de treinamento para cada modelo e os dados utilizados. Nota-se que, de forma geral, o modelo *U-Net* levou o dobro do tempo de treinamento em relação aos treinamentos do modelo CNN com três camadas.

Tabela 4.3: Horas de treinamento dos modelos e dos dados

Modelo	Identificação Dado	Horas treinamento
CNN com três camadas	sinetico_M1_aleatorio10db	12
	sinetico_M1_aleatorio15db	12
	sinetico_M1_aleatorio20db	12
	sinetico_M1_linear	13
	sinetico_M2_linear	13
	real_ES_FK_swell	14
	real_ES_FK_mute_topo_swell	14
	real_ES_passa_banda_swell	14
<i>U-Net</i>	sinetico_M1_aleatorio10db	24
	sinetico_M1_aleatorio15db	24
	sinetico_M1_aleatorio20db	24
	sinetico_M1_linear	20
	sinetico_M2_linear	20
	real_ES_FK_swell	27
	real_ES_FK_mute_topo_swell	27
	real_ES_passa_banda_swell	30

4.4 Resumo do capítulo

Neste capítulo, foram detalhados os tipos de dados utilizados neste trabalho bem como os tipos de ruídos existentes, os experimentos realizados como pré-processamento e a divisão dos dados para treinamento dos modelos. No próximo capítulo, serão apresentados os resultados dos experimentos realizados assim como as discussões.

Capítulo 5

Resultados e Discussões

Este capítulo apresenta e discute os resultados obtidos com o treinamento dos modelos CNN com três camadas e *U-Net* nos dados sísmicos sintéticos e reais com os ruídos aleatórios, lineares e *swell*, em comparação aos dados sísmicos com e sem ruídos. Neste trabalho, foram aplicadas duas técnicas para remoção de ruído a saber: a primeira trata-se de um modelo de rede neural com três (3) camadas convolucionais, enquanto a refere-se ao modelo treinado foi o modelo de rede neural *U-Net* com seis (6) camadas convolucionais, três (3) camadas para codificação e três (3) camadas para decodificação.

5.1 Estatísticas de Validação

Nesta seção, será descrita a metodologia adotada para análise e treinamento dos modelos. Primeiramente, serão apresentados os métodos para avaliação dos modelos, seguido da descrição dos experimentos realizados.

Considerando a necessidade de comparação dos resultados, utilizou-se funções de avaliação nomeadamente a razão sinal de pico/ruído (PSNR) e o erro quadrático médio (MSE).

1. PSNR: Para fins de avaliação, foi utilizada relação sinal / ruído (S / N) que é definida como a razão entre a variação da coleta original e o ruído, em que ruído é a diferença entre o sinal bruto e o sinal limpo. O S / N é definido pela fórmula descrita a seguir, na qual os dados sísmicos corrompidos são representados por S e sua amostra limpa é representada por N.

$$PSNR = 10 \log_{10} \frac{\sum_{i=1}^N s_i}{\sum_{i=1}^N s_i - q_i} \quad (5.1)$$

2. MSE: O erro quadrático médio (MSE) é definido como a média da diferença quadrada do elemento entre o sinal previsto e o sinal verdadeiro, calculada

como:

$$MSE = \frac{1}{N} \sum_{i=1}^M (y_i - \hat{y}_i)^2 \quad (5.2)$$

3. Espectro F-K O espectro F-K: para avaliação qualitativa será utilizado o Espectro F-K

5.2 Análise Quantitativa

Nesta seção, será apresentado o desempenho dos modelos CNN com três camadas e *U-Net* propostos na tarefa de atenuação de ruído por meio das funções de avaliação *Peak signal-to-noise ratio* (PSNR), entre os tiros ruidosos utilizados como entrada no treinamento e os resultados das previsões dos modelos. Esses resultados terão como base de comparação o PSNR calculado a partir dos tiros ruidosos e limpos dos dados sísmicos utilizados. Também foi utilizado a função de avaliação MSE para comparação do erro obtido entre os modelos. Neste caso, o resultado foi obtido entre os tiros limpos utilizados como saída no treinamento e os resultados das previsões dos modelos.

Nas tabelas 5.1, 5.2 e 5.3 são apresentados os resultados da média e desvio padrão dos tiros, dos conjuntos de treinamento e testes das funções de avaliação PSNR e MSE nos dados sísmicos com ruídos aleatórios, lineares e *swell*.

5.2.1 Ruídos Aleatórios

Os resultados de atenuação de ruídos aleatórios calculados por meio das funções de avaliação MSE e PSNR no conjunto de treinamento e testes dos dados sísmicos contendo ruídos aleatórios com as intensidades de 10, 15 e 20dB, são detalhados na tabela 5.1. Observa-se o mesmo resultado de média e desvio padrão de PSNR dos modelos CNN com três camadas e *U-Net* em relação ao resultado base, com valores de PSNR de 71 para 10dB, 75 para 15dB e 77 para 20dB nos conjuntos de treinamento e teste. Embora as semelhanças nos resultados de PSNR, nota-se melhor resultado de média e desvio padrão do MSE do modelo *U-Net* em todos os dados sísmicos testados.

Tabela 5.1: Resultados dos modelos propostos para ruído aleatório com intensidades de 10, 15 e 20dB - PSNR e MSE

MODELO	DADO	PSNR	MSE
Dados de Treino			
ruidoso / limpo	sinetico_M1_aleatorio10dB	71.63 / 3.43E-03	-
	sinetico_M1_aleatorio15dB	75.15 / 2.93E-03	-
	sinetico_M1_aleatorio20dB	77.65 / 73.91E-03	-
CNN com três camadas	sinetico_M1_aleatorio10dB	71.63 / 3.50E-03	9.70e-06 / 4.46e-07
	sinetico_M1_aleatorio15dB	75.15 / 3.10E-03	5.21e-06 / 2.93e-07
	sinetico_M1_aleatorio20dB	77.65 / 4.00E-03	3.66e-06 / 2.03
<i>U-Net</i>	sinetico_M1_aleatorio10dB	71.63 / 3.44E-03	7.45e-07 / 1.16e-07
	sinetico_M1_aleatorio15dB	75.15 / 2.97E-03	4.58e-07 / 5.68e-08
	sinetico_M1_aleatorio20dB	77.65 / 3.95E-03	6.58e-07 / 5.86e-08
Dados de Teste			
ruidoso / limpo	sinetico_M1_aleatorio10dB	71.63 / 3.11E-03	-
	sinetico_M1_aleatorio15dB	75.15 / 3.20E-03	-
	sinetico_M1_aleatorio20dB	77.65 / 3.30E-03	-
CNN com três camadas	sinetico_M1_aleatorio10dB	71.63 / 3.13E-03	9.69e-06 / 4.29e-07
	sinetico_M1_aleatorio15dB	75.15 / 3.58E-03	5.20e-06 / 2.92e-07
	sinetico_M1_aleatorio20dB	77.65 / 3.32E-03	3.66e-06 / 2.25e-0
<i>U-Net</i>	sinetico_M1_aleatorio10dB	71.63 / 3.13E-03	8.90e-07 / 1.02e-07
	sinetico_M1_aleatorio15dB	75.15 / 3.35E-03	4.96e-07 / 5.71e-08
	sinetico_M1_aleatorio20dB	77.65 / 3.27E-03	6.70e-07 / 6.50e-0

5.2.2 Ruídos Lineares

Os resultados de atenuação de ruído linear calculados por meio das funções de avaliação MSE e PSNR no conjunto de treinamento e testes são apresentados tabela 5.2. Observa-se o mesmo resultado de média e desvio padrão de PSNR dos modelos CNN com três camadas e *U-Net* em relação ao resultado base, com valores de PSNR de 94 para o dado sísmico M1 e 93 para o dado sísmico M2, ambos nos conjuntos de treinamento e teste. Embora as semelhanças nos resultados de PSNR, nota-se melhor resultado de média e desvio padrão do MSE do modelo *U-Net* em todos os dados sísmicos testados.

5.2.3 Ruídos *Swell*

Os resultados de atenuação de ruído *swell* calculados por meio das funções de avaliação MSE e PSNR no conjunto de treinamento e testes são apresentados na tabela

Tabela 5.2: Resultados dos Modelos propostos para ruídos lineares nos dados sísmicos M1 e M2 - PSNR e MSE

MODELO	DADO	PSNR	MSE
Dados de Treino			
ruidoso / limpo	sinetico_M1_linear	94.62 / 0.20	-
	sinetico_M2_linear	93.39 / 0.28	-
CNN com três camadas	sinetico_M1_linear	94.70 / 0.21	1.38E-06 / 5.49E-07
	sinetico_M2_linear	93.65 / 0.29	6.63E-08 / 5.75E-08
<i>U-Net</i>	sinetico_M1_linear	94.58 / 0.20	2.42E-06 / 5.02E-07
	sinetico_M2_linear	93.39 / 0.28	1.03E-07 / 8.63E-08
Dados de Teste			
ruidoso / limpo	sinetico_M1_linear	94.59 / 0.19	-
	sinetico_M2_linear	93.38 / 0.28	-
CNN com três camadas	sinetico_M1_linear	94.68 / 0.20	1.40E-06 / 6.70E-07
	sinetico_M2_linear	93.64 / 0.27	1.10E-07 / 1.74E-07
<i>U-Net</i>	sinetico_M1_linear	94.56 / 0.18	2.53E-06 / 5.26E-07
	sinetico_M2_linear	93.39 / 0.26	1.58E-07 / 1.88E-07

5.3. Nota-se a superioridade dos resultados do PSNR do modelo *U-Net* nos dados sísmicos reais com ruído *swell* com filtragem passa-banda e filtro F-K (com e sem *mute* no topo) em comparação aos resultados de PSNR entre os tiros ruidosos e limpos. Verifica-se que para os dados de treino e teste, os valores de PSNR obtidos pelo modelo *U-Net* alcançou o mesmo resultado de PSNR entre os tiros ruidosos e limpos. Ainda que o modelo CNN com três camadas tenha menor número de parâmetros que o modelo *U-Net*, este alcançou resultado de PSNR próximo ao resultado base, com uma diferença de apenas 1% de PSNR na média. Com relação aos resultados do MSE, observa-se que o modelo *U-Net* obteve também melhores resultados de média e desvio padrão em relação aos resultados do modelo CNN com três camadas. Para o dado sísmico sintético M2 com ruído *swell*. Relevante é considerar que, os modelos obtiveram os mesmos resultados em todos os testes realizados com dados de treino e de teste.

Tabela 5.3: Resultados dos modelos propostos para dado real com filtragem passa-banda - PSNR e MSE

MODELO	DADO	PSNR	MSE
Dados de Treino			
ruidoso / limpo	real_ES_passa_banda_swell	118.99 / 2.79	-
	real_ES_FK_swell	115.04 / 3.85	-
	real_ES_FK_mute_topo_swell	114.98 / 3.82	-
	sintetico_M2_swell	87.28 / 1.56E-01	-
CNN com três camadas	real_ES_passa_banda_swell	104.52 / 3.22	2.77e-06 / 1.73e-06
	real_ES_FK_swell	104.32 / 3.15	2.90E-06 / 1.82E-06
	real_ES_FK_mute_topo_swell	104.32 / 3.14	2.90e-06 / 1.82e-06
	sintetico_M2_swell	87.5 / 0.3	2.20E-06 / 3.54E-07
<i>U-Net</i>	real_ES_passa_banda_swell	118.74 / 2.84	6.12e-09 / 1.06e-08
	real_ES_FK_swell	116.43 / 3.50	8.00E-08 / 7.25E-08
	real_ES_FK_mute_topo_swell	116.41 / 3.48	7.31e-09 / 1.29e-08
	sintetico_M2_swell	87.28 / 1.60E-01	7.90E-08 / 3.48E-08
Dados de Teste			
ruidoso / limpo	real_ES_passa_banda_swell	119.14 / 2.91	-
	real_ES_FK_swell	115.12 / 3.99	-
	real_ES_FK_mute_topo_swell	115.06 / 3.96	-
	sintetico_M2_swell	87.30 / 2.10E-01	-
CNN com três camadas	real_ES_passa_banda_swell	104.37 / 3.02	2.81e-06 / 1.70e-06
	real_ES_FK_swell	104.17 / 2.95	8.76E-08 / 8.13E-08
	real_ES_FK_mute_topo_swell	104.17 / 2.95	2.93e-06 / 1.78e-06
	sintetico_M2_swell	87.52 / 0.41	2.21E-06 / 3.92E-07
<i>U-Net</i>	real_ES_passa_banda_swell	118.74 / 2.84	118.74 / 2.84
	real_ES_FK_swell	116.43 / 3.50	8.00E-08 / 7.25E-08
	real_ES_FK_mute_topo_swell	116.41 / 3.48	7.31e-09 / 1.29e-08
	sintetico_M2_swell	87.28 / 1.60E-01	7.90E-08 / 3.48E-08

5.3 Análise qualitativa entre sísmicas de mesma aquisição

Devido à superioridade do modelo *U-Net* nos resultados de PSNR e MSE na seção de Resultados Numéricos (seção 5.2), os resultados serão selecionados para uma análise qualitativa, no qual as imagens de tiros serão analisados bem como seus espectros de frequência. Para a realização deste experimento foram selecionados de forma aleatória alguns tiros pertencentes ao conjunto de testes nos dados sísmicos com ruídos aleatórios, lineares e *swell*. No conjunto de dados do ruído aleatório foi selecionado o tiro 115, no conjunto de dados com ruídos lineares foi selecionado o tiro 36 e para o conjunto de dados reais foi selecionado o tiro 6170. Para cada tiro selecionado são apresentadas as figuras do tiro ruidoso (utilizado como entrada do treinamento), tiro sem ruído (utilizado como saída do treinamento), diferença

entre entrada e saída, resultado da predição dos modelos propostos, diferença entre entrada e predição e a diferença entre a saída e a predição. Foi utilizado o programa *Seismic Unix* [111] para a geração das imagens dos tiros e espectros de frequência 2D utilizado para averiguação dos ruídos presentes nos dados sísmicos reais pré-empilhamento.

5.3.1 Ruídos Aleatórios

As figuras 5.1, 5.2, e 5.3 apresentam os resultados do tiro 115 do dados sísmicos M1 com ruídos aleatórios com intensidade de 10dB. Na figura 5.2(b) é apresentado o resultado da predição do modelo, nota-se a descontinuidade do sinal sísmico comparado com o tiro sem ruídos representado pela figura 5.1(a). Essa perda de sinal pode ser melhor observada através da diferença entre o tiro com ruído (saída) e a predição através da figura 5.2(c) no qual parte do sinal sísmico é encontrado na figura 5.3(c) com a diferença entre o tiro sem ruído e a predição.

dados sísmicos sintético M1 com ruído aleatório com intensidade de 10db

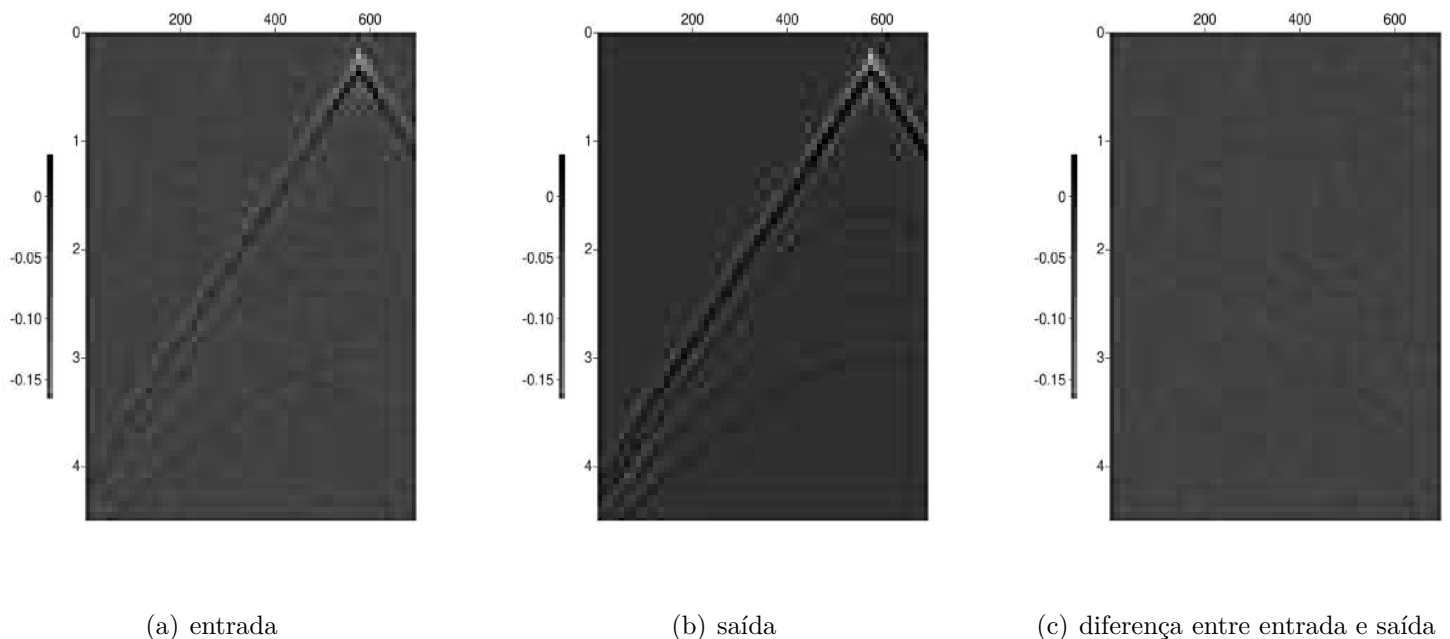
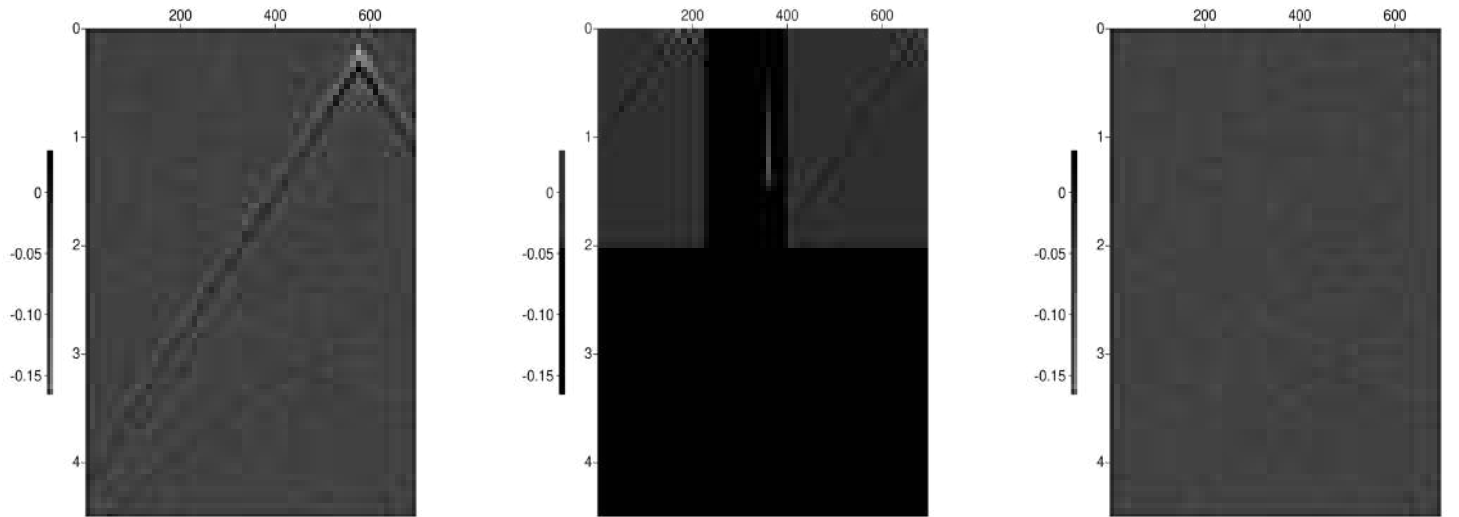


Figura 5.1: Resultado tiro 115 com ruído aleatório 10dB

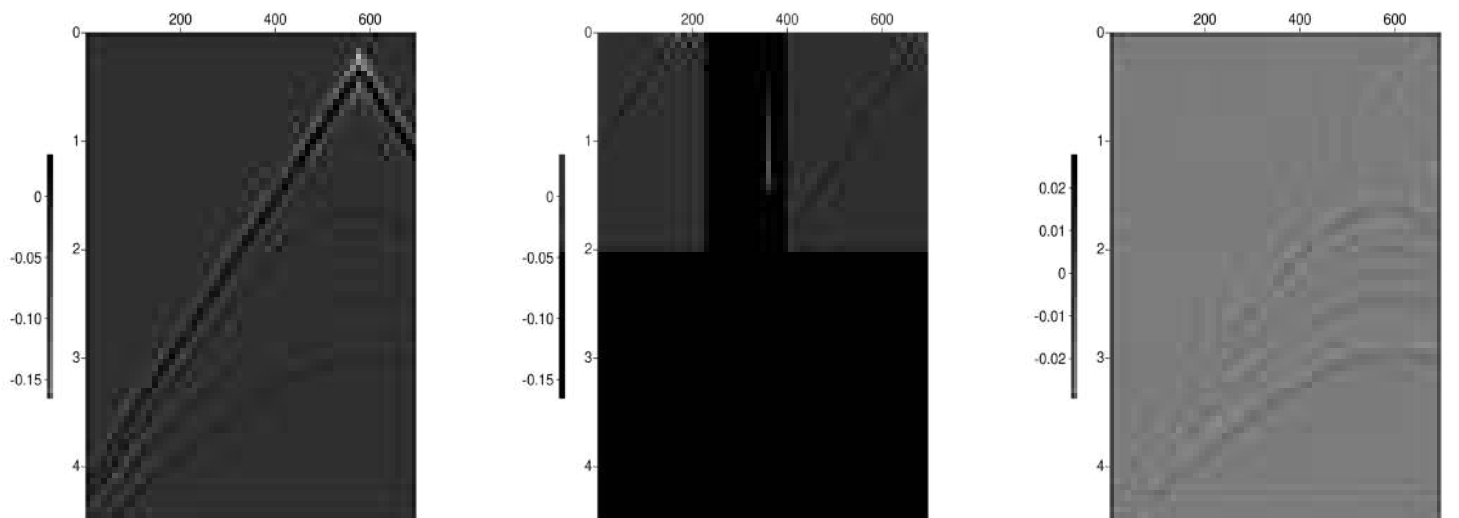


(a) entrada

(b) predição

(c) diferença entre entrada e predição

Figura 5.2: Resultado tiro 115 com ruído aleatório 10dB



(a) saída

(b) predição

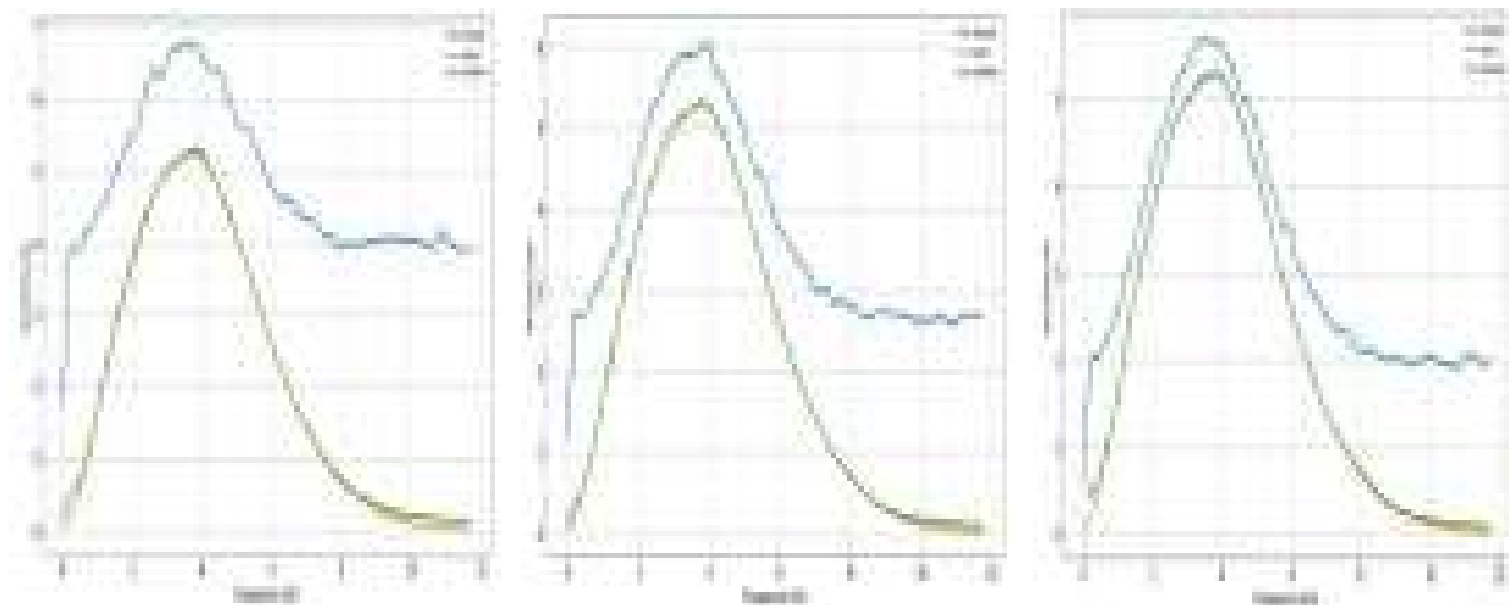
(c) diferença entre saída e predição

Figura 5.3: Resultado tiro 115 com ruído aleatório 10dB

Espectro de frequência ruído aleatório - 10, 15 e 20dB

Na figura 5.4 é apresentado o espectro de frequência dos tiros 115 de entrada, saída e predição (representada pelas cores azul, laranja e verde) da sísmica sintética com ruídos aleatórios para as intensidades de 10, 15 e 20dB. Verifica-se que os tiros

de saída e predição possuem o mesmo espectro de frequência uma vez que estão sobrepostas.



(a) Espectro fx 10dB

(b) Espectro fx 15dB

(c) Espectro fx 20dB

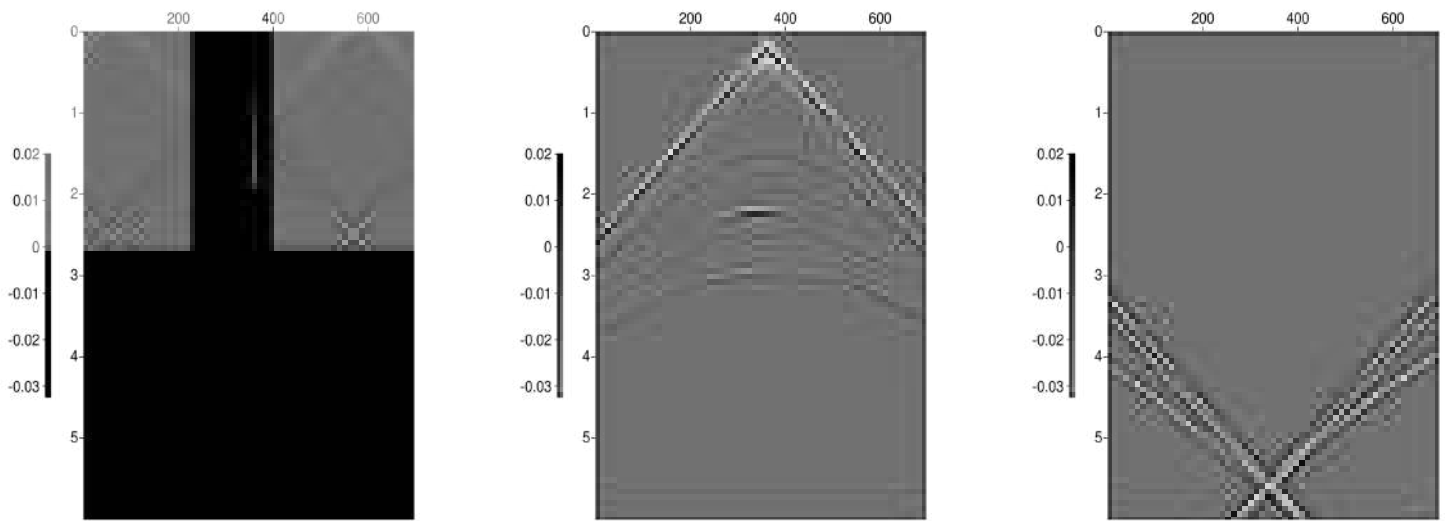
Figura 5.4: Espectro de frequência ruído aleatório 10, 15 e 20dB

5.3.2 Ruídos Lineares

Considerando a continuidade da análise, as figuras 5.5, 5.6 e 5.7 apresentam os resultados do tiro 36 do dado sísmico sintético com os ruídos lineares M1 e as figuras 5.8, 5.9 e 5.10 apresentam exemplo do tiro 36 do dado sísmico sintético com o ruído lineares modelo M2. Observa-se através da diferença entre entrada e predição representada pela figura 5.6(c) que os ruídos foram removidos, porém nota-se a presença de vestígios de ruídos e de sinal que podem ser melhor observados pela diferença entre a saída e predição representada pela figura 5.7(c), embora a sua escala seja na ordem de 10^{-4} em comparação as outras figuras com escala na ordem de 10^{-2} .

O mesmo comportamento pode ser visto nos resultados dos dados sísmicos sintéticos de ruídos lineares do modelo M2. Nota-se que através das figuras 5.9(c) e 5.10(c) que os ruídos não foram completamente removidos. Embora tenham ocorrido vestígios de ruído nos resultados dos modelos lineares, verificou-se que o ruído foi atenuado de forma mais eficaz do que os resultados de ruídos aleatórios, possibilitando-se considerar que o cenário com a sísmica sintética com ruídos aleatórios está em um nível de dificuldade maior, uma vez que este ruído sobrepõe o dado de forma significativa.

dados sísmicos sintéticos M1 com ruído linear

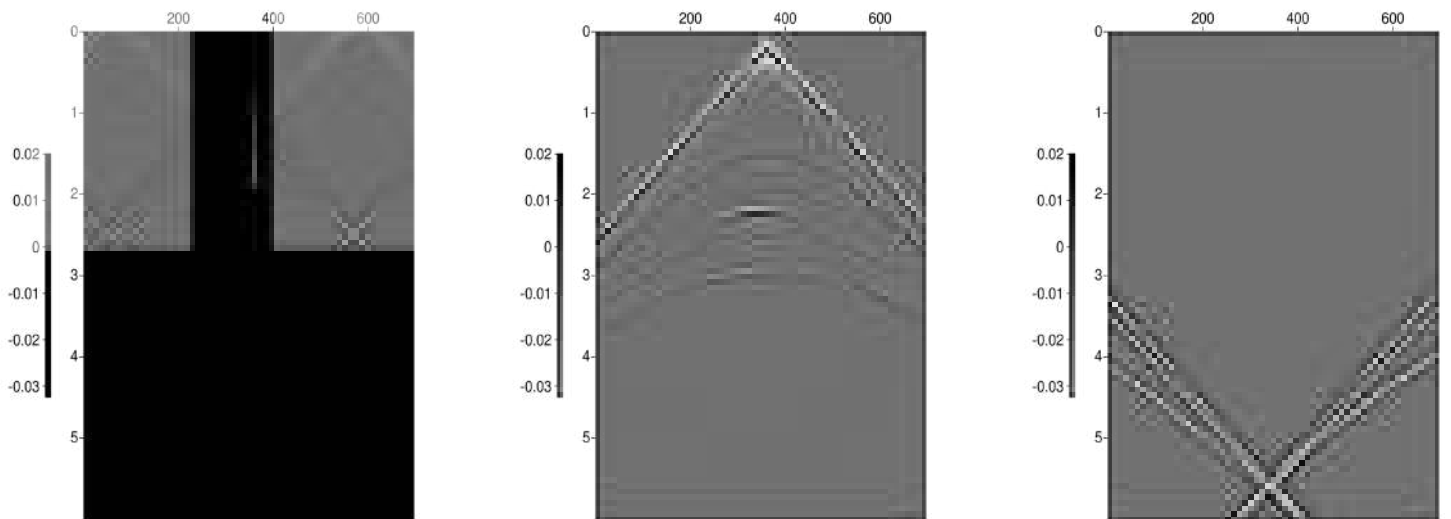


(a) entrada

(b) saída

(c) diferença entre entrada e saída

Figura 5.5: Resultado tiro 36 com ruído linear M1

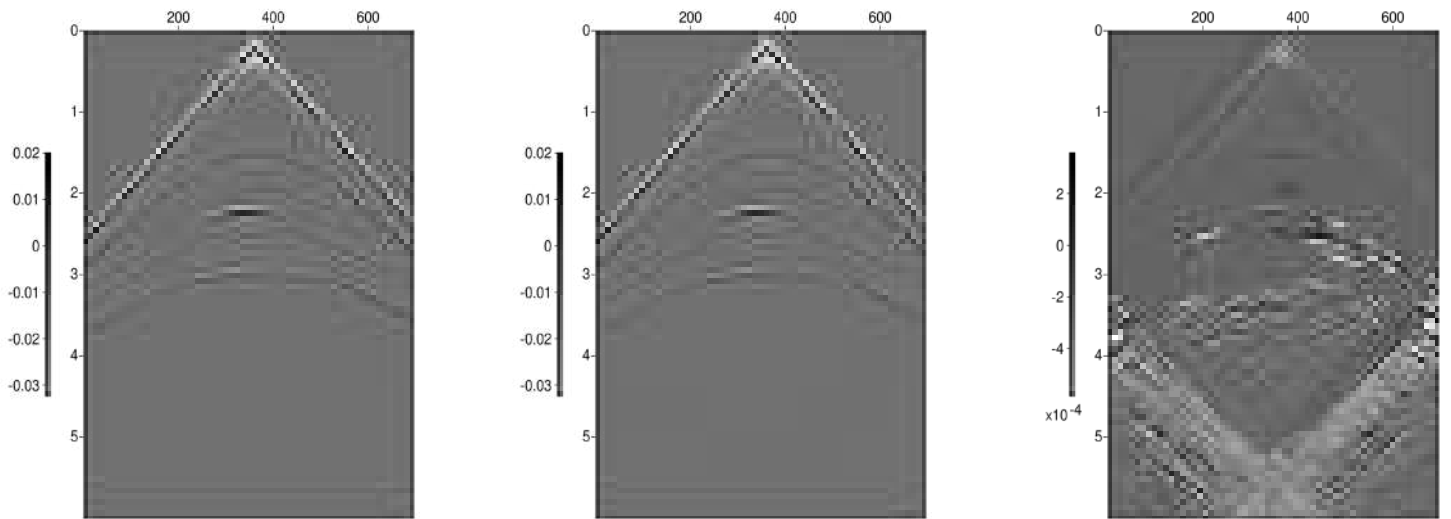


(a) entrada

(b) predição

(c) diferença entre entrada e predição

Figura 5.6: Resultado tiro 36 com ruído linear M1



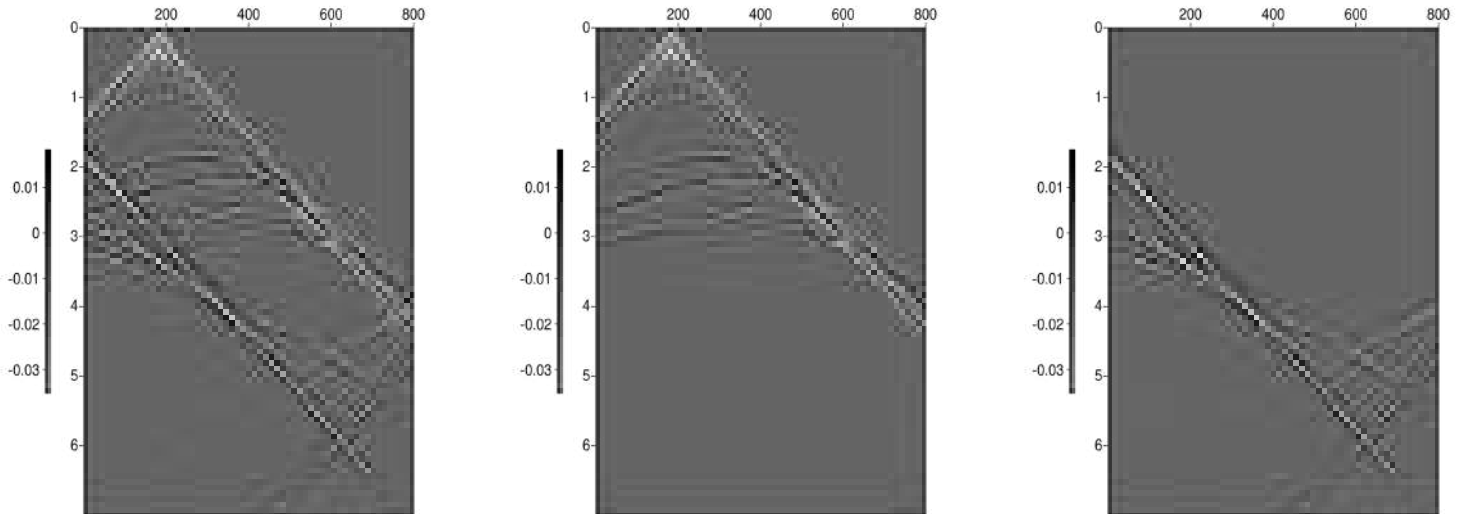
(a) saída

(b) predição

(c) diferença entre saída e predição

Figura 5.7: Resultado tiro 36 com ruído linear M1

Dado sísmico sintético M2 com ruído linear

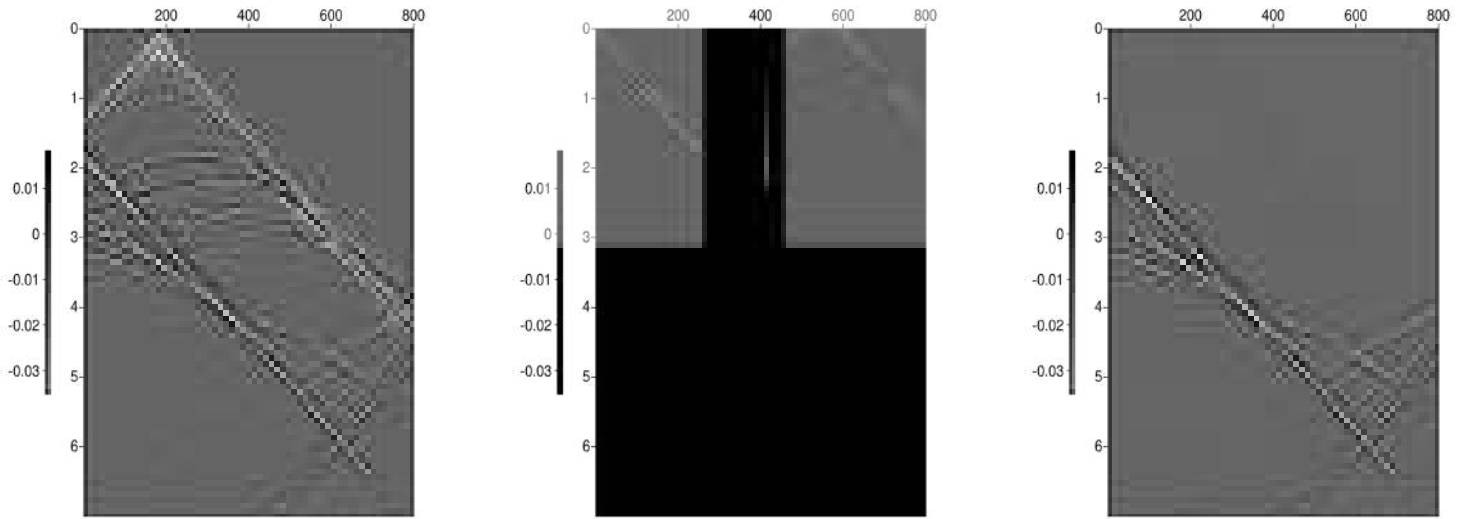


(a) entrada

(b) saída

(c) diferença entre entrada e saída

Figura 5.8: Resultado tiro 36 com ruído linear M2

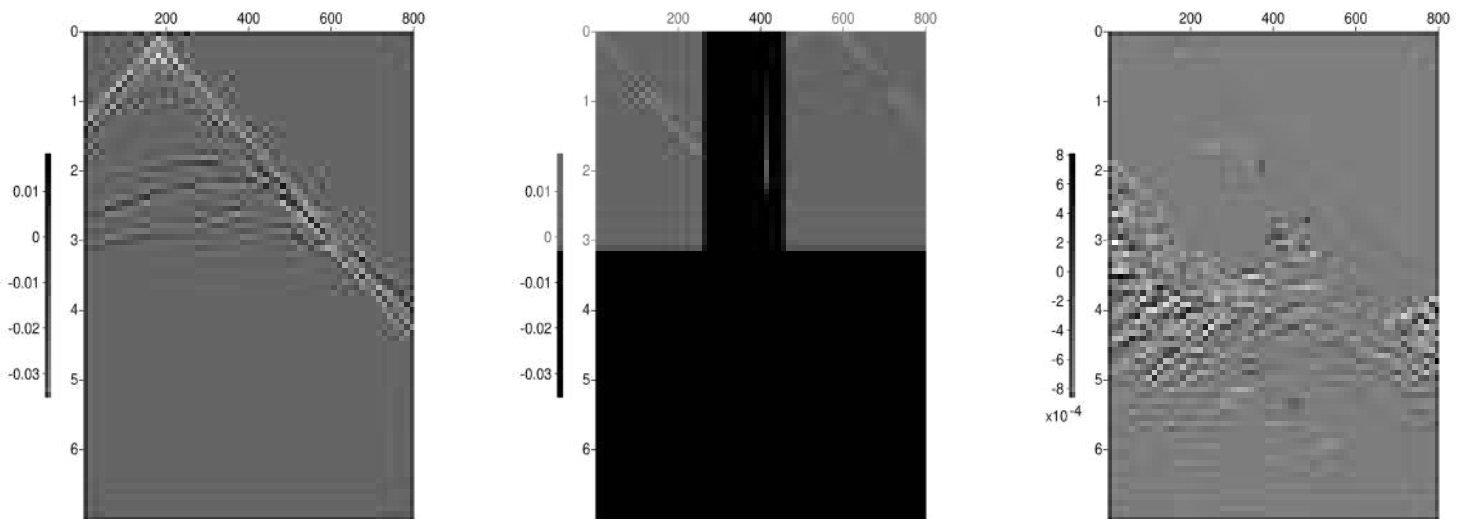


(a) entrada

(b) predição

(c) diferença entre entrada e predição

Figura 5.9: Resultado tiro 36 com ruído linear M2



(a) saída

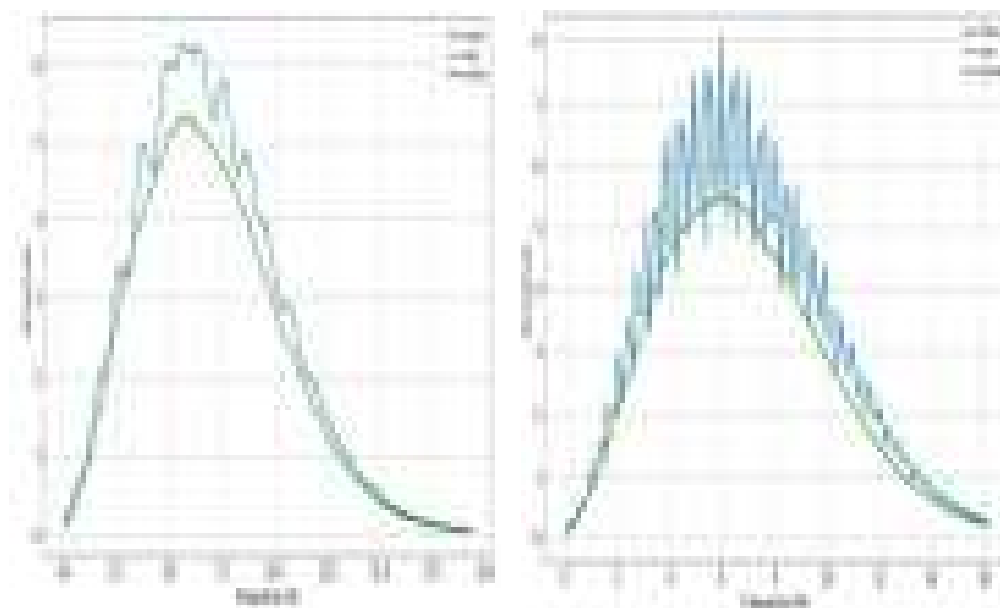
(b) predição

(c) diferença entre saída e predição

Figura 5.10: Resultado tiro 36 com ruído linear M2

Espectro de frequência ruído linear dos modelos de velocidade M1 e M2

Na figura 5.11 é apresentado o espectro de frequência dos tiros 115 de entrada, saída e predição (representada pelas cores azul, laranja e verde) das sísmicas sintéticas M1 e M2 com ruídos lineares. Verifica-se que os tiros de saída e predição possuem o mesmo espectro de frequência uma vez que estão sobrepostas.



(a) Espectro FX M1

(b) Espectro FX M2

Figura 5.11: Espectro de frequência ruído linear M1 e M2

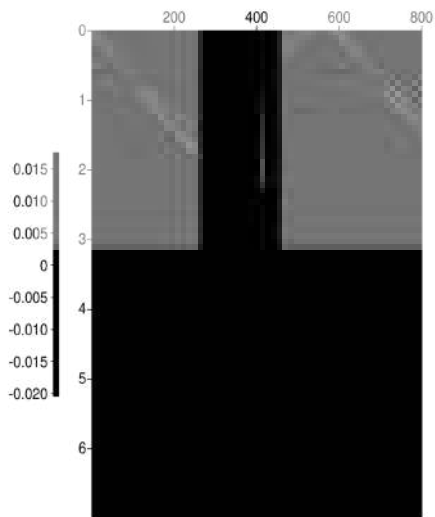
5.3.3 Ruído *Swell*

Nesta seção, serão analisados os resultados do modelo *U-Net* com o propósito de verificar sua capacidade de atenuação do ruído *swell* sem distorção dos sinais primários para dados reais e sintéticos. Segundo Zhao e colaboradores [124], uma solução com grande capacidade de atenuação de ruído precisa ser capaz de manter os sinais primários. Nos métodos convencionais, como a transformação no domínio da frequência, é comum que algum ruído residual remanescente permaneça para evitar prejudicar os sinais primários, em outras palavras, um algoritmo/solução com maior capacidade de atenuação de ruído tem maior probabilidade de prejudicar os sinais primários.

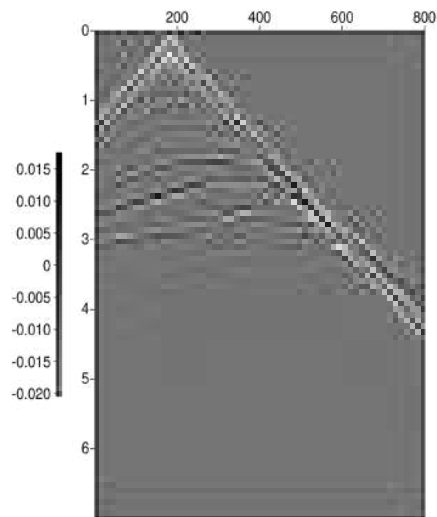
Dado sísmico sintético M2 com ruído *swell*

As figuras 5.12, 5.13 e 5.14 apresentam os resultados do tiro 36 do dado sísmico M2 com ruído *swell* extraído através da diferença entre os tiros com e sem ruídos do dado sísmico da bacia do Espírito Santo.

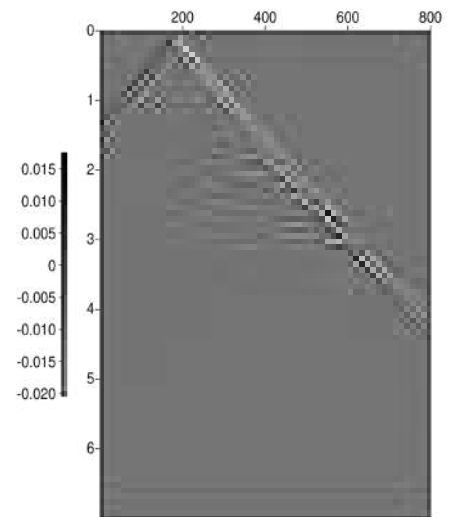
Na figura 5.13(b) é apresentado o resultado da predição do modelo, observando-se que, aparentemente, o ruído *swell* foi removido de forma efetiva quando comparado com a do tiro sem ruído na figura 5.12(b). Contudo, ao verificar a diferença entre o resultado da predição e o tiro sem ruído na figura 5.14(c), nota-se a perda de sinal, embora a sua escala seja menor que as outras figuras.



(a) entrada

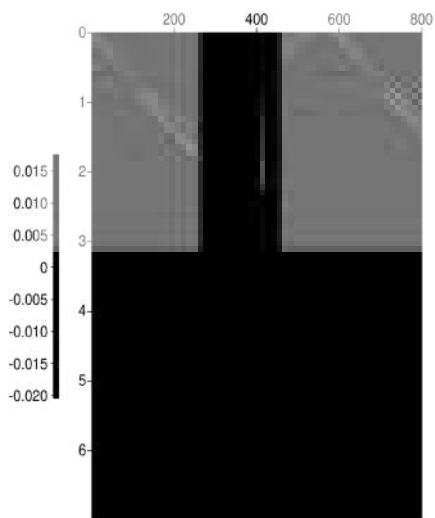


(b) saída

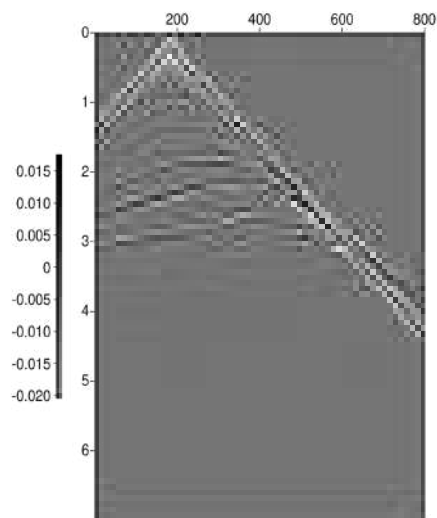


(c) diferença entre entrada e saída

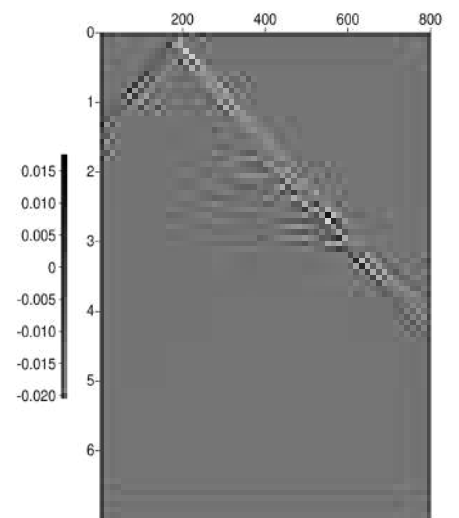
Figura 5.12: Resultado tiro 36 com ruído *swell* M2



(a) entrada

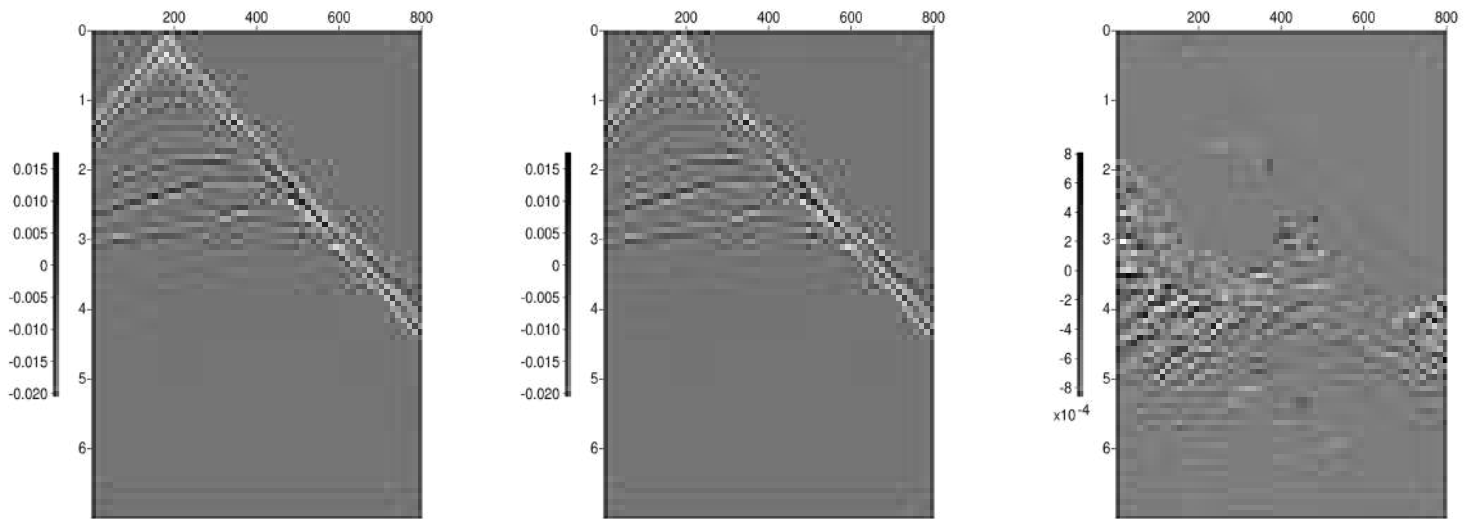


(b) predição



(c) diferença entre entrada e predição

Figura 5.13: Resultado tiro 36 com ruído *swell* M2



(a) saída

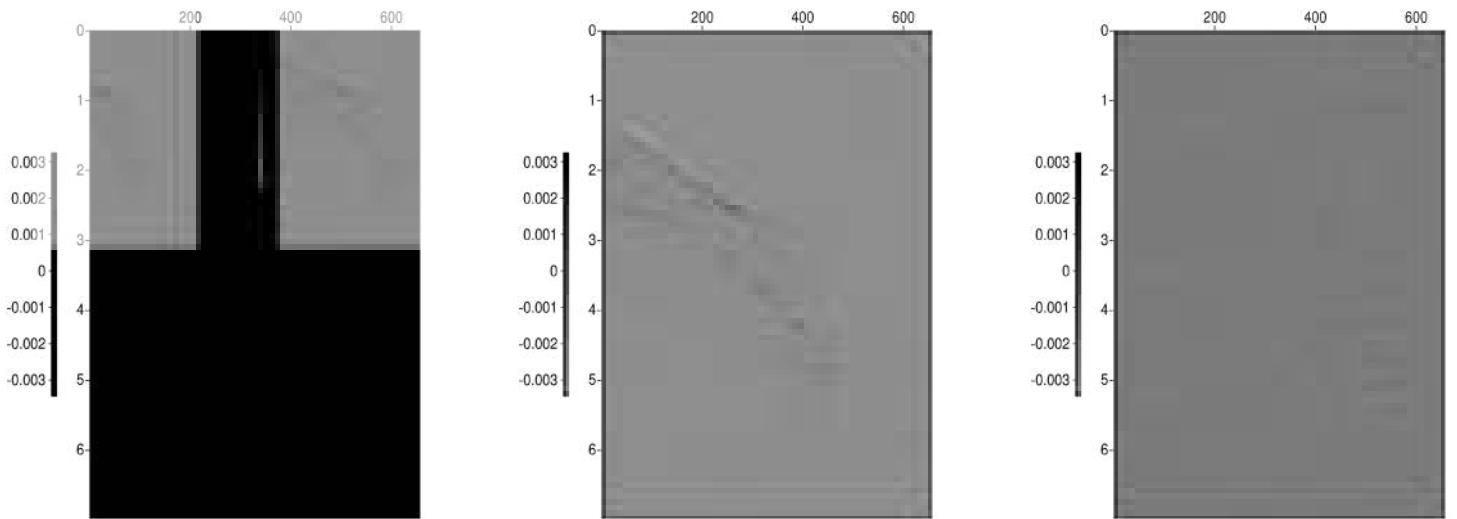
(b) predição

(c) diferença entre saída e predição

Figura 5.14: Resultado tiro 36 com ruído *swell* M2

Dado sísmico real da bacia do Espírito Santo

As figuras 5.35, 5.16 e 5.17 apresentam os resultados do tiro 6170 do dado sísmico de aquisição da bacia do Espírito Santo. Verifica-se na figura 5.16, que a remoção do ruído *swell* concentrado na baixa frequência foi removido de forma efetiva. Nota-se através das diferenças entre entrada e predição na figura 5.16(c) e através da diferença entre a saída e a predição na figura 5.17(c), no qual o sinal foi preservado. Esta informação também pode ser melhor observada através do espectro F-K na figura 5.18.

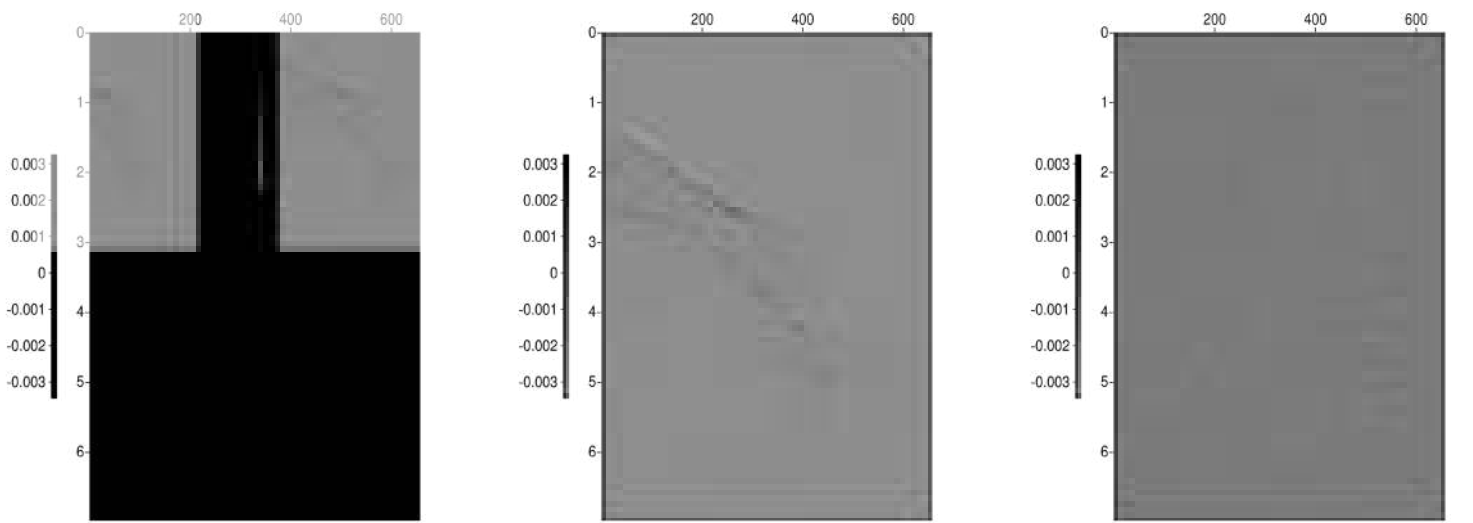


(a) entrada

(b) saída

(c) diferença entre entrada e saída

Figura 5.15: Resultado tiro 6170 com ruído *swell* com filtro passa-banda

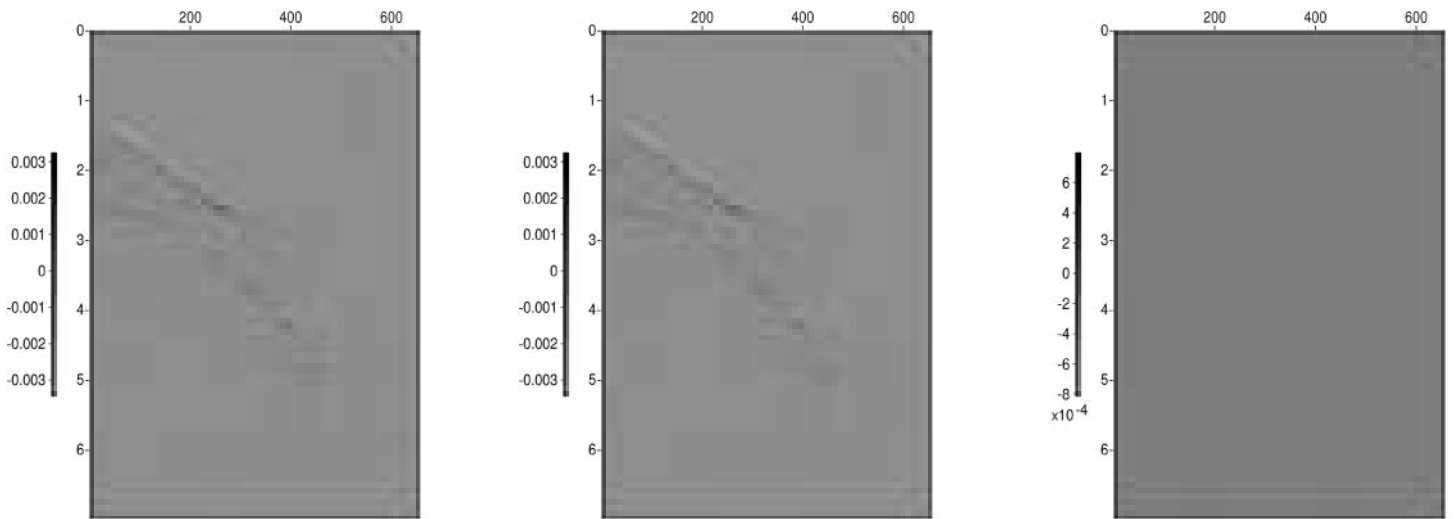


(a) entrada

(b) predição

(c) diferença entre entrada e predição

Figura 5.16: Resultado tiro 6170 com ruído *swell* com filtro passa-banda



(a) saída

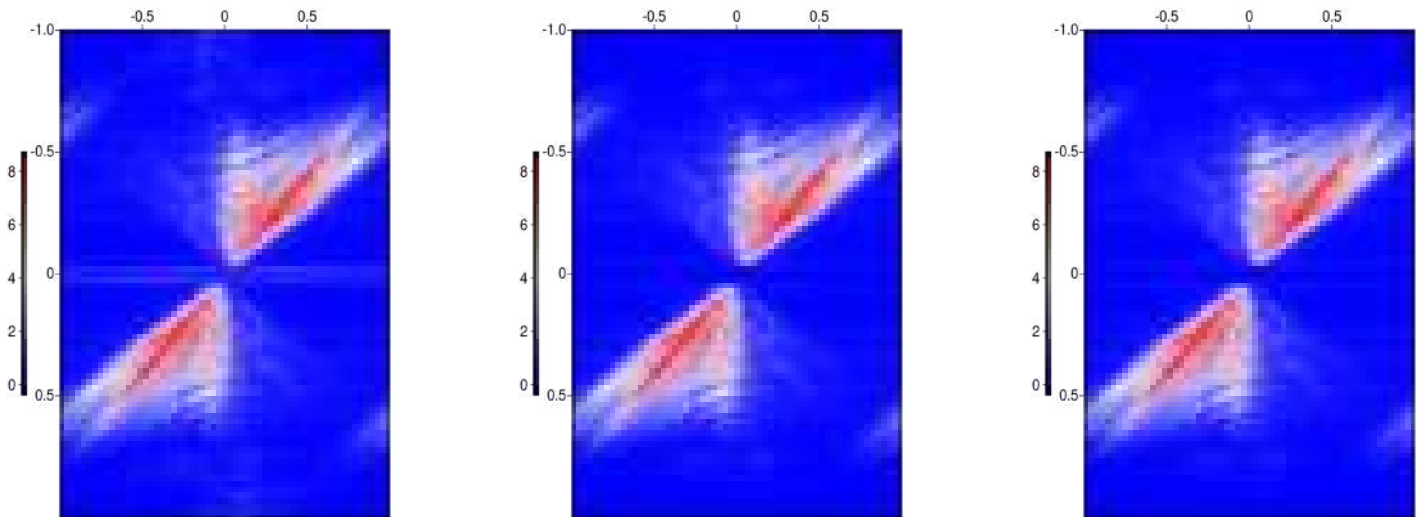
(b) predição

(c) diferença entre saída e predição

Figura 5.17: Resultado tiro 6170 com ruído *swell* com filtro passa-banda

Espectro F-K ruído *swell*

A figura 5.18 apresenta o espectro F-K do tiro 6170 de entrada, saída e predição. Verifica-se que os ruídos *swell* concentrados na baixa frequência foram removidos sem a remoção do sinal.



(a) entrada

(b) predição

(c) saída

Figura 5.18: Espectro F-K tiro 6170, ruído *swell* com filtro passa-banda

Espectro de frequência ruído *swell*

Na figura 5.19, é apresentado o espectro de frequência do tiros 115 de entrada, saída e predição (representada pelas cores azul, laranja e e verde) da sísmica real com ruídos *swell*. Verifica-se que os tiros de saída e de predição possuem o mesmo espectro de frequência uma vez que estão sobrepostas.

U-net



Figura 5.19: Espectro de frequência tiro 6170, ruído *swell* com filtro passa-banda

5.3.4 Dado sísmico por faixas de frequência

Neste experimento foram aplicados filtros passa-banda com as faixas de frequência 1-5Hz, 5-10Hz, 20-40Hz e maior que 60Hz do tiro 6170 do dado sísmico da aquisição do Espírito Santo. A figura 5.20(a) apresenta o tiro com ruídos *swell* (entrada), a figura 5.20(b) apresenta o resultado da predição do modelo *U-Net* e a figura 5.20(c) apresenta a diferença entre as figuras 5.20(a) e 5.20(b) que representa o ruído *swell* detectado atenuado pelo modelo *U-Net*.

Ao verificar cada faixa de frequência do tiro de entrada, observa-se que o ruído do *swell* está mais concentrado na faixa de baixa frequência, na faixa entre 0 a 5Hz. Verifica-se na faixa de frequência 0-5Hz da figura 5.20(b), a ausência de ruído *swell* nos dados, embora nota-se algum resíduo leve de ruído *swell* nos resultados atenuados. Por outro lado, ao observar as altas frequências, verifica-se que os sinais primários não foram prejudicados. Por fim, ao analisar a diferença entre a entrada e predição na faixa de frequência de 0-5Hz (figura 5.20(c)), verifica-se de que o ruído *swell* removido com o processo de atenuação. Conclui-se, assim, que o modelo *U-Net* foi capaz de remover efetivamente a maior parte do ruído de *swell* em cada faixa de frequência sem prejudicar os sinais primários.

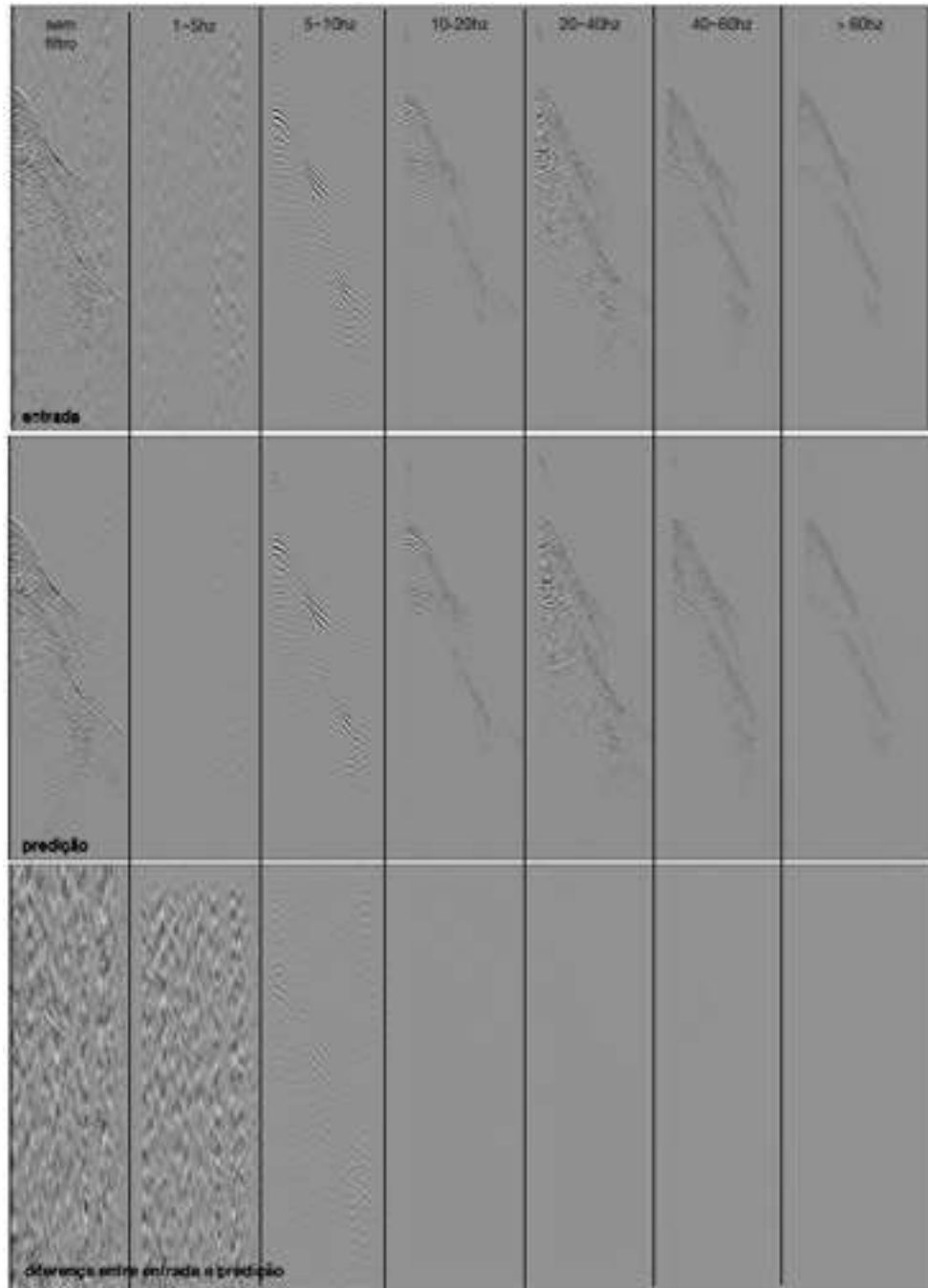


Figura 5.20: (a) Tiro de entrada, (b) Tiro predito (c) Diferença entre (a) e (b), representando o *swell* detectado

5.4 Análise qualitativa entre sísmicas de diferentes aquisições

Nesta seção, será apresentada uma análise qualitativa dos resultados de testes realizados em diferentes sísmicas das utilizadas no processo de treinamento, com o objetivo de verificar a capacidade de generalização dos modelos. A pesquisa compreendeu a análise dos testes descritos a seguir. Neste caso, não será possível fazer a comparação entre os dados originalmente processados com a predição do modelo pois foi adquirido somente os dados reais processados da bacia do Espírito Santo.

Nos testes com ruídos *swell*, o modelo *U-Net* foi treinado com dados reais do Espírito Santo e a predição foi realizada em um conjunto de dados marítimos pertencentes à bacia de Santos. Também foram realizados testes com o dado sintético M2 e ruído *swell* para treinamento, enquanto que a predição foi feita com dados reais do Espírito Santo, e vice-versa. No conjunto de dados com ruídos lineares foi selecionado o tiro 36 e para o conjunto de dados reais foi selecionado o tiro 6170.

Outros experimentos foram corporificados com ruídos lineares, sendo o modelo *U-Net* treinado com dado sísmico do modelo M1 e a predição realizada com o dado sísmico do modelo M2, e vice-versa.

A tabela 5.4 apresenta o resumo esquemático dos dados sísmicos utilizados de forma distinta para treinamento e predição, nos testes realizados em análise qualitativa para diferentes aquisições.

Tabela 5.4: Análise qualitativa - Dados sísmicos utilizados para treinamento e teste

Dado de Treinamento	Dado de Predição
Dado real da bacia do Espírito Santo (filtro passa-banda)	Dado real da bacia de Santos
Dado sintético M2 com ruído <i>swell</i>	Dado real da bacia do Espírito Santo
Dado real da bacia do Espírito Santo (filtro passa-banda)	Dado sintético M2 com ruído <i>swell</i>
Dado real da bacia do Espírito Santo (filtro F-K e <i>mute</i> no topo)	Dado sintético M2 com ruído <i>swell</i>
Dado sintético M1 com ruído linear	Dado sintético M2 com ruído linear
Dado sintético M2 com ruído linear	Dado sintético M1 com ruído linear

5.4.1 Ruído *Swell*

Na figura 5.21 é apresentado o resultado da predição do modelo *U-Net* treinado com dados do Espírito Santo (com filtro passa-banda) em um dado sísmico de diferente aquisição pertencente à bacia de Santos.

Nas figuras 5.22 e 5.23 são apresentados os resultados da tentativa de generalização realizada com modelo treinado, com dado sintético M2 com ruído *swell* e a predição feita em dado real pertencente à bacia do Espírito Santo com ruído *swell*.

Nas figuras 5.24, 5.25 e 5.26 são apresentados os resultados da tentativa de generalização realizada com modelo treinado, com dado real da bacia do do Espírito Santo (com filtro passa-banda), com ruído *swell* e predição feita em dado M2 com ruído *swell*.

Nas figuras 5.27, 5.28 e 5.29 são apresentados os resultados da tentativa de generalização realizada com modelo treinado com dado real da bacia do Espírito Santo (filtro F-K e mute no topo) , com ruído *swell* e predição feita em dados M2 com ruído *swell*.

É possível verificar através da diferença entre entrada e predição (figura 5.21(c)) que sinais primários foram atenuados junto com o ruído *swell*.

A generalização do modelo de aprendizado profundo em sísmicas de outras aquisições é um desafio e continua sendo um tópico em desenvolvimento. Outros trabalhos relacionados como [42] também apresentam dificuldades de generalização entre dados sísmicos de diferentes aquisições.

Modelo treinado com dado sísmico do Espírito Santo e predito com dado sísmico da bacia de Santos

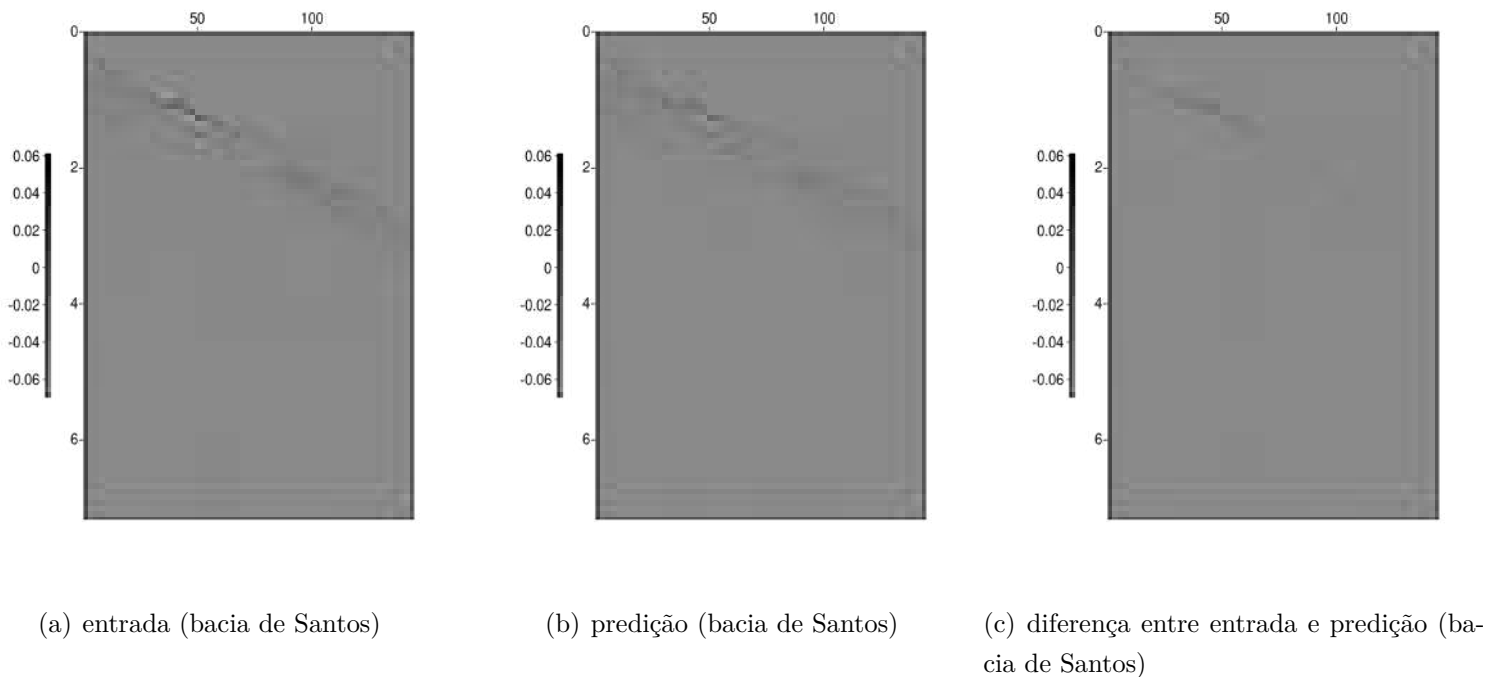


Figura 5.21: Resultado treinamento dado sísmico Espírito Santo e predição em dado sísmico da bacia de Santos

Modelo treinado com dado sintético M2 com ruído *swell* e predito em dado real da bacia do Espírito Santo com ruído *swell*

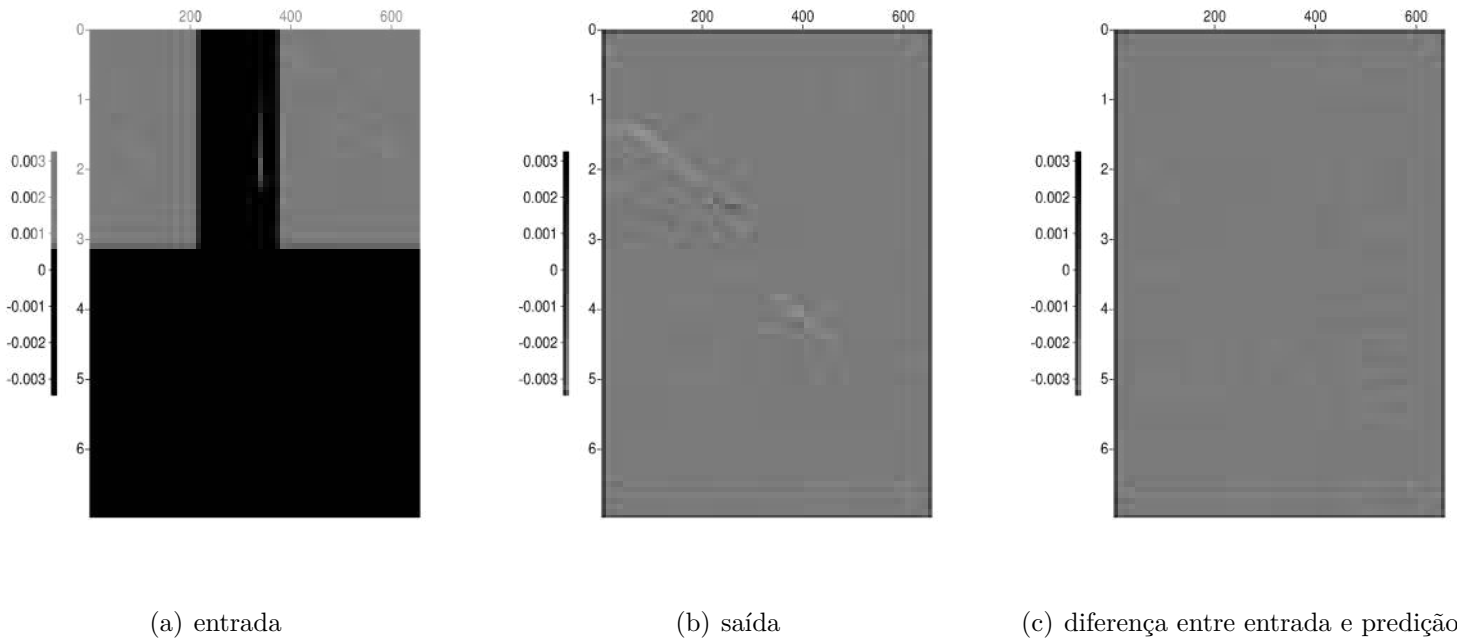


Figura 5.22: Resultado de treinamento com dado sintético M2 com ruído *swell* e predição em dado real da bacia do Espírito Santo com ruído *swell*

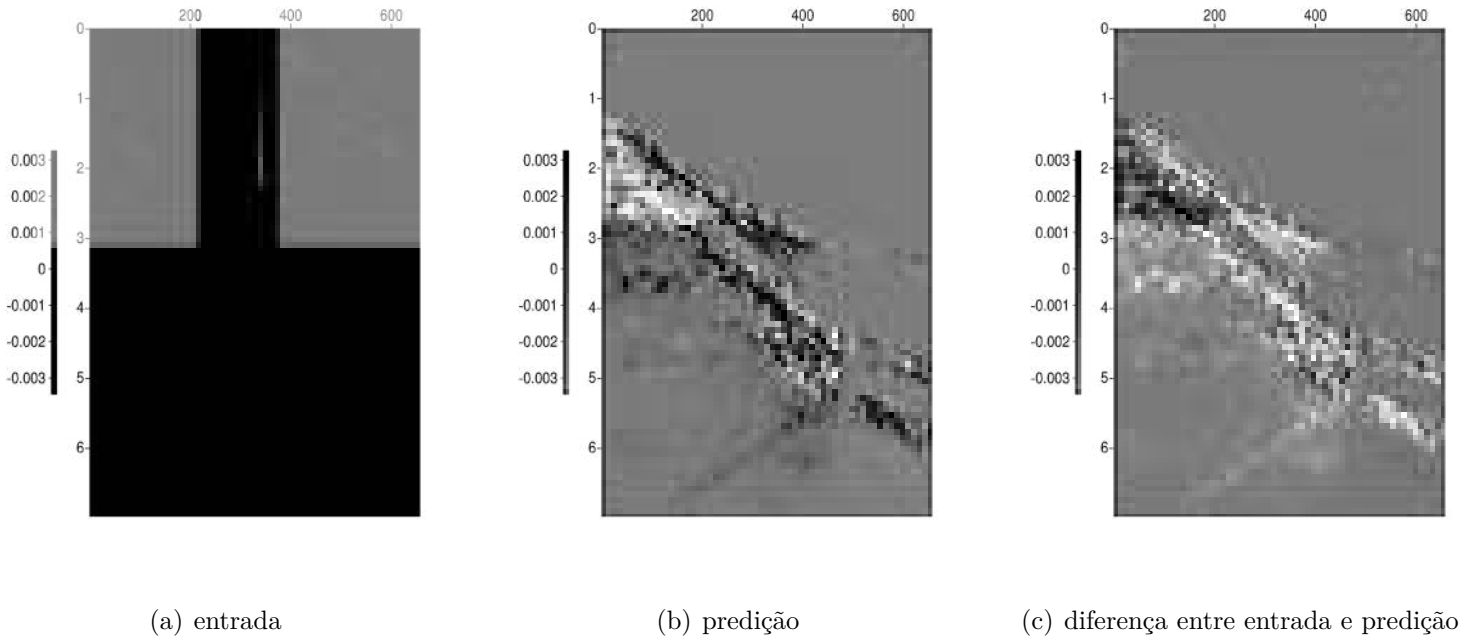


Figura 5.23: Resultado de treinamento com dado sintético M2 com ruído *swell* e predição em dado real da bacia do Espírito Santo com ruído *swell*

Modelo treinado com dado real da bacia do Espírito Santo (filtro passa-banda) com ruído *swell* e predição em dado M2 com ruído *swell*

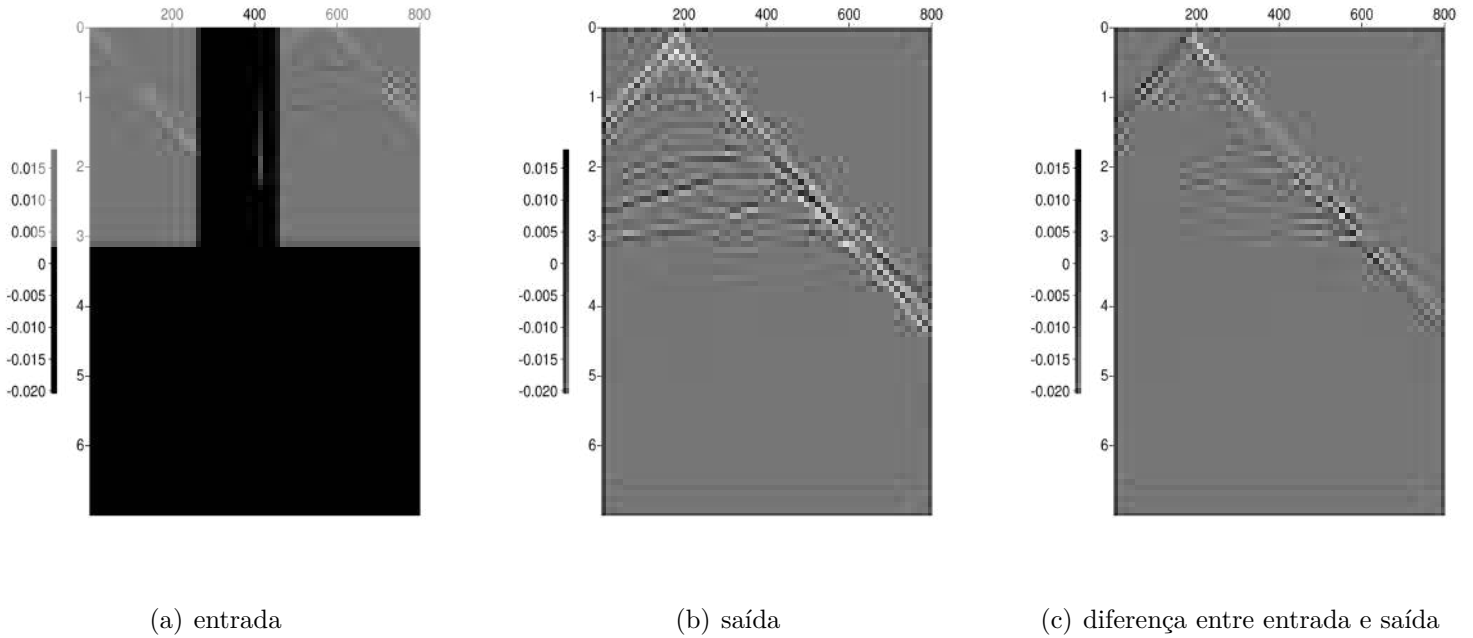


Figura 5.24: Resultado de treinamento om dado real da bacia do Espírito Santo (filtro passa-banda) com ruído *swell* e predição em dado M2 com ruído *swell*

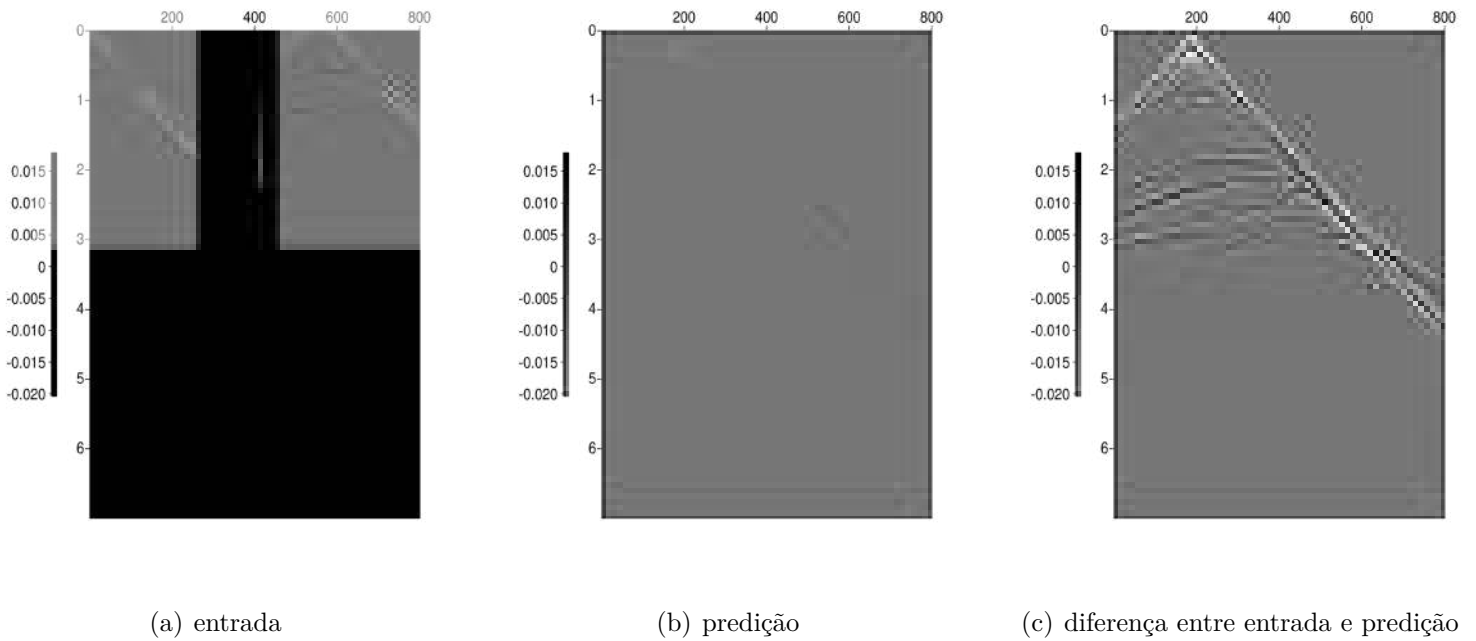
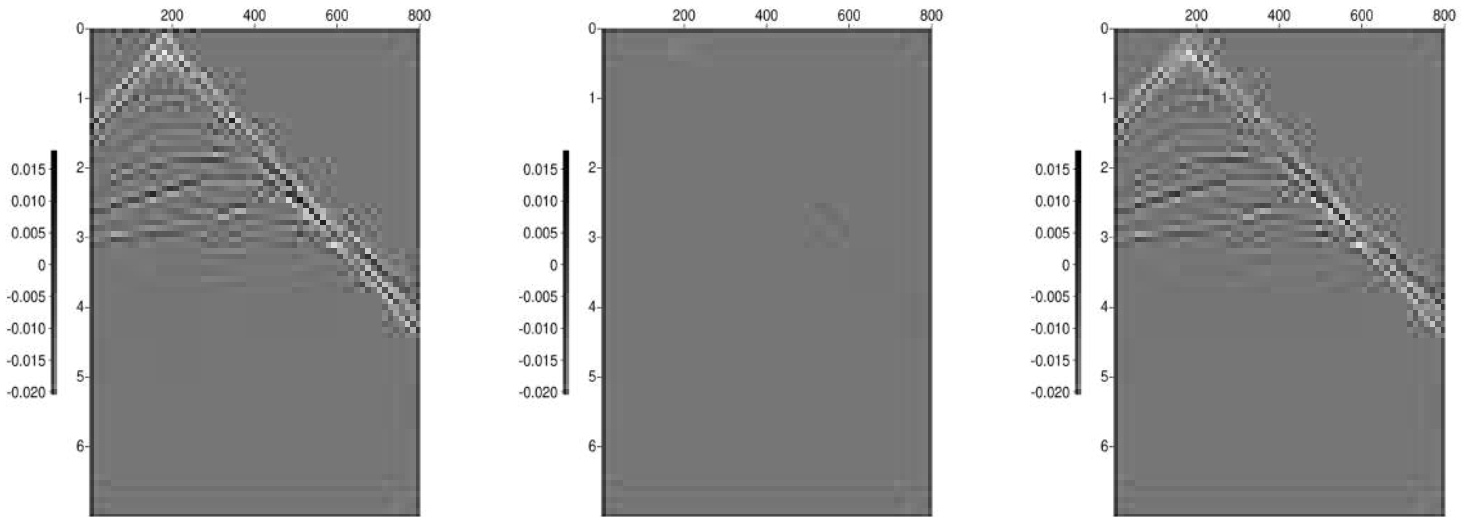


Figura 5.25: Resultado de treinamento om dado real (filtro passa-banda) da bacia do Espírito Santo com ruído *swell* e predição em dado M2 com ruído *swell*



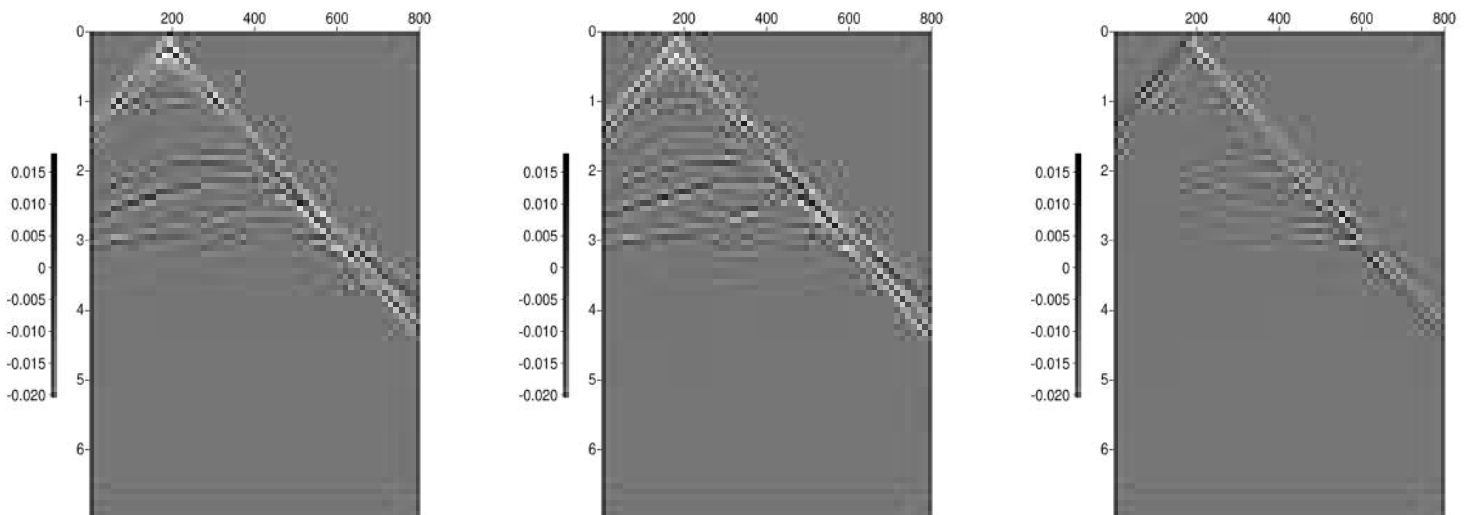
(a) saída

(b) predição

(c) diferença entre saída e predição

Figura 5.26: Resultado de treinamento com dado real da bacia do Espírito Santo (filtro passa-banda) com ruído *swell* e predição em dado M2 com ruído *swell*

Modelo treinado com dado real da bacia do Espírito Santo (filtro F-K e *mute* no topo) com ruído *swell* e predição em dado M2 com ruído *swell*

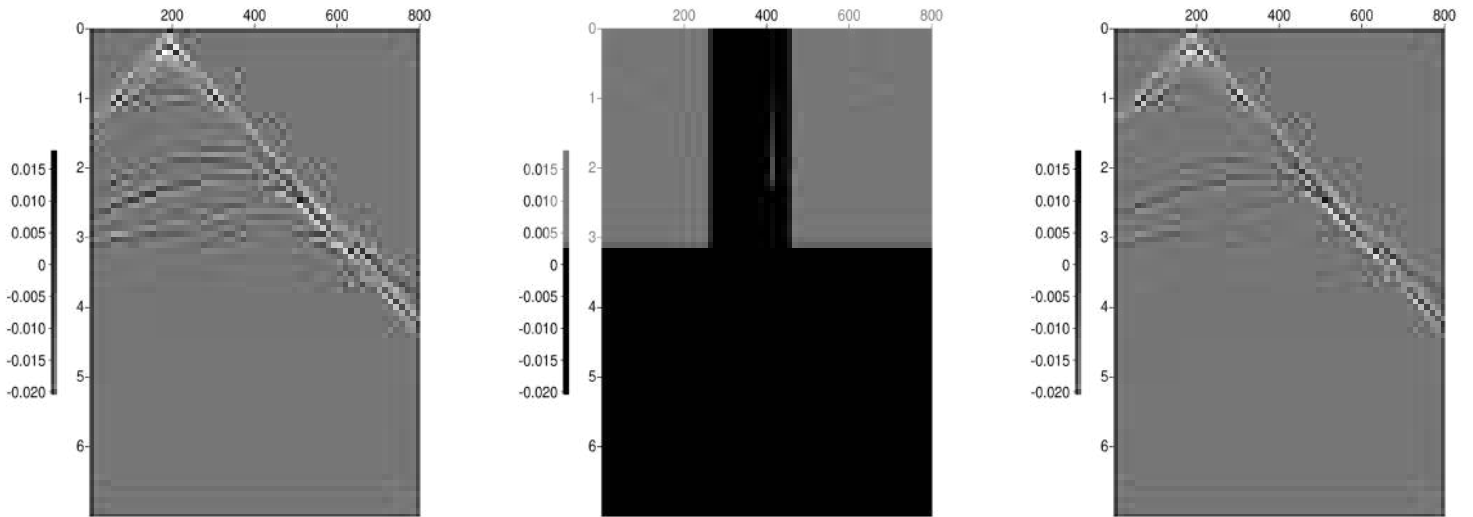


(a) entrada

(b) saída

(c) diferença entre entrada e saída

Figura 5.27: Resultado de treinamento com dado real da bacia do Espírito Santo (filtro F-K e *mute* no topo) com ruído *swell* e predição em dado M2 com ruído *swell*

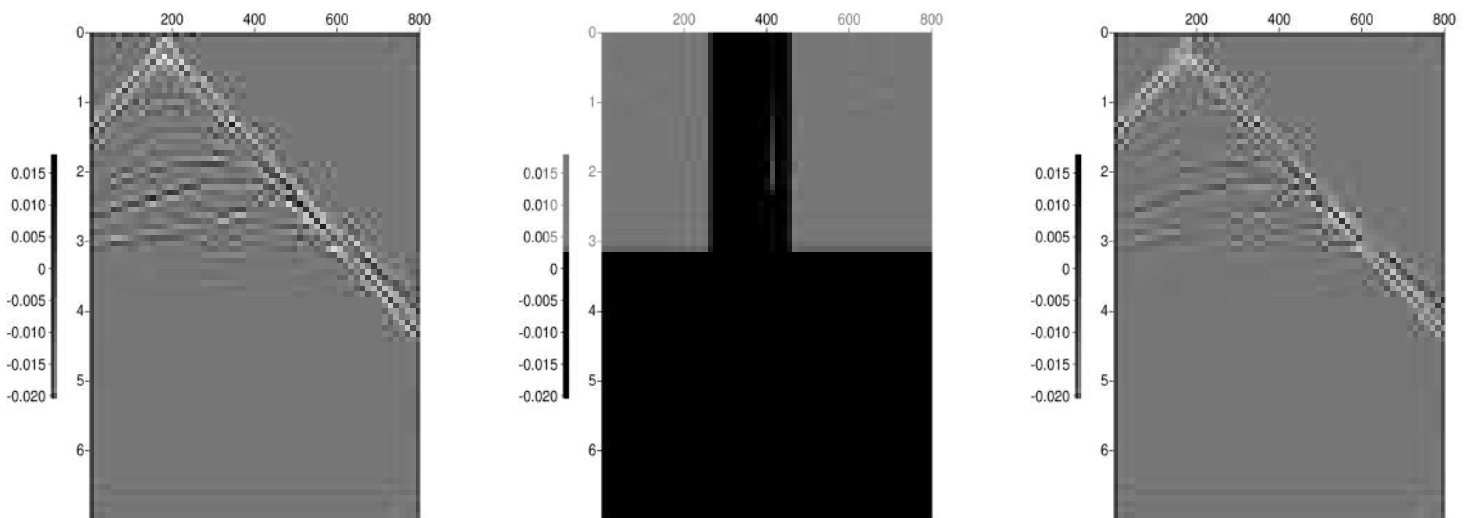


(a) entrada

(b) predição

(c) diferença entre entrada e predição

Figura 5.28: Resultado de treinamento com dado real da bacia do Espírito Santo (filtro F-K e *mute* no topo) com ruído *swell* e predição em dado M2 com ruído *swell*



(a) saída

(b) predição

(c) diferença entre saída e predição

Figura 5.29: Resultado de treinamento com dado real da bacia do Espírito Santo (filtro F-K e *mute* no topo) com ruído *swell* e predição em dado M2 com ruído *swell*

5.4.2 Ruídos Lineares

As figuras 5.30 e 5.31 apresentam os resultados da predição do modelo *U-Net* treinados e preditos com os dado sísmico M1 e M2 com ruídos lineares. Verifica-se através

das figuras 5.30(c) e 5.31(c) a vestígios da presença de ruídos lineares. Deste modo, o modelo apresentou melhor resultado na tarefa de generalização neste cenário devido a similaridade entre o tipo de ruído neste dado.

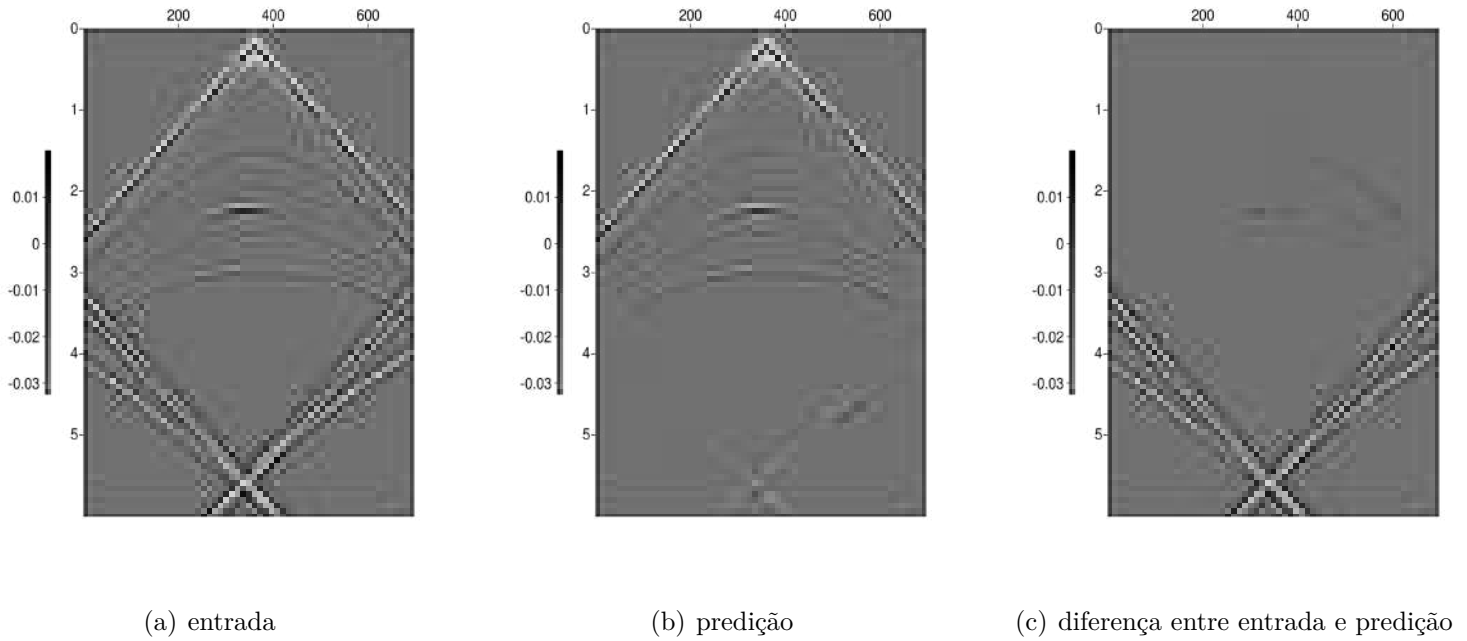


Figura 5.30: Resultado de treinamento dado sísmico modelo M1 e predição dado sísmico modelo M2

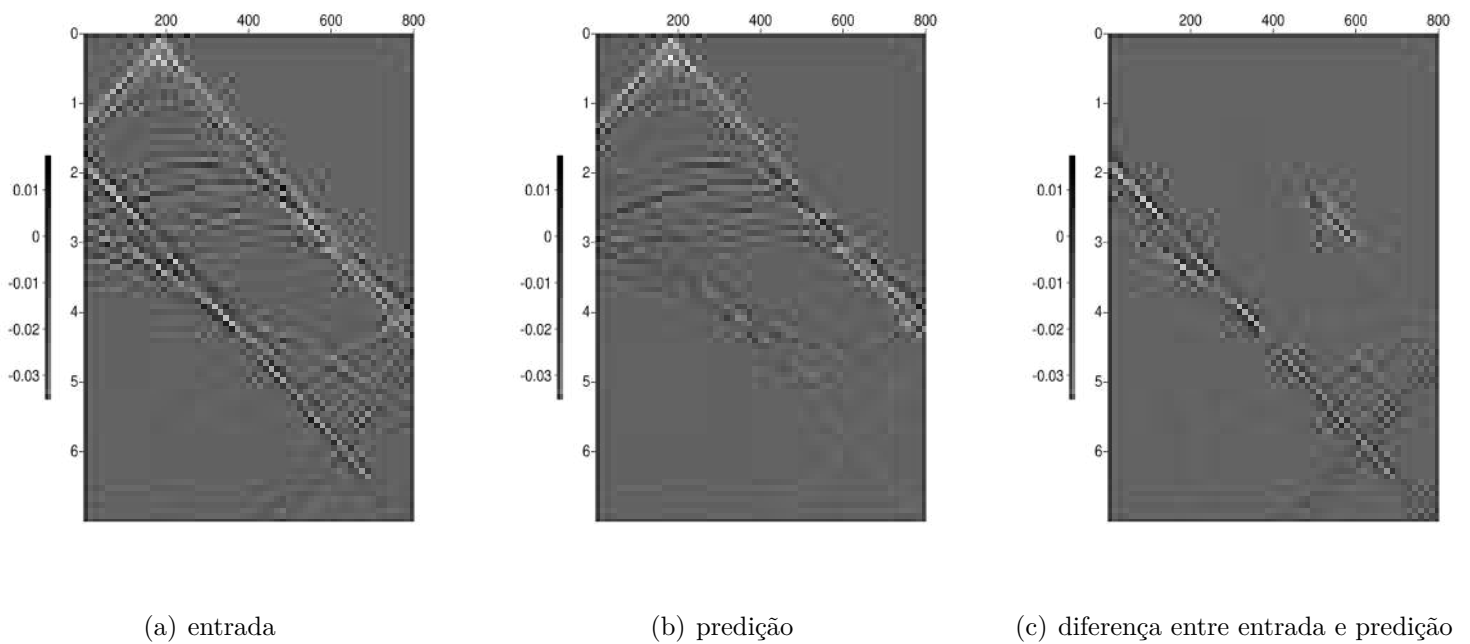


Figura 5.31: Resultado de treinamento dado sísmico modelo M2 e predição dado sísmico modelo M1

5.5 Análise quantitativa e qualitativa entre modelos *U-Net* e CNN com três camadas entre sísmicas de mesma aquisição

Nesta seção será apresentada uma análise tanto quantitativa quanto qualitativa entre os resultados das arquiteturas *U-Net* e CNN com três camadas, treinadas com dados sísmicos pré-empilhamento da bacia do Espírito Santo e com filtro passa-banda e preditas com dado sísmico de mesma aquisição. A tabela 5.3 apresenta os resultados da relação sinal ruído e MSE no conjunto de tiros de treinamento, assim como, teste. A tabela 5.5 apresenta o tempo de processamento gasto em cada uma destas etapas.

Assim sendo, as evidências demonstram que o modelo *U-Net* possui superior performance quanto em relação sinal ruído e MSE em comparação com o modelo CNN com três camadas. O modelo *U-Net* é composto por seis (6) camadas convolucionais, sendo três (3) camadas para codificação e três (3) camadas para decodificação. Outro modelo testado é uma arquitetura CNN com três camadas convolucionais. Por se tratar de um modelo mais complexo/profundo quanto ao número de camadas e parâmetros (subseção 4.2.3), conseqüentemente, o seu tempo computacional gasto para treinamento é superior. O *U-Net* consumiu 30 horas para treinamento enquanto o CNN com três camadas consumiu 14 horas, ou seja, um pouco menos do metade do tempo.

Tabela 5.5: Resultados dos Modelos *U-Net* e CNN com três camadas

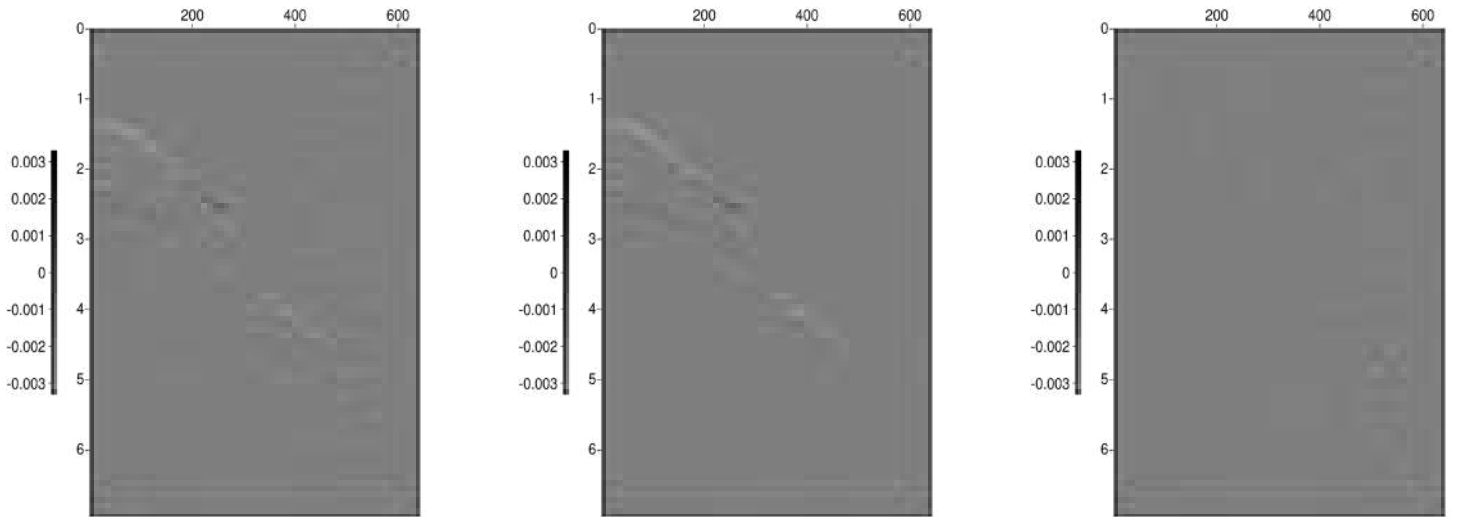
MODELO	DADO TREINAMENTO	TEMPO TREINAMENTO
ruidoso / limpo	Dado Real-Bacia do ES, filtro passa-banda, ruído <i>swell</i>	-
CNN com três camadas		14 h
<i>U-Net</i>		30 h

Quanto ao aspecto qualitativo, analisou-se a plotagem dos resultados dos modelos *U-Net* e CNN com três camadas. A figura 5.35 apresenta a plotagem tiro 6170: em (a) é mostrado o dado bruto de entrada, em (b) dado limpo de saída e em (c) a diferença entre eles. Em paralelo, a figura 5.36 traz resultados do modelo *U-Net*: em (a) entrada, em (b) predição e em (c) a diferença entre eles. Já a figura 5.37 apresenta em (a) saída, em (b) predição e em (c) a diferença entre eles. Para demonstrar a continuidade da análise, a figura 5.33 apresenta resultados do modelo CNN com três camadas: em (a) entrada, em (b) predição e em (c) a diferença entre eles. Por fim a figura 5.34 demonstra: em (a) saída, em (b) predição e em (c) a diferença entre eles.

Observa-se através dos resultados das predições entre os modelos *U-Net* 5.37(b) e CNN com três camadas 5.34(b) que a predição do modelo CNN apresentou mais

ruídos que o modelo *U-Net*. Esta diferença pode ser melhor observada no resíduo do ruído, ou seja, pela diferença entre a saída e predição *U-Net* 5.37(c) e CNN com três camadas 5.34(c). Nota-se, assim, maior quantidade de ruídos no resíduo do modelo CNN com três camadas.

CNN três camadas

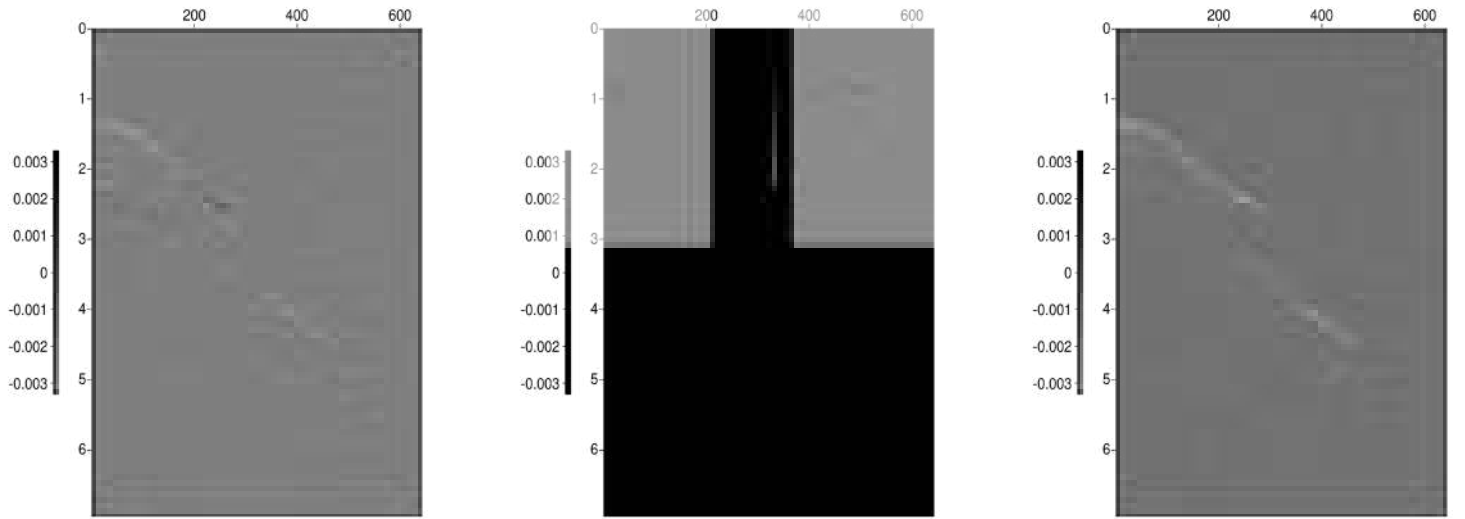


(a) entrada

(b) saída

(c) diferença entre entrada e saída

Figura 5.32: Resultado tiro 6170 referente ao modelo CNN com três camadas, com ruído *swell* e com filtro passa-banda

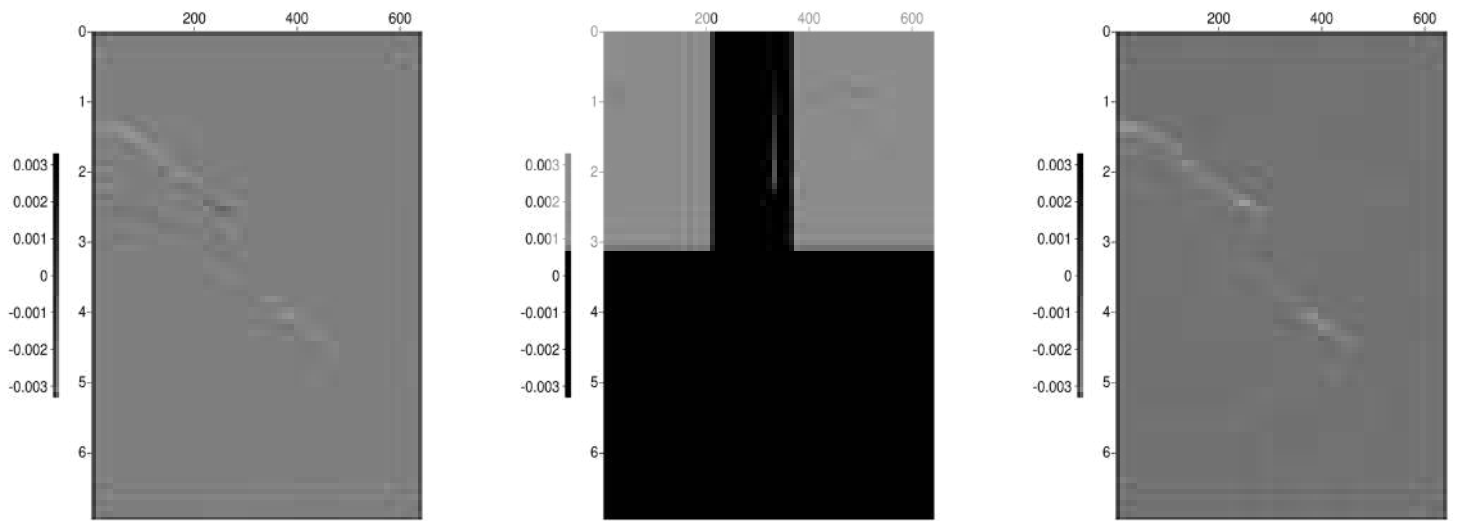


(a) entrada

(b) predição

(c) diferença entre entrada e predição

Figura 5.33: Resultado tiro 6170 referente ao modelo CNN com três camadas, com ruído *swell* e com filtro passa-banda



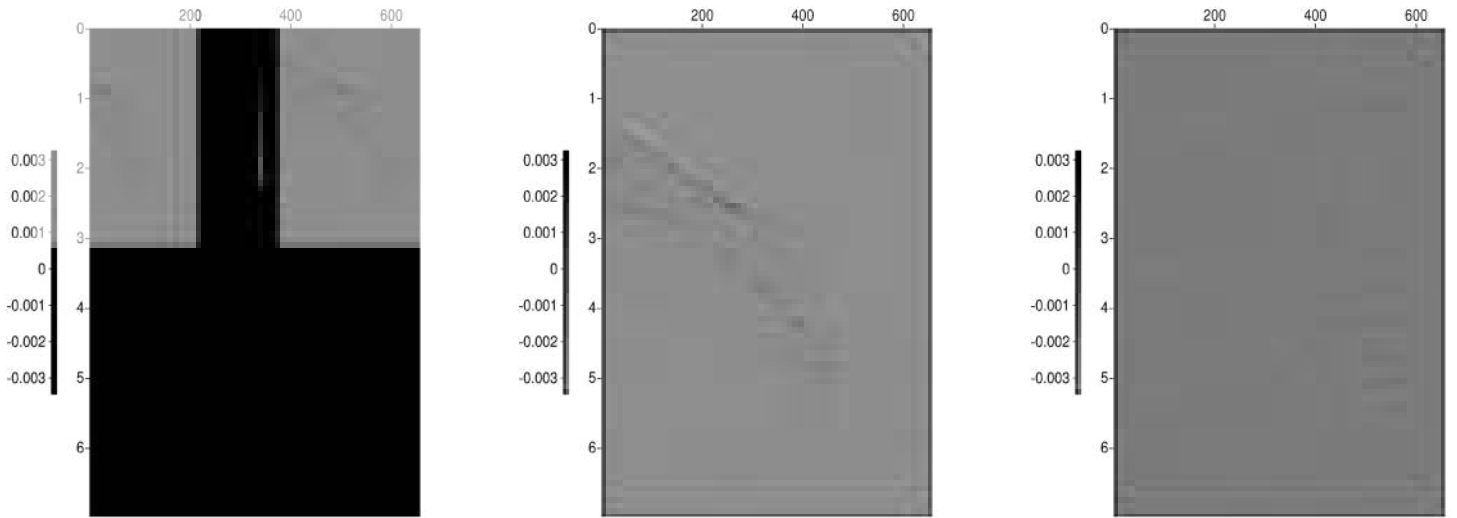
(a) saída

(b) predição

(c) diferença entre saída e predição

Figura 5.34: Resultado tiro 6170 referente ao modelo CNN com três camadas com ruído *swell* com filtro passa-banda

U-Net

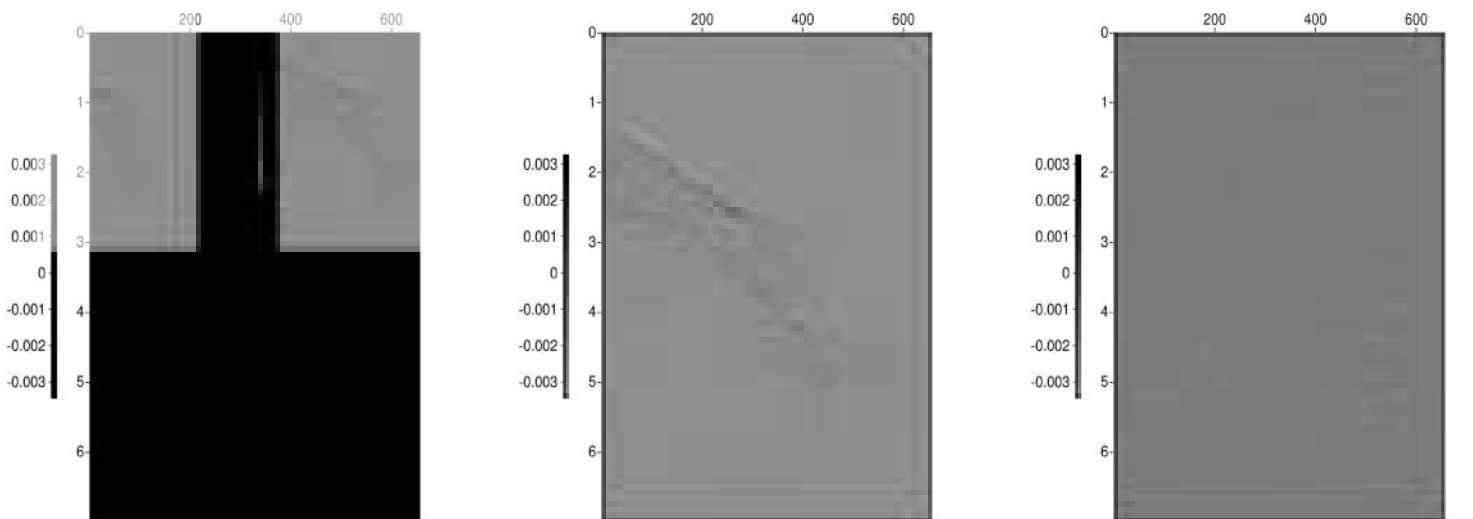


(a) entrada

(b) saída

(c) diferença entre entrada e saída

Figura 5.35: Resultado tiro 6170 referente ao modelo *U-Net* com ruído *swell* com filtro passa-banda

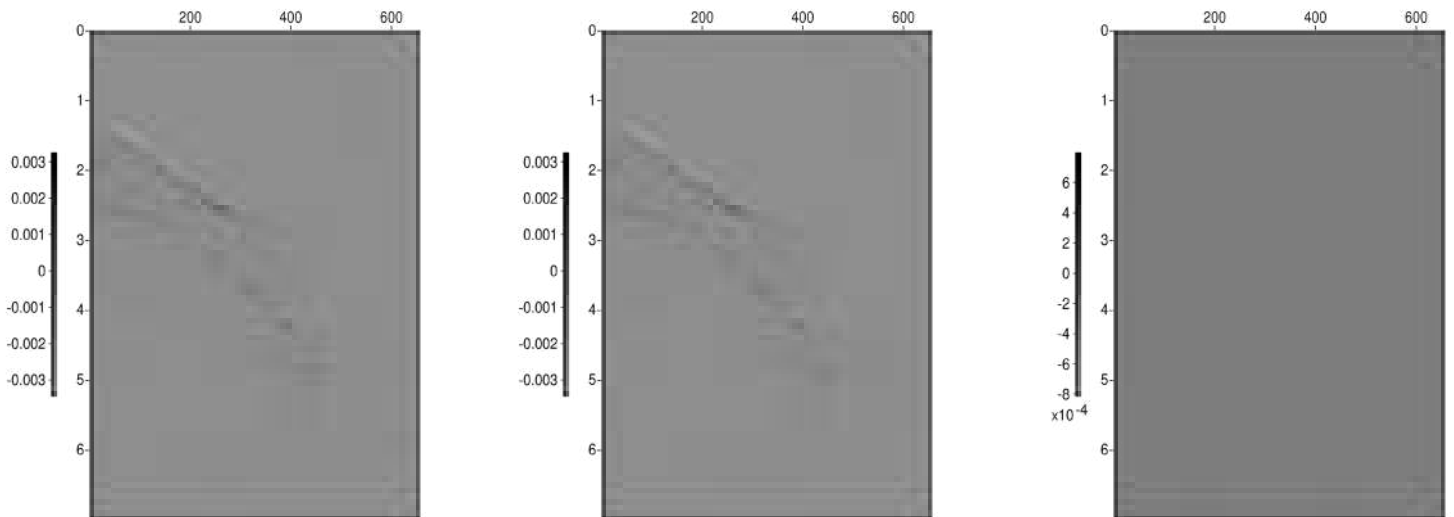


(a) entrada

(b) predição

(c) diferença entre entrada e predição

Figura 5.36: Resultado tiro 6170 com ruído *swell* e filtro passa-banda



(a) saída

(b) predição

(c) diferença entre saída e predição

Figura 5.37: Resultado tiro 6170 referente ao modelo *U-Net* com ruído *swell* com filtro passa-banda

5.6 Resumo do capítulo

Neste capítulo foram apresentados os resultados dos experimentos realizados e discussões. No próximo capítulo serão explicitadas as conclusões e os trabalhos futuros.

Capítulo 6

Conclusões e trabalhos futuros

Nesta dissertação foi proposta a utilização de modelos de aprendizado profundo com arquitetura do tipo autocodificador convolucional em tarefas de atenuação de ruídos em dados sísmicos marítimos. Foram aplicadas duas técnicas para remoção de ruído: a primeira utilizada foi o modelo de rede neural com 3 (três) camadas convolucionais e a segunda foi o modelo de rede neural *U-Net* com 6 (seis) camadas convolucionais, 3 (três) camadas para codificação e 3 (três) camadas para decodificação. Os modelos foram implementados utilizando a linguagem *Python* [19] e a biblioteca *Tensorflow* [20]. O código de treinamento foi desenvolvido para a utilização de paralelismo de GPUs. A infraestrutura utilizada para tal atividade foi um servidor Nvidia DGX2 com 8 GPUs V100 que compõe o supercomputador Santos Dumont do Laboratório Nacional de Computação Científica (LNCC) [21]. Todos os treinamentos executados utilizaram paralelização com as oito (8) GPUs contidas no servidor devido à grande quantidade de dados de entrada e os vários testes realizados com dados sísmicos pré-empilhamento reais e sintéticos com ruídos aleatórios, lineares e *swell*. O processo de treinamento dos dois (2) modelos (CNN com três camadas e *U-Net*) com todos os dados e seus respectivos ruídos apresentados neste trabalho, consumiu um total de 288 horas. Vale ressaltar que esta infraestrutura é de uso compartilhado e durante a sua utilização nesse trabalho ocorreram muitas instabilidades que interromperam o processo de treinamento dos modelos por dias. Os códigos referentes ao treinamento e implementação dos modelos *U-Net*, CNN com três camadas e de outros modelos testados neste trabalho foram disponibilizados em um repositório público do *github* no endereço https://github.com/carlossneto/seismic_denoising.

Os experimentos realizados no presente estudo foram divididos em análise quantitativa e qualitativa entre dados de uma mesma aquisição, assim como, análise qualitativa entre dados de diferentes aquisições. Com relação aos dados numéricos apresentados em análise quantitativa (seção de 5.2), verificou-se que a função de avaliação PSNR não refletiu os resultados de atenuação de ruídos nos dados sísmicos sintéticos, uma vez que, em todos os testes realizados, os valores de média e de desvio

padrão permaneceram os mesmos em relação aos testes entre tiros ruidosos e limpos utilizados para base de comparação. Já para os dados sísmicos reais, verificou-se que a função de avaliação PSNR refletiu de forma coerente os resultados dos testes realizados, destacando os resultados do modelo *U-Net* em relação ao modelo CNN com três camadas.

Já nos resultados da seção análise qualitativa entre sísmicas de mesma aquisição (5.3), observou-se no que se refere aos ruídos aleatórios, a atenuação do ruído de maneira eficaz, não obstante a descontinuidade do sinal sísmico. No caso dos ruídos lineares M1 e M2, apurou-se que o resultado da atenuação foi realizado de modo mais eficaz comparando-se com o ruído aleatório, porém, notou-se ainda a presença de vestígios de ruídos ao analisar o resíduo através da diferença entre o resultado da predição e o tiro limpo, embora a sua escala seja na ordem de 10^{-4} em comparação aos outros resultados com escala na ordem de 10^{-2} . Vale considerar que o cenário com a sísmica sintética com ruídos aleatórios está em um nível de dificuldade maior, uma vez que este ruído sobrepõe o dado de forma significativa. Por fim, se tratando dos dados sísmicos com ruído *swell*, os resultados do dado sintético apresentaram atenuação do ruído juntamente com uma leve atenuação do sinal.

Por outro lado, os dados sísmicos reais pré-empilhamento da bacia do Espírito Santo, revelam que o ruído *swell*, concentrado na baixa frequência (0-5Hz), foi removido de forma efetiva, resultado este melhor observado nos painéis de faixas de frequência (seção 5.3.4) que indica que o ruído do *swell* foi efetivamente detectado e removido com leves lesões primárias.

Portanto, na maioria dos casos apresentados, com exceção dos dados sintéticos com ruídos aleatórios e com destaque nos dados sísmicos reais, os resultados mostram que a técnica proposta pode retornar uma imagem com boa qualidade e com influência limitada no sinal de interesse.

No que se refere à análise qualitativa entre dados sísmicos de diferentes aquisições (seção 5.4), foi possível apurar que modelo *U-Net* não foi capaz de atenuar o ruído sem prejudicar significativamente o sinal. Em quase todos os testes com os dados sísmicos sintéticos e reais, que continham os ruídos *swell*, os sinais primários foram severamente prejudicados, com exceção dos dados com ruídos lineares, nos qual apresentou melhor resultado, obtendo a remoção parcial do ruído. Concluindo que a rede não obteve dificuldades no processo de generalização, ao ser treinada e predita utilizando dados sísmicos distintos. Verifica-se na literatura que a generalização de modelo de aprendizado profundo em sísmicas de outras aquisições é um desafio e continua sendo um tópico em desenvolvimento, uma vez que, trabalhos relacionados como [42] também apresentam dificuldades de generalização entre dados sísmicos de diferentes aquisições. Para objetivar maior sucesso neste tipo de teste, seria necessário a obtenção de mais dados com diferentes características a fim de

enriquecer o processo de treinamento dos modelos. Durante o processo de execução deste trabalho, foi desafiante a aquisição de dados reais pré-empilhamento sem a presença de ruídos, devido a necessidade de execução desta atividade por parte do Geofísico.

A presente pesquisa tem por objetivo acelerar o processamento de atenuação de ruídos automatizando parte deste processo através da utilização de um modelo convolucional para aprender o processo de atenuação de ruídos sísmicos. À vista de todo o exposto no decorrer do trabalho, é possível concluir com os resultados da análise quantitativa e qualitativa destacando-se os dados reais da bacia do Espírito Santo com ruído *swell* e dado sintético com ruído linear, a capacidade da rede em reproduzir o filtro com a qualidade empregada no tratamento do dado sísmico limpo (saída). Pode-se afirmar que este é o caminho para criar um filtro semi-automático e posteriormente um filtro baseado na física do problema, e neste caso, seria preciso modelar sinteticamente o processo de filtragem. Esse processo pode ser estendido para diferentes filtros, tanto os empregados no tratamento do dado quanto os filtros naturais presentes nas técnicas de modelagem do campo de onda sísmica.

Como forma de continuação desta pesquisa a principal abordagem seguindo no processo de avaliação qualitativa seria aplicar o processo de migração sísmica nos resultados preditos pelos modelos como executado pelo autor Zhao e colaboradores em [124]. Com a diferença dos resultados das migrações sísmicas realizadas entre dados sem a presença de ruído utilizado como alvo no treinamento e dos resultados das predições, será possível observar qualitativamente o potencial mais amplo na atenuação de ruídos diferentes das observações qualitativas locais realizadas entre determinados tiros.

O presente trabalho ainda pode ser evoluído empregando-se modelos mais profundos no processo de treinamento ou redes mais complexas como arquiteturas generativas GAN [125]. Uma outra sugestão seria utilizar a técnica conhecida como transferência de aprendizado nesses tipos de modelos. Esta técnica consiste no treinamento prévio de um modelo em um conjunto de dados diferentes do interesse principal, ou seja, do seu retreinamento, para ajuste fino dos parâmetros, no conjunto de dados de interesse [126].

Ademais, um outro enfoque para evolução deste trabalho seria modificar os dados de entrada, utilizando-se apenas recortes dos dados com trechos ruidosos. Evoluindo neste aspecto, outra alternativa seria utilizar o espectro F-K (com e sem a presença de ruído) para treinamento do modelo. Neste caso, o tiro sísmico seria obtido através da transformada inversa de *Fourier* a partir do espectro F-K predito pelo modelo.

Referências Bibliográficas

- [1] DONDURUR, D. *Acquisition and processing of marine seismic data*. Elsevier, 2018.
- [2] “Figura de aquisição terrestre”. <https://www.argas.com/land-data-acquisition/>, . Acessado em 2020-03-01.
- [3] “Figura de aquisição marinha”. <https://www.semanticscholar.org/paper/Seismic-blending-and-deblending-of-crossline-Urruticoechea/045e509c2cd39b25cafb978a2062bc5259c3466>, . Acessado em 2020-03-01.
- [4] “Processamento Sísmico: Notas de aula de Ricardo Bilotie”. <http://www.ime.unicamp.br/~biloti/geo/notas.pdf>, . Acessado em 2020-03-28.
- [5] YILMAZ, Ö. *Seismic data analysis: Processing, inversion, and interpretation of seismic data*. Society of exploration geophysicists, 2001.
- [6] MAJUMDAR, A. “Blind Denoising Autoencoder”, *IEEE transactions on neural networks and learning systems*, v. 30, n. 1, pp. 312–317, 2018.
- [7] SZEGEDY, C., IOFFE, S., VANHOUCHE, V., et al. “Inception-v4, inception-resnet and the impact of residual connections on learning”, *arXiv preprint arXiv:1602.07261*, 2016.
- [8] LIU, L., OUYANG, W., WANG, X., et al. “Deep learning for generic object detection: A survey”, *International journal of computer vision*, v. 128, n. 2, pp. 261–318, 2020.
- [9] LONG, J., SHELFHAMER, E., DARRELL, T. “Fully Convolutional Networks for Semantic Segmentation”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.
- [10] HE, K., GKIOXARI, G., DOLLAR, P., et al. “Mask R-CNN”. In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

- [11] BOURLARD, H. A., MORGAN, N. *Connectionist speech recognition: a hybrid approach*, v. 247. Springer Science & Business Media, 2012.
- [12] VASWANI, A., BENGIO, S., BREVDO, E., et al. “Tensor2tensor for neural machine translation”, *arXiv preprint arXiv:1803.07416*, 2018.
- [13] ZHANG, L., WANG, S., LIU, B. “Deep learning for sentiment analysis: A survey”, *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, v. 8, n. 4, pp. e1253, 2018.
- [14] LECUN, Y., OTHERS. “Generalization and network design strategies”. In: *Connectionism in perspective*, v. 19, Citeseer, 1989.
- [15] KRIZHEVSKY, A., SUTSKEVER, I., HINTON, G. E. “Imagenet classification with deep convolutional neural networks”. In: *Advances in neural information processing systems*, pp. 1097–1105, 2012.
- [16] DENG, J., DONG, W., SOCHER, R., et al. “Imagenet: A large-scale hierarchical image database”. In: *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.
- [17] LECUN, Y., TOURESKY, D., HINTON, G., et al. “A theoretical framework for back-propagation”. In: *Proceedings of the 1988 connectionist models summer school*, v. 1, pp. 21–28. CMU, Pittsburgh, Pa: Morgan Kaufmann, 1988.
- [18] CARCIONE, J. M., HERMAN, G. C., TEN KROODE, A. “Seismic modeling”, *Geophysics*, v. 67, n. 4, pp. 1304–1325, 2002.
- [19] “Python”. <https://www.python.org/>, . Acessado em 2019-05-30.
- [20] “Open Source Deep Learning Library”. <https://www.tensorflow.org/>. Acessado em 2019-05-30.
- [21] “Santos Dumont - LNCC”. <https://sdumont.lncc.br/>, . Acessado em 2020-05-30.
- [22] GOODFELLOW, I., BENGIO, Y., COURVILLE, A. *Deep Learning*. MIT Press, 2016. <http://www.deeplearningbook.org>.
- [23] IKELLE, L., AMUNDSEN, L. “Introduction to petroleum seismology: Society of Exploration Geophysicists”, *Tulsa, Oklahoma*, 2005.
- [24] ERNST, F., HERMAN, G. “Tomography of dispersive media: Journal of the Acoustic Society of America”, 2000.

- [25] SHERIFF, R. E., GELDART, L. P. *Exploration seismology*. Cambridge university press, 1995.
- [26] GLOSSARY, S. O. “The oilfield glossary”, 2018.
- [27] ELBOTH, T., GEOTEAM, F., HERMANSEN, D. “Attenuation of noise in marine seismic data”. In: *SEG Technical Program Expanded Abstracts 2009*, Society of Exploration Geophysicists, pp. 3312–3316, 2009.
- [28] DONDURUR, D., KARSLI, H. “Swell noise suppression by wiener prediction filter”, *Journal of applied geophysics*, v. 80, pp. 91–100, 2012.
- [29] CANALES, L. L. “Random noise reduction”. In: *SEG Technical Program Expanded Abstracts 1984*, Society of Exploration Geophysicists, pp. 525–527, 1984.
- [30] ABMA, R., CLAERBOUT, J. “Lateral prediction for noise attenuation by tx and fx techniques”, *Geophysics*, v. 60, n. 6, pp. 1887–1896, 1995.
- [31] TRICKETT, S. “F-xy Cadzow noise suppression”. In: *SEG Technical Program Expanded Abstracts 2008*, Society of Exploration Geophysicists, pp. 2586–2590, 2008.
- [32] OROPEZA, V., SACCHI, M. “Simultaneous seismic data denoising and reconstruction via multichannel singular spectrum analysis”, *Geophysics*, v. 76, n. 3, pp. V25–V32, 2011.
- [33] BONAR, D., SACCHI, M. “Denoising seismic data using the nonlocal means algorithm”, *Geophysics*, v. 77, n. 1, pp. A5–A8, 2012.
- [34] TIAN, Y., LI, Y., YANG, B. “Variable-eccentricity hyperbolic-trace TFPF for seismic random noise attenuation”, *IEEE Transactions on Geoscience and Remote Sensing*, v. 52, n. 10, pp. 6449–6458, 2014.
- [35] ZHU, W., MOUSAVI, S. M., BEROZA, G. C. “Seismic signal denoising and decomposition using deep neural networks”, *arXiv preprint arXiv:1811.02695*, 2018.
- [36] ZHANG, R., ULRYCH, T. J. “Physical wavelet frame denoising”, *Geophysics*, v. 68, n. 1, pp. 225–231, 2003.
- [37] FOMEL, S. “Towards the seislet transform”. In: *SEG Technical Program Expanded Abstracts 2006*, Society of Exploration Geophysicists, pp. 2847–2851, 2006.

- [38] HERRMANN, F. J., HENNENFENT, G. “Non-parametric seismic data recovery with curvelet frames”, *Geophysical Journal International*, v. 173, n. 1, pp. 233–248, 2008.
- [39] NEELAMANI, R., BAUMSTEIN, A. I., GILLARD, D. G., et al. “Coherent and random noise attenuation using the curvelet transform”, *The Leading Edge*, v. 27, n. 2, pp. 240–248, 2008.
- [40] DARCHE, G. “Spatial interpolation using a fast parabolic transform”. In: *SEG Technical Program Expanded Abstracts 1990*, Society of Exploration Geophysicists, pp. 1647–1650, 1990.
- [41] TRAD, D. O., ULRYCH, T. J., SACCHI, M. D. “Accurate interpolation with high-resolution time-variant Radon transforms”, *Geophysics*, v. 67, n. 2, pp. 644–656, 2002.
- [42] SCHONEWILLE, M., VIGNER, A., RYDER, A. “Swell-noise attenuation using an iterative FX prediction filtering approach”. In: *SEG Technical Program Expanded Abstracts 2008*, Society of Exploration Geophysicists, pp. 2647–2651, 2008.
- [43] BEKARA, M., VAN DER BAAN, M. “High-amplitude noise detection by the expectation-maximization algorithm with application to swell-noise attenuation”, *Geophysics*, v. 75, n. 3, pp. V39–V49, 2010.
- [44] STERNFELS, R., VIGUIER, G., GONDOIN, R., et al. “Joint low-rank and sparse inversion for multidimensional simultaneous random/erratic noise attenuation and interpolation”. In: *77th EAGE Conference and Exhibition 2015*, v. 2015, pp. 1–5. European Association of Geoscientists & Engineers, 2015.
- [45] CHEN, Y., MA, J., FOMEL, S. “Double-sparsity dictionary for seismic noise attenuation”, *Geophysics*, v. 81, n. 2, pp. V103–V116, 2016.
- [46] CHEN, Y. “Fast dictionary learning for noise attenuation of multidimensional seismic data”, *Geophysical Journal International*, v. 209, n. 1, pp. 21–31, 2017.
- [47] SIAHSAR, M. A. N., GHOLTASHI, S., ABOLGHASEMI, V., et al. “Simultaneous denoising and interpolation of 2D seismic data using data-driven non-negative dictionary learning”, *Signal Processing*, v. 141, pp. 309–321, 2017.

- [48] HEIDEMAN, M., JOHNSON, D., BURRUS, C. “Gauss and the history of the fast Fourier transform”, *IEEE ASSP Magazine*, v. 1, n. 4, pp. 14–21, 1984.
- [49] PRANDONI, P., VETTERLI, M. *Signal processing for communications*. Collection le savoir suisse, 2008.
- [50] BRACEWELL, R. N., BRACEWELL, R. N. *The Fourier transform and its applications*, v. 31999. McGraw-Hill New York, 1986.
- [51] OPPENHEIM, A. V., WILLSKY, A. S., NAWAB, S. H. *Signals Systems (2nd Ed.)*. USA, Prentice-Hall, Inc., 1996. ISBN: 0138147574.
- [52] CLAERBOUT, J. F. *Fundamentals of geophysical data processing*. Citeseer, 1985.
- [53] DAI, W., FOWLER, P., SCHUSTER, G. T. “Multi-source least-squares reverse time migration”, *Geophysical Prospecting*, v. 60, n. 4, pp. 681–695, 2012.
- [54] BULAND, A., OMRE, H. “Bayesian linearized AVO inversion”, *Geophysics*, v. 68, n. 1, pp. 185–198, 2003.
- [55] LI, T., MALLICK, S. “Multicomponent, multi-azimuth pre-stack seismic waveform inversion for azimuthally anisotropic media using a parallel and computationally efficient non-dominated sorting genetic algorithm”, *Geophysical Journal International*, v. 200, n. 2, pp. 1136–1154, 2015.
- [56] PRATT, R. G. “Seismic waveform inversion in the frequency domain, Part 1: Theory and verification in a physical scale model”, *Geophysics*, v. 64, n. 3, pp. 888–901, 1999.
- [57] CHEN, J., ZELT, C. A. “Application of frequency-dependent traveltimes tomography and full waveform inversion to realistic near-surface seismic refraction data”, *Journal of Environmental and Engineering Geophysics*, v. 21, n. 1, pp. 1–12, 2016.
- [58] CHEN, J., SIXTA, D., RANEY, G., et al. “Improved sub-salt imaging from reflection full waveform inversion guided salt scenario interpretation: A case history from deep water Gulf of Mexico”. In: *SEG Technical Program Expanded Abstracts 2018*, Society of Exploration Geophysicists, pp. 3773–3777, 2018.
- [59] BROWN, A. R. *Interpretation of three-dimensional seismic data*. Society of Exploration Geophysicists and American Association of Petroleum . . . , 2011.

- [60] HOCHREITER, S. “The vanishing gradient problem during learning recurrent neural nets and problem solutions”, *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, v. 6, n. 02, pp. 107–116, 1998.
- [61] NWANKPA, C., IJOMAH, W., GACHAGAN, A., et al. “Activation functions: Comparison of trends in practice and research for deep learning”, *arXiv preprint arXiv:1811.03378*, 2018.
- [62] LECUN, Y., JACKEL, L., BOTTOU, L., et al. “Learning algorithms for classification: A comparison on handwritten digit recognition”, *Neural networks: the statistical mechanics perspective*, v. 261, pp. 276, 1995.
- [63] “Imagenet Large Scale Visual Recognition Challenge (ILSVRC)”. <http://www.image-net.org/challenges/LSVRC/>, . Acessado em 2020-07-25.
- [64] SRIVASTAVA, N., HINTON, G., KRIZHEVSKY, A., et al. “Dropout: A Simple Way to Prevent Neural Networks from Overfitting”, *Journal of Machine Learning Research*, v. 15, n. 56, pp. 1929–1958, 2014. Disponível em: <<http://jmlr.org/papers/v15/srivastava14a.html>>.
- [65] SIMONYAN, K., ZISSERMAN, A. “Very deep convolutional networks for large-scale image recognition”, *arXiv preprint arXiv:1409.1556*, 2014.
- [66] SZEGEDY, C., LIU, W., JIA, Y., et al. “Going deeper with convolutions”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1–9, 2015.
- [67] DAUPHIN, Y., DE VRIES, H., CHUNG, J., et al. “Rmsprop and equilibrated adaptive learning rates for non-convex optimization. arXiv: 150204390”, 2015.
- [68] HE, K., ZHANG, X., REN, S., et al. “Deep residual learning for image recognition”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.
- [69] ZHAI, J., ZHANG, S., CHEN, J., et al. “Autoencoder and Its Various Variants”. In: *2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 415–419. IEEE, 2018.
- [70] LECUN, Y. *PhD thesis: Modeles connexionnistes de l'apprentissage (connectionist learning models)*. Universite P. et M. Curie (Paris 6), jun. 1987.
- [71] HINTON, G. E., ZEMEL, R. S. “Autoencoders, minimum description length and Helmholtz free energy”. In: *Advances in neural information processing systems*, pp. 3–10, 1994.

- [72] BOURLARD, H., KAMP, Y. “Auto-association by multilayer perceptrons and singular value decomposition”, *Biological cybernetics*, v. 59, n. 4-5, pp. 291–294, 1988.
- [73] GOODFELLOW, I., BENGIO, Y., COURVILLE, A. *Deep learning*. MIT press, 2016.
- [74] GOODFELLOW, I., POUGET-ABADIE, J., MIRZA, M., etal. “Generative adversarial nets”. In: *Advances in neural information processing systems*, pp. 2672–2680, 2014.
- [75] DENG, L. “Three classes of deep learning architectures and their applications: a tutorial survey”, *APSIPA transactions on signal and information processing*, 2012.
- [76] VINCENT, P. “A connection between score matching and denoising autoencoders”, *Neural computation*, v. 23, n. 7, pp. 1661–1674, 2011.
- [77]
- [78] RIFAI, S., VINCENT, P., MULLER, X., etal. “Contractive auto-encoders: Explicit invariance during feature extraction”, 2011.
- [79] CHEVITARESE, D., SZWARCMAN, D., SILVA, R. M. D., etal. “Seismic facies segmentation using deep learning”, *AAPG Annual and Exhibition*, 2018.
- [80] KARCHEVSKIY, M., ASHRAPOV, I., KOZINKIN, L. “Automatic salt deposits segmentation: A deep learning approach”, *arXiv preprint arXiv:1812.01429*, 2018.
- [81] WRONA, T., PAN, I., GAWTHORPE, R. L., etal. “Seismic facies analysis using machine learning”, *Geophysics*, v. 83, n. 5, pp. O83–O95, 2018.
- [82] RICHARDSON, A. “Seismic full-waveform inversion using deep learning tools and techniques”, *arXiv preprint arXiv:1801.07232*, 2018.
- [83] LIN, Y., WU, Y. “InversionNet: A real-time and accurate full waveform inversion with convolutional neural network”, *The Journal of the Acoustical Society of America*, v. 144, n. 3, pp. 1683–1683, 2018.
- [84] LI, S., LIU, B., REN, Y., etal. “Deep learning inversion of seismic data”, *arXiv preprint arXiv:1901.07733*, 2019.
- [85] RABIE, T. “Robust estimation approach for blind denoising”, *IEEE Transactions on Image Processing*, v. 14, n. 11, pp. 1755–1765, 2005.

- [86] JAIN, V., SEUNG, S. “Natural image denoising with convolutional networks”. In: *Advances in neural information processing systems*, pp. 769–776, 2009.
- [87] XIE, J., XU, L., CHEN, E. “Image denoising and inpainting with deep neural networks”. In: *Advances in neural information processing systems*, pp. 341–349, 2012.
- [88] BURGER, H. C., SCHULER, C. J., HARMELING, S. “Image denoising: Can plain neural networks compete with BM3D?” In: *2012 IEEE conference on computer vision and pattern recognition*, pp. 2392–2399. IEEE, 2012.
- [89] IOFFE, S., SZEGEDY, C. “Batch normalization: Accelerating deep network training by reducing internal covariate shift”, *arXiv preprint arXiv:1502.03167*, 2015.
- [90] CHEN, Y., ZHOU, Y., CHEN, W., et al. “Empirical low-rank approximation for seismic noise attenuation”, *IEEE Transactions on Geoscience and Remote Sensing*, v. 55, n. 8, pp. 4696–4711, 2017.
- [91] LEHTINEN, J., MUNKBERG, J., HASSELGREN, J., et al. “Noise2noise: Learning image restoration without clean data”, *arXiv preprint arXiv:1803.04189*, 2018.
- [92] GUO, G., LI, S. Z. “Content-based audio classification and retrieval by support vector machines”, *IEEE transactions on Neural Networks*, v. 14, n. 1, pp. 209–215, 2003.
- [93] ZHANG, Y., LIN, H., LI, Y. “Noise attenuation for seismic image using a deep residual learning”. In: *SEG Technical Program Expanded Abstracts 2018*, Society of Exploration Geophysicists, pp. 2176–2180, 2018.
- [94] MAO, X., SHEN, C., YANG, Y.-B. “Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections”. In: *Advances in neural information processing systems*, pp. 2802–2810, 2016.
- [95] YANG, D., SUN, J. “BM3D-Net: A convolutional neural network for transform-domain collaborative filtering”, *IEEE Signal Processing Letters*, v. 25, n. 1, pp. 55–59, 2017.
- [96] VAN VEEN, D., JALAL, A., SOLTANOLKOTABI, M., et al. “Compressed sensing with deep image prior and learned regularization”, *arXiv preprint arXiv:1806.06438*, 2018.

- [97] MANDELLI, S., LIPARI, V., BESTAGINI, P., et al. “Interpolation and Denoising of Seismic Data using Convolutional Neural Networks”, *arXiv preprint arXiv:1901.07927*, 2019.
- [98] RONNEBERGER, O., FISCHER, P., BROX, T. “U-net: Convolutional networks for biomedical image segmentation”. In: *International Conference on Medical image computing and computer-assisted intervention*, pp. 234–241. Springer, 2015.
- [99] JIN, K. H., MCCANN, M. T., FROUSTEY, E., et al. “Deep convolutional neural network for inverse problems in imaging”, *IEEE Transactions on Image Processing*, v. 26, n. 9, pp. 4509–4522, 2017.
- [100] JIN, Y., WU, X., CHEN, J., et al. “Seismic data denoising by deep-residual networks”. In: *SEG Technical Program Expanded Abstracts 2018*, Society of Exploration Geophysicists, pp. 4593–4597, 2018.
- [101] ZHANG, F., LIU, D., WANG, X., et al. “Random noise attenuation method for seismic data based on deep residual networks”. In: *International Geophysical Conference, Beijing, China, 24-27 April 2018*, pp. 1774–1777. Society of Exploration Geophysicists and Chinese Petroleum Society, 2018.
- [102] YU, F., KOLTUN, V. “Multi-scale context aggregation by dilated convolutions”, *arXiv preprint arXiv:1511.07122*, 2015.
- [103] ZHAO, H., GALLO, O., FROSIO, I., et al. “Loss functions for image restoration with neural networks”, *IEEE Transactions on Computational Imaging*, v. 3, n. 1, pp. 47–57, 2016.
- [104] JOHNSON, J., ALAHI, A., FEI-FEI, L. “Perceptual losses for real-time style transfer and super-resolution”. In: *European conference on computer vision*, pp. 694–711. Springer, 2016.
- [105] ALWON, S. “Generative adversarial networks in seismic data processing: 88th Annual International Meeting”. In: *SEG, Expanded Abstracts*, v. 1995, 1991.
- [106] RADFORD, A., METZ, L., CHINTALA, S. “Unsupervised representation learning with deep convolutional generative adversarial networks”, *arXiv preprint arXiv:1511.06434*, 2015.
- [107] ARJOVSKY, M., CHINTALA, S., BOTTOU, L. “Wasserstein gan”, *arXiv preprint arXiv:1701.07875*, 2017.

- [108] ISOLA, P., ZHU, J.-Y., ZHOU, T., et al. “Image-to-image translation with conditional adversarial networks”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1125–1134, 2017.
- [109] BARRY, K., CAVERS, D., KNEALE, C. “Recommended standards for digital tape formats”, *Geophysics*, v. 40, n. 2, pp. 344–352, 1975.
- [110] “segio lib”. <https://github.com/equinor/segio>, . Acessado em 2020-05-30.
- [111] “Processamento Sísmico: Notas de aula de Ricardo Bilotie”. http://web.mit.edu/cwpsu_v44r1/sumannual_600dpi_letter.pdf. Acessado em 2020-04-14.
- [112] MARTIN, G. S., MARFURT, K. J., LARSEN, S. “Marmousi-2: An updated model for the investigation of AVO in structurally complex areas”. In: *SEG Technical Program Expanded Abstracts 2002*, Society of Exploration Geophysicists, pp. 1979–1982, 2002.
- [113] BROUGOIS, A., BOURGET, M., LAILLY, P., et al. “Marmousi, model and data”. In: *EAGE workshop-practical aspects of seismic data inversion*, 1990.
- [114] “SeisSpace® ProMAX® Software”. <https://www.landmark.solutions/SeisSpace-ProMAX>, . Acessado em 2020-04-20.
- [115] ABADI, M., BARHAM, P., CHEN, J., et al. “Tensorflow: A system for large-scale machine learning”. In: *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*, pp. 265–283, 2016.
- [116] “Pytorch: Open Source Deep Learning Library”. <https://pytorch.org/>, . Acessado em 2019-05-30.
- [117] “Open Source Deep Learning Library”. <https://mxnet.apache.org/>. Acessado em 2019-05-30.
- [118] “Open Source Deep Learning Library”. <https://docs.microsoft.com/en-us/cognitive-toolkit/>. Acessado em 2019-05-30.
- [119] FLAMARY, R. “Astronomical image reconstruction with convolutional neural networks”. In: *2017 25th European Signal Processing Conference (EUSIPCO)*, pp. 2468–2472. IEEE, 2017.

- [120] XU, L., REN, J. S., LIU, C., et al. “Deep convolutional neural network for image deconvolution”. In: *Advances in neural information processing systems*, pp. 1790–1798, 2014.
- [121] RUMELHART, D. E., HINTON, G. E., WILLIAMS, R. J. *Learning internal representations by error propagation*. Relatório técnico, California Univ San Diego La Jolla Inst for Cognitive Science, 1985.
- [122] RUMELHART, D. E., HINTON, G. E., WILLIAMS, R. J. “Learning representations by back-propagating errors”, *nature*, v. 323, n. 6088, pp. 533–536, 1986.
- [123] KINGMA, D. P., BA, J. “Adam: A method for stochastic optimization”, *arXiv preprint arXiv:1412.6980*, 2014.
- [124] ZHAO, X., LU, P., ZHANG, Y., et al. “Swell-noise attenuation: A deep learning approach”, *The Leading Edge*, v. 38, n. 12, pp. 934–942, 2019.
- [125] GUI, J., SUN, Z., WEN, Y., et al. “A review on generative adversarial networks: Algorithms, theory, and applications”, *arXiv preprint arXiv:2001.06937*, 2020.
- [126] ZHUANG, F., QI, Z., DUAN, K., et al. “A comprehensive survey on transfer learning”, *arXiv preprint arXiv:1911.02685*, 2019.
- [127] “NumPy: a numerical extension for the computer language Python”. <https://www.numpy.org/>. Acessado em 2019-05-30.
- [128] “Pandas: Python Data Analysis Library”. <https://pandas.pydata.org/>. Acessado em 2019-05-30.
- [129] “Matplotlib: Python Plotting Library”. <https://matplotlib.org/>. Acessado em 2019-05-30.
- [130] “Anaconda: Manage your packages, dependencies and environments”. <https://anaconda.org/>. Acessado em 2019-05-30.
- [131] “Jupyter Notebook”. <http://jupyter.org/>. Acessado em 2019-05-30.

Apêndice A

Anexo

A.1 Ferramentas

Nesta seção, serão apresentadas as tecnologias utilizadas no desenvolvimento desse trabalho. O projeto consistiu-se na implementação de redes neurais artificiais para atenuação de ruídos em dados sísmicos pré-empilhamento.

Neste sentido, faz-se importante destacar que o processamento de otimização em redes neurais pode ser computacionalmente custoso dependendo do tamanho da rede, sendo essencial a escolha certa de linguagem e ferramentas de desenvolvimento a serem utilizadas.

A seção será iniciada com apresentação da linguagem na qual o código foi escrito, junto com algumas das principais bibliotecas utilizadas para processamento, manipulação e visualização dos dados. Em seguida, serão apresentados o gerenciador de dependências e a interface na qual o código foi escrito.

Concluindo a seção, será apresentada a principal ferramenta para o desenvolvimento do projeto, a qual foi usada para implementação das redes neurais artificiais utilizadas.

A.1.1 Python

Python [19] é uma linguagem de programação de alto nível, interpretada, de *script* e orientada a objetos. É uma linguagem muito conhecida e difundida principalmente na área de ciência de dados.

A linguagem escolhida foi criada priorizando-se a legibilidade do código em relação à velocidade de processamento. Em função dessa escolha, a utilização da linguagem pura, ou seja, sem nenhuma biblioteca, não mostra-se eficiente para processamentos mais complexos, como a multiplicação de matrizes, surgindo, assim, algumas bibliotecas para contornar o problema do processamento mais lento.

Numpy

Numpy [127] é uma biblioteca de computação numérica muito utilizado como base para outras bibliotecas, algumas, inclusive, utilizadas neste estudo. Devido à sua implementação diferenciada em C/C++ e FORTRAN, o Numpy se mostra muito melhor em processamento de matrizes em comparação ao Python puro, permitindo, assim, o trabalho com arranjos, vetores e matrizes de N dimensões, que são de grande importância para os ajustes de parâmetros em redes neurais e mantendo a facilidade de uma escrita mais simples e limpa, fornecida pelo Python.

Pandas

Pandas [128] é uma biblioteca de código aberto, de fácil implementação e uso para análise e processamento de dados. É de grande importância nesse projeto, devido à facilidade que provê para controle e manipulação dos dados em formato tabular, maneira pela qual foi feito o controle dos locais dos arquivos utilizados e suas anotações.

Matplotlib

Em razão da pesquisa trabalhar com imagens, uma biblioteca para visualização das mesmas se torna necessária, sendo escolhida para o projeto foi o Matplotlib [129], feita pra criação de gráficos e imagens em uma API de fácil utilização.

A.1.2 Anaconda

O Anaconda [130] é uma distribuição gratuita e de código aberto [6] das linguagens de programação Python para computação científica (ciência de dados, aplicativos de aprendizado de máquina, processamento de dados em larga escala, análise preditiva etc.), que visa simplificar o gerenciamento de pacotes e desdobramento, desenvolvimento.

Jupyter

O Jupyter [131] é uma aplicação baseada na interface web, que permite o desenvolvimento de software, funcionando como uma IDE. Além das vantagens trazidas por funcionar como uma IDE, facilita a geração de uma documentação limpa e visualmente adequado para pesquisa.

A.1.3 Tensorflow

O TensorFlow [20] é uma biblioteca de código aberto para computação numérica usando gráficos de fluxo de dados. Ele foi originalmente desenvolvido pela equipe do *Google Brain* dentro da organização de pesquisa *Machine Intelligence* do *Google* para aprendizado de máquina e pesquisa em redes neurais profundas, mas o sistema é geral o suficiente para ser aplicável em uma ampla variedade de outros domínios também. O TensorFlow é multiplataforma. Ele é executado em CPUs e GPUs - incluindo plataformas móveis e incorporadas - e até mesmo unidades de processamento de tensores (TPUs), que são hardware especializado para realizar cálculos de tensores.