



UNIVERSIDADE FEDERAL DA BAHIA
INSTITUTO DE GEOCIÊNCIAS
CURSO DE GRADUAÇÃO EM GEOFÍSICA

GEO213 – TRABALHO DE GRADUAÇÃO

ESTIMATIVA DE *WAVELETS* DE FASE
ZERO A PARTIR DE TRAÇOS SÍSMICOS
SINTÉTICOS UTILIZANDO A REDE
NEURAL PERCEPTRON
MULTICAMADAS

LORENA DA SILVA OLIVEIRA SANTOS

SALVADOR – BAHIA
DEZEMBRO – 2021

**Estimativa de *wavelets* de fase zero a partir de traços sísmicos sintéticos
utilizando a rede neural perceptron multicamadas**

por

LORENA DA SILVA OLIVEIRA SANTOS

Orientador: Prof. Dr. Aleksandro Guerra Cerqueira

GEO213 – TRABALHO DE GRADUAÇÃO

DEPARTAMENTO DE GEOFÍSICA

DO

INSTITUTO DE GEOCIÊNCIAS

DA

UNIVERSIDADE FEDERAL DA BAHIA

COMISSÃO EXAMINADORA

_____ Dr. Aleksandro Guerra Cerqueira

_____ Dr. Michelângelo Gomes da Silva

_____ MC. Anderson Silva Santos

DATA DA APROVAÇÃO: 06/12/2021

À minha família.

Resumo

A estimativa de *wavelets* sísmicas é um processo fundamental para o método sísmico de reflexão, em particular, para a amarração de poços e inversão sísmica. *Wavelets* são pulsos associados a fonte sísmica, em geral, desconhecidos e seus principais métodos de estimativa podem ser divididos em: estatísticos e determinísticos. Métodos determinísticos utilizam informações da sísmica e de poços para a obtenção do pulso sísmico. Por outro lado, os métodos estatísticos utilizam apenas o traço sísmico para a estimativa. As redes neurais artificiais são algoritmos de aprendizado de máquina que buscam reproduzir o funcionamento do sistema nervoso humano, possuindo alta capacidade de generalização, reconhecimento de padrões e predição de dados. Nesse sentido, as redes neurais Perceptron multicamadas são apresentadas como uma alternativa para estimativa de *wavelets* a partir de traços sísmicos sintéticos, gerados segundo diferentes configurações de espessura, velocidade e densidade de camadas plano-paralelas. Com base nesses modelos, é possível calcular os coeficientes de reflexão, e convolve-los com diferentes tipos de *wavelets* idealizadas. Um primeiro treinamento de uma rede neural é realizado sem considerar a adição de ruídos. Posteriormente, para adicionar um grau de complexidade ao treinamento da rede neural, é somado um conteúdo de ruído ao traço sísmico sintético, de forma a verificar se é possível estimar *wavelets* sísmica com conteúdo de frequência adequado a partir de fatias de traços sísmicos sintéticos.

Abstract

The seismic wavelets estimation is a fundamental step for the seismic reflection method, particularly for well tie and seismic inversion. Wavelets are pulses associated with a seismic source, in general unknown, and their main estimation methods can be split into statistical and deterministic. Deterministic methods use information from seismic and well data to obtain the seismic signal. On the other hand, statistical methods use only the seismic trace for estimation. Artificial neural networks are machine learning algorithms that seek to reproduce the functioning of the human nervous system, having a high capacity for generalization, pattern recognition, and data prediction. In this sense, Multilayer Perceptron neural networks are presented as an alternative for estimating wavelets from synthetic seismic traces generated from different thickness, velocity, and density configurations on plane-parallel layers. From these models, reflection coefficients can be calculated and convolved with different types of idealized wavelets. First, we performed the training of the neural network without considering the addition of noise. Subsequently, to add some complexity to the neural network training, a noise content is added to the synthetic seismic trace to verify if it is possible to estimate seismic wavelets with adequate frequency content from slices of synthetic seismic traces.

Sumário

Resumo	3
Abstract	4
Introdução	13
1 Embasamento Geofísico Teórico	15
1.1 O modelo convolucional	15
1.2 Transformada de Fourier	16
1.3 <i>Wavelet</i> sísmica	18
1.3.1 <i>Wavelets</i> Sísmicas Idealizadas	20
1.4 Estimativa de <i>wavelets</i> sísmicas	22
1.4.1 Análise espectral	22
1.4.2 Autocorrelação	24
1.4.3 Divisão espectral	24
1.4.4 Mínimos Quadrados	25
2 Redes neurais artificiais	27
2.1 O neurônio artificial	27
2.1.1 Funções de ativação	28
2.2 Arquitetura de uma rede neural	31
2.2.1 Arquitetura <i>feedforward</i> de camada simples	32
2.2.2 Arquitetura <i>feedforward</i> de camadas múltiplas	32
2.2.3 Arquitetura recorrente	32
2.3 Redes neurais Perceptron	34
2.3.1 Treinamento da rede Perceptron	34
2.4 Redes neurais Perceptron multicamadas	35
2.4.1 Treinamento da rede Perceptron multicamadas	36
2.4.2 Algoritmo de retropropagação de erros (<i>backpropagation</i>)	36

2.4.3	O otimizador Adam	37
3	Metodologia	39
3.1	Criação dos modelos sintéticos	39
3.2	Validação cruzada	42
3.3	Treinamento da rede neural	44
3.3.1	Configuração	44
3.3.2	Avaliação dos hiperparâmetros	45
3.3.3	Seleção dos melhores hiperparâmetros	46
4	Resultados e Discussão	50
4.1	Análise dos resultados obtidos no primeiro treinamento	50
4.2	Análise dos resultados obtidos no segundo treinamento	51
4.3	Comparações com o método de autocorrelação	65
5	Conclusões	73
	Agradecimentos	75
A	Tabelas de resultados dos treinamentos da rede neural	76
	Referências	79

Lista de Tabelas

3.1	A tabela indica o número das <i>wavelets</i> sísmicas utilizadas para a criação dos 70 mil traços sintéticos.	41
3.2	Tabela que apresenta o percentual adotado para adição de ruído nos traços sísmicos sintéticos, os números absolutos dos traços em que os ruídos foram adicionados e sua porcentagem.	42
4.1	Coefficientes de correlação de Pearson médio, mínimo e máximo dos resultados obtidos pela predição de <i>wavelets</i> utilizando a RNA com ruído e o método da autocorrelação para cada <i>wavelet</i> e para o conjunto total de treinamento. . .	66
4.2	Coefficientes de correlação de Pearson médio, mínimo e máximo dos resultados obtidos pela predição de <i>wavelets</i> utilizando a RNA com ruído e o método da autocorrelação para cada <i>wavelet</i> e para o conjunto total de teste.	72
4.3	Desempenho dos resultados da rede neural para predição de <i>wavelets</i> em traços sintéticos sem ruídos, da rede neural que extrai <i>wavelets</i> sísmicas considerando o ruído e da autocorrelação, também considerando os traços ruidosos, para cada uma das <i>wavelets</i> e para todo o conjunto de dados.	72
A.1	Treinamento da primeira da rede neural Perceptron multicamadas (sem ruídos) para obtenção do melhor valor para a taxa de aprendizado (α). São apresentados os resultados para a função custo (cosseno hiperbólico) e métrica (similaridade dos cossenos), além do tempo de processamento gasto em cada um dos treinamentos. Os valores de <i>batch</i> e épocas foram de 256 e 1000. . .	76
A.2	Treinamento da primeira da rede neural Perceptron multicamadas (sem ruídos) para obtenção do melhor valor para o número de (<i>batches</i>). São apresentados os resultados para a função custo (cosseno hiperbólico) e métrica (similaridade dos cossenos), além do tempo de processamento gasto em cada um dos treinamentos. Os valores de α e de épocas foram de 0.001 e 1000. . .	77

-
- A.3 Treinamento da primeira da rede neural Perceptron multicamadas (sem ruídos) para obtenção do melhor valor para o número de épocas. São apresentados os resultados para a função custo (cosseno hiperbólico) e métrica (similaridade dos cossenos), além do tempo de processamento gasto em cada um dos treinamentos. Os valores de α e do número de *batches* foram de 0.001 e 64. 77
- A.4 Treinamento da segunda rede neural Perceptron multicamadas (com ruídos) para obtenção do melhor valor para o número de épocas. Foram mantidos os valores para o α e o tamanho de *batch* de melhor desempenho do treinamento da RNA com traços sísmicos sintéticos sem ruído. São apresentados os resultados para a função custo (cosseno hiperbólico) e métrica (similaridade dos cossenos), além do tempo de processamento gasto em cada um dos treinamentos. 78

Lista de Figuras

1.1	Representação do modelo convolucional do traço sísmico.	16
1.2	Representação de um número Z , no plano complexo.	18
1.3	Representação de senóides em diferentes fases ($\phi = -\pi/2, 0$ e $\pi/2$). (<i>a</i>) com frequência de 30 Hz; (<i>b</i>) com frequência de 45 Hz e; (<i>c</i>) com frequência de 60 Hz.	19
1.4	Uma mesma <i>wavelet</i> Ricker de 25 Hz de (<i>a</i>) fase zero, (<i>b</i>) fase mínima e (<i>c</i>) fase mista.	20
1.5	Representação das <i>wavelets</i> idealizadas e seus respectivos espectros de frequência: (<i>a</i>) e (<i>b</i>) <i>wavelet</i> Ricker; (<i>c</i>) e (<i>d</i>) <i>wavelet</i> Gabor; (<i>e</i>) e (<i>f</i>) <i>wavelet</i> Ormsby; (<i>g</i>) e (<i>h</i>) <i>wavelet</i> Klauder e; (<i>i</i>) e (<i>j</i>) <i>wavelet</i> Sinc.	23
2.1	Representação de um neurônio artificial.	29
2.2	Funções de ativação parcialmente diferenciáveis: (<i>a</i>) função degrau, (<i>b</i>) função degrau bipolar e (<i>c</i>) função rampa simétrica.	30
2.3	Funções ativação totalmente diferenciáveis: (<i>a</i>) função linear, (<i>b</i>) função sigmóide e (<i>c</i>) função tangente hiperbólica. Em (<i>b</i>) e (<i>c</i>) também é mostrado como o parâmetro β influencia na curva.	31
2.4	Esquematização de uma rede <i>feedforward</i> simples.	32
2.5	Esquematização de uma rede <i>feedforward</i> de camadas múltiplas.	33
2.6	Esquematização de uma rede recorrente. Fonte: da Silva et al. (2016).	33
2.7	Ilustração do processo de treinamento de uma rede Perceptron. Fonte: da Silva et al. (2016).	35
3.1	Fluxograma da metodologia implementada no trabalho.	40
3.2	Exemplo de modelo e traços sísmicos sintéticos gerado pela metodologia: (<i>a</i>) é o modelo de impedância acústicas, (<i>b</i>) o modelo da função refletividade, (<i>c</i>) o traço sísmico sintético, (<i>d</i>) o traço sísmico sintético com a adição de ruído e (<i>e</i>) a <i>wavelet</i> utilizada na geração dos traços sintéticos.	43

3.3	Ilustração da rede neural utilizada para predição das <i>wavelets</i> sísmicas. A camada de <i>input</i> é composta por 300 amostras do traço sísmicos, a primeira, segunda e terceira camada oculta têm 300, 300 e 200 neurônios respectivamente, e o <i>output</i> é a <i>wavelet</i> sísmica com 97 amostras.	44
3.4	Análise dos hiperparâmetros utilizados para o treinamento da rede neural utilizada para a predição de <i>wavelets</i> utilizando traços sísmicos sintéticos sem ruídos: (a), (c) e (e) são os gráficos dos valores de função custo (<i>logcosh</i>) e (b), (d) e (f) são os gráficos de resultados da métrica utilizada para a análise do treinamento para os valores de α , tamanho de <i>batch</i> e número de épocas, respectivamente.	48
3.5	Resultados dos treinamentos feitos na segunda etapa da RNA — com adição de ruídos aos traços sísmicos: (a) é a variação dos valores da função custo e (b) os valores da métrica de cada treinamento para o número de épocas. . . .	49
4.1	Resultado da predição de uma <i>wavelet</i> Ricker pela primeira RNA: (a), (c) e (e) são as <i>wavelets</i> observada, calculada e o resíduo; enquanto (b), (d) e (f) são os respectivos espectros de frequência. O coeficiente de correlação entre as <i>wavelets</i> observada e calcula é de 0,993.	52
4.2	Resultado da predição de uma <i>wavelet</i> Gabor pela primeira RNA: (a), (c) e (e) são as <i>wavelets</i> observada, calculada e o resíduo; enquanto (b), (d) e (f) são os respectivos espectros de frequência. O coeficiente de correlação entre as <i>wavelets</i> observada e calcula é de 0,998.	53
4.3	Resultado da predição de uma <i>wavelet</i> Ormsby pela primeira RNA: (a), (c) e (e) são as <i>wavelets</i> observada, calculada e o resíduo; enquanto (b), (d) e (f) são os respectivos espectros de frequência. O coeficiente de correlação entre as <i>wavelets</i> observada e calcula é de 0,993.	54
4.4	Resultado da predição de uma <i>wavelet</i> Klauder pela primeira RNA: (a), (c) e (e) são as <i>wavelets</i> observada, calculada e o resíduo; enquanto (b), (d) e (f) são os respectivos espectros de frequência. O coeficiente de correlação entre as <i>wavelets</i> observada e calcula é de 0,989.	55
4.5	Resultado da predição de uma <i>wavelet</i> Sinc pela primeira RNA: (a), (c) e (e) são as <i>wavelets</i> observada, calculada e o resíduo; enquanto (b), (d) e (f) são os respectivos espectros de frequência. O coeficiente de correlação entre as <i>wavelets</i> observada e calcula é de 0,997.	56
4.6	Traços sísmicos dos resultados apresentados para as <i>wavelets</i> : (a) Ricker (Figura 4.1), (b) Gabor (Figura 4.2), (c) Ormsby (Figura 4.3), (d) Klauder (Figura 4.4), e (e) Sinc (Figura 4.5).	57

4.7	Resultado da predição de uma <i>wavelet</i> Ricker pela segunda RNA: <i>(a)</i> , <i>(c)</i> e <i>(e)</i> são as <i>wavelets</i> observada, calculada e o resíduo; enquanto <i>(b)</i> , <i>(d)</i> e <i>(f)</i> são os respectivos espectros de frequência. O coeficiente de correlação entre as <i>wavelets</i> observada e calcula é de 0,989.	59
4.8	Resultado da predição de uma <i>wavelet</i> Gabor pela segunda RNA: <i>(a)</i> , <i>(c)</i> e <i>(e)</i> são as <i>wavelets</i> observada, calculada e o resíduo; enquanto <i>(b)</i> , <i>(d)</i> e <i>(f)</i> são os respectivos espectros de frequência. O coeficiente de correlação entre as <i>wavelets</i> observada e calcula é de 0,993.	60
4.9	Resultado da predição de uma <i>wavelet</i> Ormsby pela segunda RNA: <i>(a)</i> , <i>(c)</i> e <i>(e)</i> são as <i>wavelets</i> observada, calculada e o resíduo; enquanto <i>(b)</i> , <i>(d)</i> e <i>(f)</i> são os respectivos espectros de frequência. O coeficiente de correlação entre as <i>wavelets</i> observada e calcula é de 0,944.	61
4.10	Resultado da predição de uma <i>wavelet</i> Klauder pela segunda RNA: <i>(a)</i> , <i>(c)</i> e <i>(e)</i> são as <i>wavelets</i> observada, calculada e o resíduo; enquanto <i>(b)</i> , <i>(d)</i> e <i>(f)</i> são os respectivos espectros de frequência. O coeficiente de correlação entre as <i>wavelets</i> observada e calcula é de 0,952.	62
4.11	Resultado da predição de uma <i>wavelet</i> Sinc pela segunda RNA: <i>(a)</i> , <i>(c)</i> e <i>(e)</i> são as <i>wavelets</i> observada, calculada e o resíduo; enquanto <i>(b)</i> , <i>(d)</i> e <i>(f)</i> são os respectivos espectros de frequência. O coeficiente de correlação entre as <i>wavelets</i> observada e calcula é de 0,987.	63
4.12	Traços sísmicos ruidosos utilizados na predição dos resultados apresentados para as <i>wavelets</i> : <i>(a)</i> Ricker (Figura 4.7) com 0% de ruído, <i>(b)</i> Gabor (Figura 4.8) com 10% de ruído, <i>(c)</i> Ormsby (Figura 4.9) com 10% de ruído, <i>(d)</i> Klauder (Figura 4.10) com 10% de ruído, e <i>(e)</i> Sinc (Figura 4.11) com 10% de ruído.	64
4.13	Comparação de resultados de uma <i>wavelet</i> sísmica do tipo Ricker: <i>(a)</i> Comparação entre a <i>wavelet</i> observada e calculada utilizando a rede neural Perceptron multicamadas, <i>(b)</i> <i>wavelet</i> sísmica observada, <i>(c)</i> <i>wavelet</i> sísmica extraída a partir da RNA, <i>(d)</i> <i>wavelet</i> sísmica extraída a partir da técnica de autocorrelação, <i>(e)</i> Comparação entre a <i>wavelet</i> observada e calculada através o método de autocorrelação, <i>(f)</i> espectro de frequência da <i>wavelet</i> observada, <i>(g)</i> espectro de frequência da <i>wavelet</i> calculada pela RNA e <i>(h)</i> espectro de frequência calculada a partir da técnica de autocorrelação.	67

-
- 4.14 Comparação de resultados de uma *wavelet* sísmica do tipo Gabor: (a) Comparação entre a *wavelet* observada e calculada utilizando a rede neural Perceptron multicamadas, (b) *wavelet* sísmica observada, (c) *wavelet* sísmica extraída a partir da RNA, (d) *wavelet* sísmica extraída a partir da técnica de autocorrelação, (e) Comparação entre a *wavelet* observada e calculada através o método de autocorrelação, (f) espectro de frequência da *wavelet* observada, (g) espectro de frequência da *wavelet* calculada pela RNA e (h) espectro de frequência calculada a partir da técnica de autocorrelação. 68
- 4.15 Comparação de resultados de uma *wavelet* sísmica do tipo Ormsby: (a) Comparação entre a *wavelet* observada e calculada utilizando a rede neural Perceptron multicamadas, (b) *wavelet* sísmica observada, (c) *wavelet* sísmica extraída a partir da RNA, (d) *wavelet* sísmica extraída a partir da técnica de autocorrelação, (e) Comparação entre a *wavelet* observada e calculada através o método de autocorrelação, (f) espectro de frequência da *wavelet* observada, (g) espectro de frequência da *wavelet* calculada pela RNA e (h) espectro de frequência calculada a partir da técnica de autocorrelação. 69
- 4.16 Comparação de resultados de uma *wavelet* sísmica do tipo Klauder: (a) Comparação entre a *wavelet* observada e calculada utilizando a rede neural Perceptron multicamadas, (b) *wavelet* sísmica observada, (c) *wavelet* sísmica extraída a partir da RNA, (d) *wavelet* sísmica extraída a partir da técnica de autocorrelação, (e) Comparação entre a *wavelet* observada e calculada através o método de autocorrelação, (f) espectro de frequência da *wavelet* observada, (g) espectro de frequência da *wavelet* calculada pela RNA e (h) espectro de frequência calculada a partir da técnica de autocorrelação. 70
- 4.17 Comparação de resultados de uma *wavelet* sísmica do tipo Sinc: (a) Comparação entre a *wavelet* observada e calculada utilizando a rede neural Perceptron multicamadas, (b) *wavelet* sísmica observada, (c) *wavelet* sísmica extraída a partir da RNA, (d) *wavelet* sísmica extraída a partir da técnica de autocorrelação, (e) Comparação entre a *wavelet* observada e calculada através o método de autocorrelação, (f) espectro de frequência da *wavelet* observada, (g) espectro de frequência da *wavelet* calculada pela RNA e (h) espectro de frequência calculada a partir da técnica de autocorrelação. 71

Introdução

No contexto da sismica de reflexão, a obtenção de *wavelets* sísmicas é um processo fundamental para as etapas de processamento de dados, amarração de poços, inversão sísmica e interpretação dos dados adquiridos. Entretanto, como a *wavelet* é um pulso associado à fonte, em geral, desconhecido e é registrado nos sismogramas convolvido com a resposta impulsiva da Terra (refletividade), também desconhecida, esse processo não é trivial (Brown et al., 1988).

Os métodos de estimativa de *wavelet* sísmicas podem ser classificados em três tipos: determinísticos, estatísticos e analíticos. Os métodos determinísticos são aqueles que utilizam sismogramas e dados de poço para estimar os coeficientes de reflexão e assim, fazer a predição das *wavelets* através de processos de filtragem e/ou deconvolução (Yilmaz, 2001; Oldenburg et al., 1981). Por outro lado, métodos estatísticos utilizam apenas as informações do traço sísmico para sua estimativa (Cui e Margrave, 2014; Acquaviva et al., 2015). Ademais, os métodos analíticos são aqueles que utilizam *wavelets* idealizadas como assinatura da fonte do dado.

Algoritmos de aprendizado de máquina são técnicas designadas para extrair informações dos dados para automatizar uma determinada atividade ou extrair um certo padrão de interesse. Nesse contexto, as Redes Neurais Artificiais (RNA) são algoritmos de aprendizado de máquina cuja popularidade têm crescido nos últimos anos. De acordo com da Silva et al. (2016), a primeira publicação relacionada a redes neurais é atribuída a McCulloch e Pitts (1943), que realizaram uma modelagem matemática inspirada no neurônio biológico, o que resultou na primeira concepção de "neurônio artificial". As redes neurais procuram reproduzir o funcionamento do sistema nervoso central biológico e vêm se mostrando eficazes na solução de problemas não-lineares. Existem diversas arquiteturas de redes neurais que permitem lidar com soluções de problemas supervisionados — quando as medidas e as respostas são fornecidas (regressão e classificação) —, ou para encontrar padrões em conjunto de dados onde os rótulos são ausentes utilizando a abordagem não-supervisionada — análise de agrupamentos (Bishop, 2007; James et al., 2013; da Silva et al., 2016).

No campo da geofísica, redes neurais artificiais já foram empregadas com sucesso na

interpretação de dados sísmicos para detecção de fácies sísmicas, falhas e rastreamento de horizontes sísmicos (Wrona et al., 2018; Zheng et al., 2019; Peters et al., 2019), para predição de resistividade e condutividade termal (Calderón-Macías et al., 2000; Goutorbe et al., 2006), classificação de fácies utilizando perfis geofísicos de poços (Dubois et al., 2007), na filtragem e controle de dados no processamento sísmico (Mejri e Bekara, 2020; Li et al., 2018) e localização de alvos eletromagnéticos (Poulton, Sternberg e Glass, 1992). Outras diversas aplicações de redes neurais em problemas geofísicos foram apresentadas por Poulton (2001).

Aproveitando o fato de que redes neurais são algoritmos que lidam muito bem com problemas não-lineares devido a sua configuração, esse presente trabalho de graduação tem como objetivos:

- Gerar diversas configurações sintéticas de traços sísmicos sem ruído e com ruídos, que foram convolvidos com *wavelets* sísmicas idealizadas para serem utilizados como dados de entrada de um algoritmo de aprendizado de máquina supervisionado;
- Treinar uma rede neural artificial, particularmente, a rede perceptron multicamadas, para estimar de forma satisfatória as *wavelets* sísmicas a partir de traços sísmicos, utilizando boas práticas de aprendizado de máquina;
- Avaliar o espectro de frequência dos resultados obtidos pela RNA e a *wavelet* esperada;
- Avaliar o desempenho da rede neural treinada com a adição de ruídos em um percentual dos traços sísmicos;
- Extrair *wavelets* sísmicas através de uma técnica já conhecida, como a autocorrelação do traço sísmico; e
- Comparar as *wavelets* estimadas pela RNA com a técnica convencional.

Este trabalho está organizado em cinco capítulos: o Capítulo 1, que apresenta o conceito teórico das *wavelets* sísmicas e o embasamento geofísico teórico necessário para sua análise, além de abordar algumas técnicas de estimativa do pulso sísmico que são habitualmente empregadas; Capítulo 2 que trata da definição de redes neurais artificiais, discorrendo sobre seus elementos, classificações, técnicas de otimizações e limitações; Capítulo 3 que apresenta a metodologia empregada no trabalho, bem como o fluxograma de geração de dados de entrada e o processo de treinamento dos parâmetros da rede; Capítulo 4 que apresenta os resultados obtidos e faz análises e comparações com uma das técnicas de obtenção de *wavelets* sísmicas convencionais; e o Capítulo 5 que mostra as conclusões e análises do que se foi apresentado nos resultados. De forma adicional, serão abordadas sugestões de melhorias, na metodologia proposta, que devem ser implementadas em trabalhos futuros relacionadas ao escopo desse trabalho.

Capítulo 1

Embasamento Geofísico Teórico

As *wavelets* sísmicas são pulsos associados à fonte cujo conhecimento é fundamental, principalmente, para o processamento, interpretação e inversão de dados acústicos. Esse capítulo tem como objetivo apresentar o modelo convolucional do traço sísmico (Seção 1.1), fazer uma revisão teórica da transformada discreta de Fourier (Seção 1.2), definir e discutir os aspectos teóricos das assinaturas sísmicas (Seção 1.3), e por fim, apresentar os métodos para extração de *wavelets*, já estabelecidos na literatura, que serão utilizados para comparação de resultados (Seção 1.4).

1.1 O modelo convolucional

A forma mais simples de se representar o traço sísmico é através do modelo convolucional, que define o traço sísmico como uma convolução da resposta impulsiva da terra com a assinatura da fonte com um componente de ruído aleatório incorporado (Yilmaz, 2001). Matematicamente, o modelo convolucional é dado por (Eq. 1.1):

$$\mathbf{s}(t) = \mathbf{r}(t) * \mathbf{w}(t) + \mathbf{n}(t) \quad (1.1)$$

onde $\mathbf{s}(t)$ é o traço sísmico, $\mathbf{r}(t)$ é a resposta impulsiva da terra ou série de coeficientes de reflexão, $\mathbf{w}(t)$ é a assinatura da fonte ou *wavelet* sísmica, $\mathbf{n}(t)$ é um ruído aleatório adicionado e $*$ representa a operação de convolução.

O coeficiente de reflexão, para um raio incidente de forma normal, pode ser expresso pela seguinte expressão (Eq. 1.2):

$$R_i = \frac{\rho_{i+1}v_{i+1} - \rho_i v_i}{\rho_{i+1}v_{i+1} + \rho_i v_i} = \frac{Z_{i+1} - Z_i}{Z_{i+1} + Z_i}, \quad (1.2)$$

em que ρ_i , v_i , Z_i , e ρ_{i+1} , v_{i+1} , Z_{i+1} são os valores da densidade, velocidade e impedância acústica referente a uma camada i sobreposta a uma camada $i + 1$. A Figura 1.1 exibe uma

representação do modelo.

Segundo (Yilmaz, 1987), a elaboração do modelo convolucional da traço sísmico segue as seguintes premissas:

- O interior da terra é representado por um modelo de camadas plano-paralelas com velocidade de propagação das ondas sísmicas constantes nas camadas;
- A fonte sísmica produz somente ondas compressoriais com incidência normal nas camadas; e
- O pulso sísmico é dito ser estacionário, ou seja, a sua forma de onda não altera durante o seu trajeto na subsuperfície.

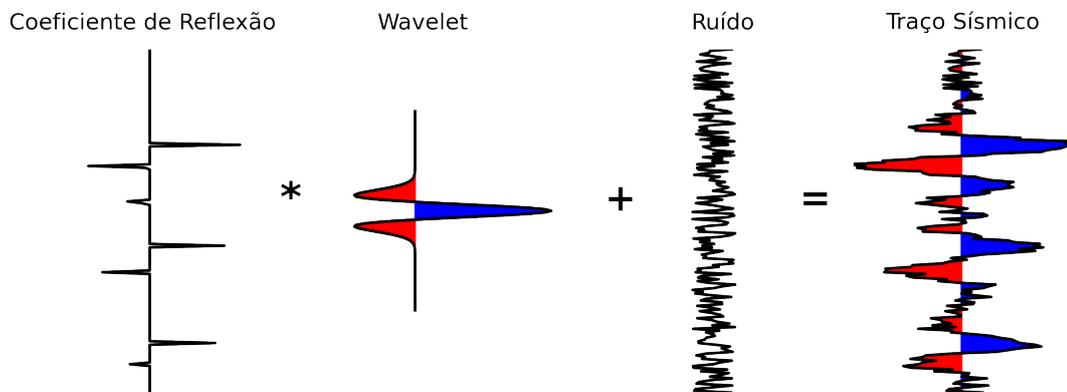


Figura 1.1: Representação do modelo convolucional do traço sísmico.

1.2 Transformada de Fourier

Tanto as *wavelets* quanto os traços sísmicos são séries temporais, logo, são sinais representados no domínio do tempo. O sinal sísmico pode ser representado através de um somatório infinito de senos e cossenos, ou seja, através de funções trigonométricas que carregam informações sobre a amplitude, frequência e fase do sinal original. A decomposição do sinal através dessas funções trigonométricas e a sua representação no domínio da frequência se dá através da transformada de Fourier. A série temporal que foi levada para o domínio da frequência pode ser trazida para o domínio do tempo através da transformada inversa de Fourier (Yilmaz, 1987).

Portanto, para definir a contribuição de cada harmônico no sinal original, o dado registrado no domínio do tempo é levado para o domínio da frequência (Simm e Bacon, 2014). Essa mudança de domínio pode ser representada como:

$$x(t) \Leftrightarrow X(\omega)$$

onde $x(t)$ é uma série temporal e $X(\omega)$ é a função no domínio da frequência.

Matematicamente, a transformada de Fourier de uma função contínua é correspondente a seguinte expressão (Eq. 1.3):

$$X(\omega) = \int_{-\infty}^{\infty} x(t) \cdot e^{-i\omega t} dt = \int_{-\infty}^{\infty} x(t) \cdot e^{-i2\pi f t} dt \quad (1.3)$$

onde, t é o tempo e f , a frequência.

Em geral, as operações sobre sinais, a nível computacional, não são realizadas de forma contínua, e sim como uma série de pontos discretos $\mathbf{x} = \{x_0, x_1, \dots, x_{N-1}\}$, dentro de um intervalo $[t_0, t_{N-1}]$ e com taxa de amostragem Δt . Logo, a transformada discreta de Fourier (TDF) é dada por (Eq. 1.4):

$$X_n = \sum_{n=0}^{N-1} x_n \cdot e^{-\frac{i2\pi f t}{N}} \quad (1.4)$$

Do mesmo modo, é possível obter a transformada inversa do sinal, através da expressão:

$$x_n = \frac{1}{N} \sum_{n=0}^{N-1} X_n \cdot e^{\frac{i2\pi f t}{N}} \quad (1.5)$$

Considerando o termo $\frac{2\pi f t}{N}$ como uma variável b_n , pode-se reescrever a Equação 1.4 como:

$$X_k = \sum_{n=0}^{N-1} x_n \cdot e^{-ib_n} \quad (1.6)$$

Segundo a fórmula de Euler, o termo exponencial pode ser expresso por:

$$e^{ix} = \cos(x) + i\text{sen}(x) \quad (1.7)$$

De forma que, a Equação 1.6 seja expressa por:

$$X_k = A_k + iB_k \quad (1.8)$$

onde,

$$A_k = \sum_{n=0}^{N-1} x_n \cdot \cos(b_n) = \sum_{n=0}^{N-1} x_n \cdot \cos\left(\frac{2\pi f t}{N}\right) \quad (1.9)$$

$$B_k = \sum_{n=0}^{N-1} -x_n \cdot \text{sen}(b_n) = \sum_{n=0}^{N-1} -x_n \cdot \text{sen}\left(\frac{2\pi f t}{N}\right) \quad (1.10)$$

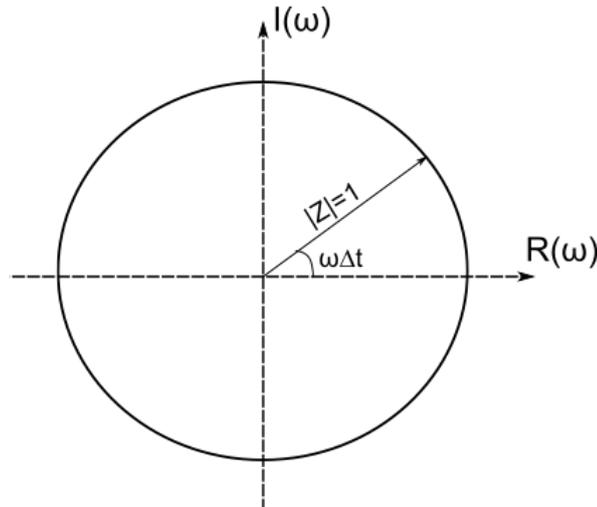


Figura 1.2: Representação de um número Z , no plano complexo.

Dessa forma, analisando o círculo unitário no plano complexo (Fig. 1.2), pode-se obter a amplitude e a fase de um vetor contido no mesmo através das expressões:

$$|X_k| = \sqrt{A_k^2 + B_k^2} \quad (1.11)$$

$$\theta = \tan^{-1} \left(\frac{B_k}{A_k} \right) \quad (1.12)$$

1.3 *Wavelet* sísmica

Wavelets são séries temporais associadas a uma fonte sísmica, e é um aspecto fundamental no processamento de dados e interpretação sísmica. Dessa forma, se faz necessário avaliar os elementos característicos de uma *wavelet*, tais como sua frequência, fase e amplitude.

A **amplitude** (A) de uma *wavelet* corresponde ao valor do sinal num dado ponto, ou seja, corresponde ao quanto o sinal se diferencia de zero, podendo ser positiva ou negativa.

A **frequência** (f) de um pulso representa o número de oscilações por unidade de tempo. Para *wavelets*, costuma-se descrever uma banda de frequências, ou seja, as frequências que estão dominando o sinal. A avaliação dessa banda de frequências é feita através da análise do espectro de amplitude versus frequência, que pode ser obtido à partir da transformada discreta de Fourier (TDF) (Eq. 1.11), e é definida como todos os componentes de frequência presentes até a metade da amplitude máxima do espectro obtido.

A **fase** (ϕ) de um sinal pode ser definida como a medida da posição de uma onda senoidal em relação a um ponto de referência que pode ser tomado com o tempo $t = 0$. De forma similar a frequência, pode-se estimar o espectro de fase de sinal sísmico através da

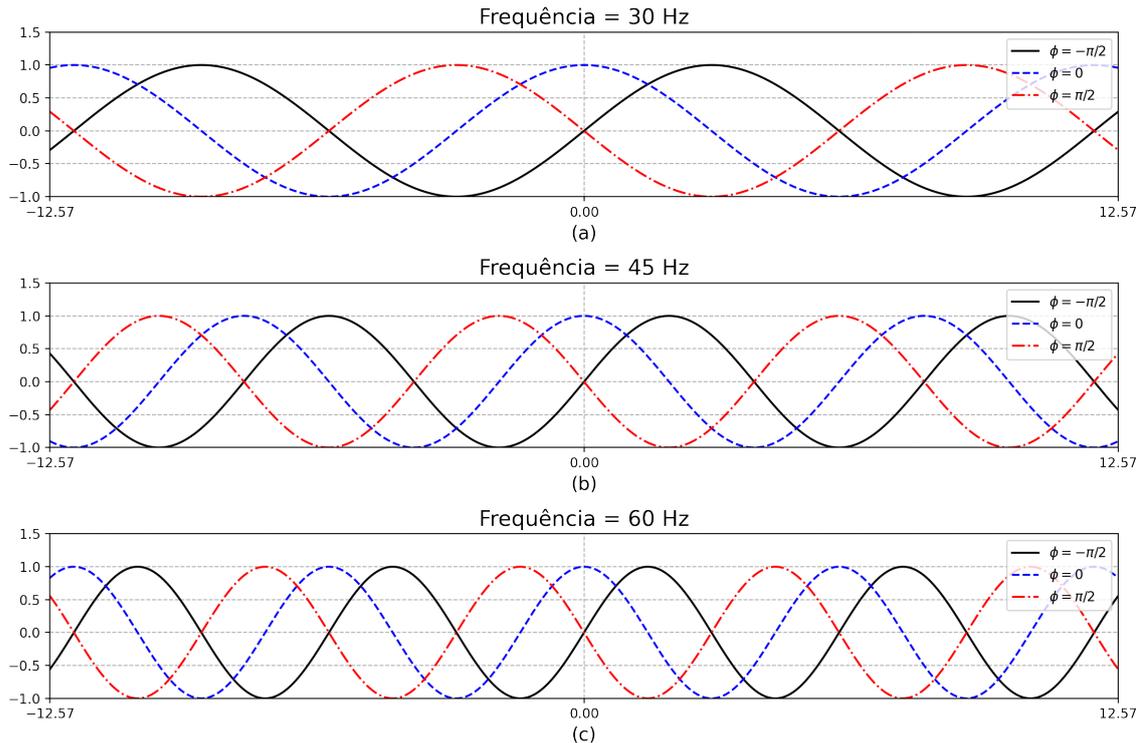


Figura 1.3: Representação de senóides em diferentes fases ($\phi = -\pi/2$, 0 e $\pi/2$). (a) com frequência de 30 Hz; (b) com frequência de 45 Hz e; (c) com frequência de 60 Hz.

TDF, como mostra a Eq. 1.12. A Figura 1.3 mostra como a frequência e a fase podem influenciar no comportamento de uma senoide de amplitude máxima A , dada pela equação (Eq. 1.13):

$$y(t) = A \cos(2\pi ft + \phi) \quad (1.13)$$

Ainda discutindo acerca da fase, uma *wavelet* sísmica pode ser classificada como:

- **Fase zero:** quando sua amplitude máxima se encontra no instante $t = 0$. *Wavelets* de fase zero, também chamadas de *wavelets* não-causais, possuem simetria em relação ao instante $t = 0$ e, em geral, são as mais utilizadas para interpretação sísmica;
- **Fase mínima:** *wavelets* cujo sinal apenas é registrado a partir do instante $t = 0$, podendo também serem chamadas de *wavelets* causais. As assinaturas da fonte, em geral, são *wavelets* de fase mínima; e
- **Fase mista:** *Wavelets* de fase mista são aquelas que não se enquadram nem como de fase zero, nem como de fase mínima. Costumam ser registradas antes do instante $t = 0$ mas não possuem simetria em relação à ele.

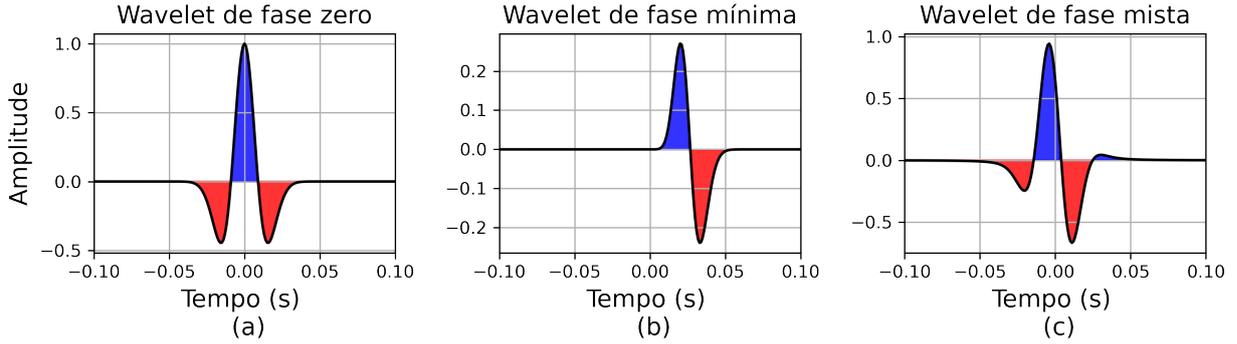


Figura 1.4: Uma mesma *wavelet* Ricker de 25 Hz de (a) fase zero, (b) fase mínima e (c) fase mista.

A Figura 1.4 mostra uma representação esquemática de cada uma das assinaturas sísmicas definidas acima. Apesar da estimativa da *wavelet* sísmica ser complexa, existem modelos de assinaturas sísmicas.

1.3.1 *Wavelets* Sísmicas Idealizadas

Existem diversos tipos de assinaturas sísmicas idealizadas que são amplamente utilizadas em amarrações de sísmica-poço (Simm e Bacon, 2014), onde podem ser citadas as *wavelets* Ricker, Gabor, Ormsby, Sinc e Klauder, apresentadas na Figura 1.5 com seus respectivos espectros de frequência. Uma descrição mais detalhada dessas *wavelets* será fornecida nas subseções a seguir.

Wavelet Ricker

A *wavelet* Ricker — segunda derivada de uma função Gaussiana — é uma das mais simples e consiste em um pulso de fase zero com um pico central e dois pequenos lóbulos laterais. É também conhecida como *wavelet* "chapéu mexicano" por sua forma. A expressão matemática utilizada para a obtenção de uma *wavelet* Ricker é dada por (Eq. 1.14):

$$w(t) = (1 - 2\pi^2 f_D^2 t^2) e^{-\pi^2 f_D^2 t^2} \quad (1.14)$$

onde, f_D é a frequência dominante da assinatura sísmica. A *wavelet* Ricker é comumente utilizada como o primeiro pulso sísmico de fase zero escolhido na geração de traços sísmicos sintéticos para amarração sísmica devido a sua simplicidade (Simm e Bacon, 2014).

Wavelet Gabor

A *wavelet* Gabor é um pulso de fase zero similar a *wavelet* Morlet e, assim como a Ricker, apenas necessita de uma frequência dominante f_D . Matematicamente,

$$w(t) = e^{(-2f_D^2 t^2)} \cos(2\pi f_D t). \quad (1.15)$$

De acordo com Ji e Yan (2017), Morlet et al. (1982) introduziram pela primeira vez as *wavelets* Gabor para o processamento de dados sísmicos.

Wavelet Sinc

A função *Sinc* (seno cardinal) é definida pela Equação 1.16. A *wavelet* Sinc possui um formato retangular no domínio da frequência fazendo com que ela seja considerada um filtro passa-baixa ideal.

$$w(t) = \text{sinc}(t) = \frac{\sin(2\pi f_D t)}{2\pi f_D t} \quad (1.16)$$

Wavelet Ormsby

As *wavelets* Ormsby também são de fase zero mas se diferenciam por possuir um conjunto de frequências e seu espectro de amplitudes apresentar uma forma trapezoidal. Sua expressão é dada por:

$$w(t) = \frac{\pi f_1^2}{f_2 - f_1} \text{sinc}^2(\pi f_1 t) - \frac{\pi f_2^2}{f_2 - f_1} \text{sinc}^2(\pi f_2 t) - \frac{\pi f_3^2}{f_4 - f_3} \text{sinc}^2(\pi f_3 t) + \frac{\pi f_4^2}{f_4 - f_3} \text{sinc}^2(\pi f_4 t) \quad (1.17)$$

onde f_1 , f_2 , f_3 e f_4 são as frequências principais da *wavelet* e a expressão *sinc* é a função seno cardinal apresentada na subseção anterior.

Wavelet Klauder

A *wavelet* Klauder é definida como a autocorrelação de um sinal *sweep* linear utilizado em Vibroseis (Ryan, 1994). Assim como a Ormsby, o sinal Klauder não possui uma frequência dominante, mas sim, utiliza uma frequência terminal baixa f_1 e uma frequência terminal alta f_2 , além da duração do sinal T . Sua expressão é dada por:

$$w(t) = \text{Re} \left[\frac{\sin(\pi k t (T - t))}{\pi k t e^{2\pi i f_0 t}} \right] \quad (1.18)$$

onde os termos k e f_0 representam, respectivamente, a taxa de mudança da frequência em relação ao tempo e a frequência média. Matematicamente, k e f_0 são expressos como (Eq. 1.19 e 1.20):

$$k = \frac{f_1 - f_2}{T} \quad (1.19)$$

$$f_0 = \frac{f_1 + f_2}{2}. \quad (1.20)$$

1.4 Estimativa de *wavelets* sísmicas

A estimativa de *wavelets* é uma etapa fundamental na sísmica de reflexão, pois impacta diretamente os processos de amarração de poços e inversão sísmica. As *wavelets* podem ser estimadas através de três métodos:

- **Métodos estatísticos:** Os métodos estatísticos são aqueles que utilizam apenas o traço sísmico para estimativa de *wavelets*, como por exemplo, deconvoluções, autocorrelação cruzada, entre outros (Cui e Margrave, 2014; Acquaviva et al., 2015);
- **Métodos determinísticos:** os métodos determinísticos utilizam o traço sísmico e informações provenientes de perfilagens, como os coeficientes de reflexão, por exemplo, as filtragens e as deconvoluções (Yilmaz, 2001; Oldenburg et al., 1981); e
- **Métodos analíticos:** métodos analíticos representam o pulso sísmico do dado como uma das *wavelets* idealizadas. Em geral, a *wavelet* Ricker é uma das mais empregadas.

As subseções 1.4.1, 1.4.2, 1.4.3 e 1.4.4 têm como objetivo apresentar algumas técnicas utilizadas para a extração do pulso sísmico.

1.4.1 Análise espectral

A análise espectral é a maneira mais simples de estimar uma *wavelet*. Para tal, utiliza-se apenas o traço sísmico, logo, ela se encaixa no conceito de método estatístico.

Segundo Yilmaz (2001), assumindo que a refletividade é um processo aleatório, ou seja, o conhecimento das reflexões iniciais não auxilia na predição de reflexões mais profundas, pode-se atribuir as características da *wavelet* ao sismograma, logo, suas autocorrelações e espectros de frequência são semelhantes. Assim, é possível definir frequências dominantes no espectro do traço sísmico e, com base na sua forma e frequências, definir uma *wavelet* idealizada sintética cujo espectro se aproxime ao do traço. Apesar de funcionar em alguns

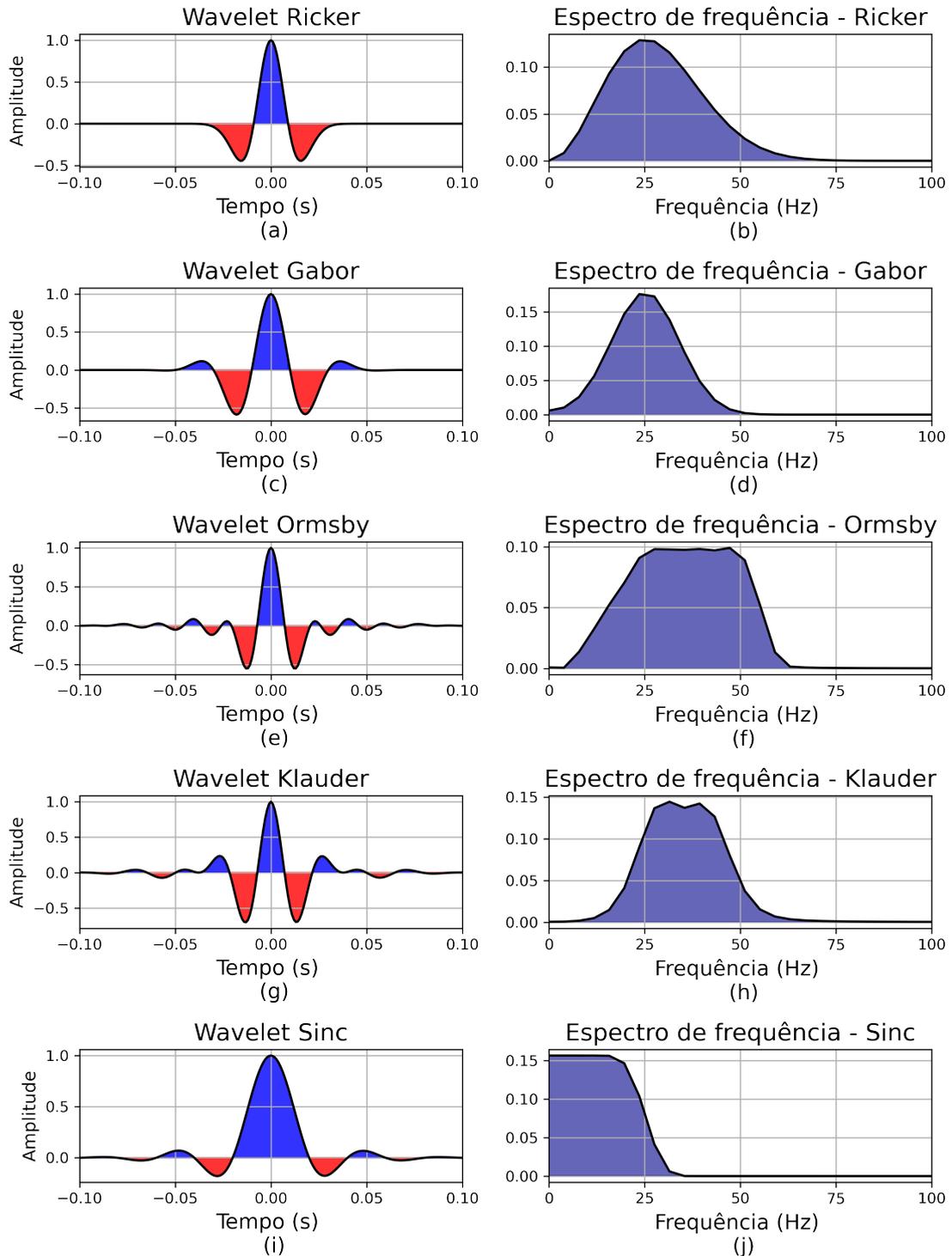


Figura 1.5: Representação das *wavelets* idealizadas e seus respectivos espectros de frequência: (a) e (b) *wavelet* Ricker; (c) e (d) *wavelet* Gabor; (e) e (f) *wavelet* Ormsby; (g) e (h) *wavelet* Klauder e; (i) e (j) *wavelet* Sinc.

casos, esse método nem sempre é adotado, já que diferentes tipos de *wavelets* podem possuir espectros de frequências similares.

1.4.2 Autocorrelação

Ainda assumindo que a refletividade é um processo aleatório e atribuindo as particularidades da *wavelet* ao sismograma, outro método estatístico de estimar a assinatura da fonte é fazer a correlação cruzada do traço sísmico com ele mesmo, ou seja, a autocorrelação do traço.

Para Rosa (2018), ao se correlacionar o traço com ele mesmo, o resultado é um sinal de fase zero derivado do espectro de amplitude do traço sísmico. Matematicamente, a autocorrelação pode ser expressa por (Eq. 1.21):

$$C_k = \sum_{j=-\infty}^{\infty} b_{j+k} b_j^* \quad (1.21)$$

onde, C_k é a autocorrelação e b^* representa o conjugado de b .

1.4.3 Divisão espectral

A divisão espectral é um método determinístico de estimativa de *wavelets*. Assumindo um modelo mais simples da Equação 1.1, pode-se representar um traço sísmico $s(t)$ como apenas a convolução de uma resposta impulsiva da terra $r(t)$ com o pulso $w(t)$. Sendo assim, pode-se expressar a Equação 1.1 como (Eq. 1.22):

$$s(t) = w(t) * r(t) \quad (1.22)$$

Usando uma das propriedades da transformada de Fourier que afirma que a convolução entre dos sinais no domínio do tempo equivale a multiplicação dos mesmos no domínio da frequência, isso nos permite escrever a Eq. 1.22 no domínio da frequência como (Eq. 1.23):

$$S(\omega) = W(\omega) \cdot R(\omega) \quad (1.23)$$

onde, $S(\omega)$, $W(\omega)$, $R(\omega)$ representam, respectivamente, o traço sísmico, a *wavelet* e a série de coeficientes de reflexão no domínio da frequência.

Dessa forma, é possível admitir que:

$$W(\omega) = S(\omega)/R(\omega) \quad (1.24)$$

Logo, pode-se estimar $w(t)$ ao fazer a Transforma Inversa de Fourier do componente $W(\omega)$.

Bianco (2016) salienta que esse método pode falhar na presença de ruídos, múltiplas, efeitos da migração, entre outros, podendo não funcionar bem ao ser aplicado em dados reais.

1.4.4 Mínimos Quadrados

Seguindo com os métodos determinísticos, o método dos mínimos quadrados parte da suposição que é possível representar o modelo convolucional do traço sísmico de maneira matricial por:

$$\mathbf{s} = \mathbf{R}\mathbf{w} \quad (1.25)$$

onde, \mathbf{s} e \mathbf{w} são vetores que representam o traço sísmico e a *wavelet*, respectivamente. A variável \mathbf{R} é uma matriz do tipo Toeplitz construída a partir dos coeficientes de reflexão, de forma que:

$$\mathbf{R} = \begin{bmatrix} r_0 & r_{-1} & r_{-2} & \dots & \dots & r_{-n+1} \\ r_1 & r_0 & r_{-1} & \ddots & & \vdots \\ r_2 & r_1 & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & r_{-1} & r_{-2} \\ \vdots & & \ddots & r_1 & r_0 & r_{-1} \\ r_{n-1} & \dots & \dots & r_2 & r_1 & r_0 \end{bmatrix}$$

Logo, por mínimos quadrados, busca-se minimizar a expressão (Eq. 1.26):

$$\min |\mathbf{R}\mathbf{w} - \mathbf{s}|^2 \quad (1.26)$$

de forma que, a solução pode ser obtida por (Eq. 1.27):

$$\hat{w} = (\mathbf{R}^T \mathbf{R})^{-1} \mathbf{R}^T \mathbf{s}. \quad (1.27)$$

Entretanto, ao fazer essa solução, é encontrada uma *wavelet* que não só se adequa ao sinal sísmico mas também ao ruído que esteja presente. Assim, deve-se utilizar um método de regularização aliado à solução por mínimos quadrados (Bianco, 2016). Aplicando a regularização de Tikhonov, temos que, a expressão que deve ser minimizada é:

$$|\mathbf{R}\mathbf{w} - \mathbf{s}|^2 + |\Gamma\mathbf{w}|^2$$

onde, Γ representa a matriz de Tikhonov que pode ser calculada por: $\Gamma = \alpha I$, onde α é um parâmetro de regularização e I é a matriz identidade. De forma que a solução regularizada será:

$$\hat{w} = (\mathbf{R}^T \mathbf{R} + \alpha^2 I)^{-1} \mathbf{R}^T \mathbf{s}. \quad (1.28)$$

Apesar da possibilidade de se obter a solução a partir da pseudo-inversa regularizada (Eq. 1.28), é comum utilizar métodos de otimização para estimar os coeficientes da regressão, e.g: métodos baseados na técnica gradiente.

Capítulo 2

Redes neurais artificiais

Redes neurais artificiais (RNA) são algoritmos computacionais que buscam simular o funcionamento dos neurônios e do sistema nervoso central (Graupe, 2007). As redes neurais são modeladas para reproduzir a forma com que o cérebro humano realiza determinadas tarefas (Haykin, 2008). As RNAs se destacam por possuir uma série de características, em especial (da Silva et al., 2016):

- **Adaptabilidade:** Os parâmetros de uma rede neural são ajustados durante o processo de treinamento, examinando o comportamento dos dados e se adaptando;
- **Capacidade de aprendizado:** As RNAs têm capacidade de reconhecer relações entre diferentes variáveis através do seu método de aprendizado; e
- **Capacidade de generalização:** RNAs conseguem aprender padrões existentes no dado, fazendo uma estimativa de soluções com base nos mesmos.

2.1 O neurônio artificial

As redes neurais artificiais são estruturadas por neurônios artificiais. De acordo com da Silva et al. (2016), um neurônio artificial é uma estrutura de processamento que busca copiar o funcionamento de um neurônio biológico. O modelo mais simples de um neurônio artificial foi proposto por McCulloch e Pitts (1943), no qual o neurônio é descrito como uma soma ponderada entre um dado de entrada (\mathbf{X}) e os pesos — coeficientes — dessas entradas (\mathbf{w}). O resultado final é subtraído de um *bias* (θ) e então, enviado para uma função ativação (g), resultando num valor de saída (y). O modelo de neurônio artificial é ilustrado na Figura 2.1. Matematicamente, tem-se que:

$$u = \sum_{i=1}^n w_i \cdot x_i - \theta \quad (2.1)$$

e

$$y = g(u) \tag{2.2}$$

onde n , na Equação 2.1, é o número de atributos - ou variáveis - disponíveis no conjunto de dados \mathbf{X} .

Segundo da Silva et al. (2016), os elementos básicos de um neurônio artificial são:

- Dado de entrada ou *input* (\mathbf{X}): são os sinais do meio externo que representam os valores assumidos pelas variáveis da aplicação da rede. Em geral, os dados de entrada são normalizados para aprimorar a eficiência do aprendizado da rede.
- Pesos ou pesos sinápticos (\mathbf{w}): são valores utilizados para quantificar a relevância de cada dado de entrada para a funcionalidade do neurônio. O peso sináptico junto com o *bias* representam os parâmetros da RNA.
- Combinador linear (\sum): Reúne todos os valores ponderados pelos respectivos pesos para produzir um potencial de ativação.
- *Bias* ou limiar de ativação (θ): é uma variável utilizada para especificar o limite que o resultado produzido pelo combinador linear deve ter para que possa gerar um valor de gatilho para a saída do neurônio.
- Potencial de ativação (u): é o resultado produzido pela diferença entre o combinador linear e o *bias*.
- Função de ativação (g): é uma transformação não linear que tem como objetivo limitar o valor de saída do neurônio dentro de faixa de valores.
- Dado de saída ou *output* (\mathbf{y}): é o valor final produzido pelo neurônio. Também pode ser utilizado como dado de entrada para outros neurônios que estejam interconectados.

2.1.1 Funções de ativação

Ainda tratando da estrutura interna de um neurônio artificial, a função ativação é aquela que irá definir o intervalo de valores que terá a saída de cada neurônio. Segundo da Silva et al. (2016), as funções de ativação podem ser divididas em: parcialmente diferenciáveis e totalmente diferenciáveis.

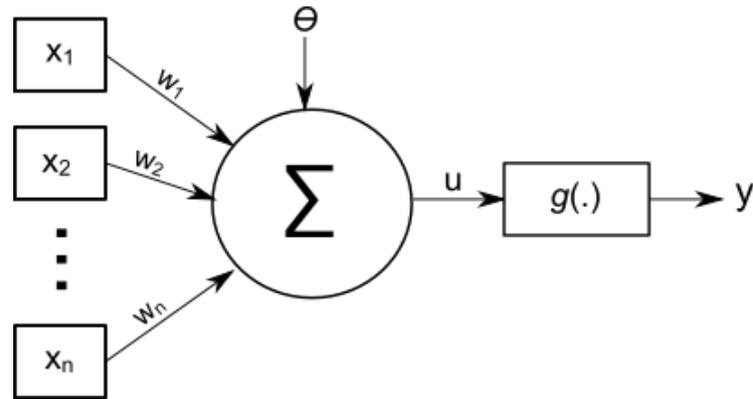


Figura 2.1: Representação de um neurônio artificial.

Funções parcialmente diferenciáveis

As funções parcialmente diferenciáveis são aquelas que possuem pontos cuja derivada de primeira ordem é inexistente, sendo as principais funções deste grupo: função degrau, função degrau bipolar e função rampa simétrica. Essas funções são representadas na Figura 2.2, e são descritas da seguinte forma:

- Função degrau: A função degrau assume um valor unitário positivo quando o potencial de ativação do neurônio for maior ou igual a zero, caso o potencial seja menor que zero, o neurônio terá resultado nulo. Logo,

$$g(u) = \begin{cases} 1, & \text{se } u \geq 0 \\ 0, & \text{se } u < 0. \end{cases} \quad (2.3)$$

- Função degrau bipolar: O resultado dessa função assume valores unitários positivos caso o potencial de ativação for maior que zero; valores nulos quando o potencial é nulo e valores unitários negativos quando o potencial for menor que zero. Matematicamente pode ser escrita como (Eq. 2.4):

$$g(u) = \begin{cases} 1, & \text{se } u > 0 \\ 0, & \text{se } u = 0 \\ -1, & \text{se } u < 0. \end{cases} \quad (2.4)$$

- Função rampa simétrica: Os valores que retornam por essa função são os próprios valores do potencial de ativação quando estão dentro do intervalo $[-a, a]$. Fora desse intervalo, os valores que retornam são os próprios limites. Assim,

$$g(u) = \begin{cases} a, & \text{se } u > a \\ u, & \text{se } -a \leq u \leq a \\ -a, & \text{se } u < -a. \end{cases} \quad (2.5)$$

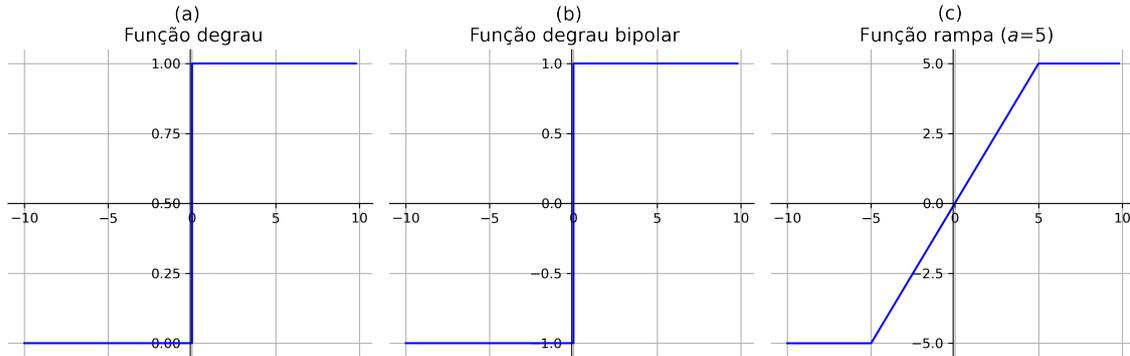


Figura 2.2: Funções de ativação parcialmente diferenciáveis: (a) função degrau, (b) função degrau bipolar e (c) função rampa simétrica.

Funções totalmente diferenciáveis

Por outro lado, funções totalmente diferenciáveis possuem derivadas de primeira ordem para todos os pontos em seu domínio, sendo as principais: a função linear, função sigmóide e a função tangente hiperbólica. A Figura 2.3 ilustra as funções descritas a seguir.

- Função linear: também chamada de função identidade, essa função produz um resultado de saída igual ao potencial de ativação u . Matematicamente,

$$g(u) = u \quad (2.6)$$

- Função sigmóide: Os dados de saída produzidos por essa função sempre assumem valores entre zero e um. A expressão matemática que representa a função é:

$$g(u) = \frac{1}{1 + e^{-\beta u}} \quad (2.7)$$

onde o β é uma constante real associada à declividade da curva.

- Função tangente hiperbólica: Diferentemente da função sigmóide, o resultado da função tangente hiperbólica sempre assume valores entre -1 e 1, seguindo a expressão matemática:

$$g(u) = \frac{1 - e^{-\beta u}}{1 + e^{-\beta u}} \quad (2.8)$$

assim como na função anterior, o β é associado à declividade da função.

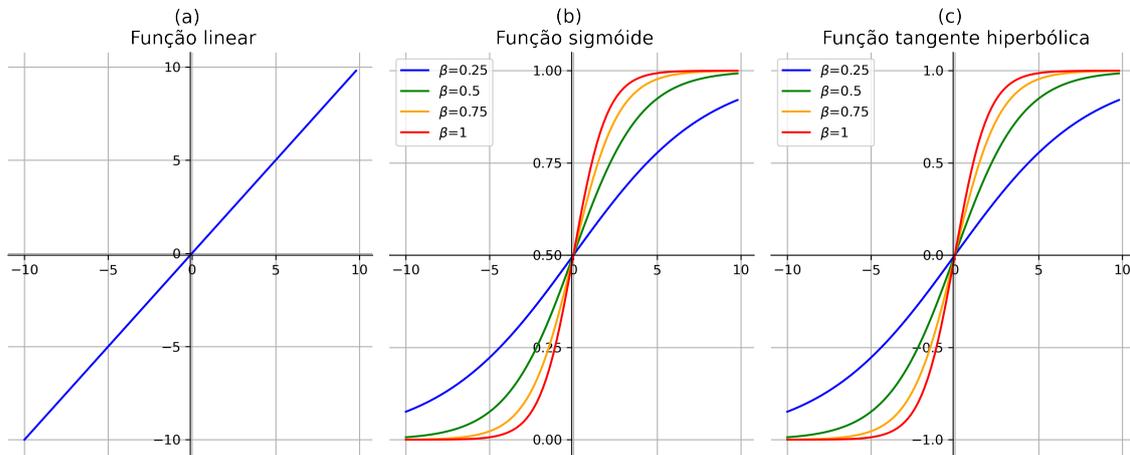


Figura 2.3: Funções ativação totalmente diferenciáveis: (a) função linear, (b) função sigmóide e (c) função tangente hiperbólica. Em (b) e (c) também é mostrado como o parâmetro β influencia na curva.

2.2 Arquitetura de uma rede neural

A arquitetura de uma RNA diz respeito a como seus neurônios estão estruturados. A forma com que os neurônios se configuram está ligada ao algoritmo de aprendizado utilizado para treinar a rede (Haykin, 2008). Em geral, uma RNA pode ser dividida em três partes:

- (i) Camada de entrada: esta é a primeira camada da rede e é responsável por receber as informações do ambiente externo;
- (ii) Camada oculta ou intermediária: essa camada é formada por neurônios cujo valor de entrada é o *output* da camada de entrada. É a responsável por extrair os padrões do dado analisado; e
- (iii) Camada de saída: esta camada é responsável por produzir e apresentar o *output* final da rede, resultado do processamento feito nas camadas anteriores.

Considerando a disposição dos neurônios, número de camadas e como elas se conectam, é possível classificar as arquiteturas de redes. As principais arquiteturas são: rede *feedforward*

de camada simples (Figura 2.4), rede *feedforward* de camadas múltiplas (Figura 2.5) e redes recorrentes (Figura 2.6).

Uma rede com conexão do tipo *feedforward* (ou acíclica) é aquela cuja saída de um neurônio na i -ésima camada não pode ser usado como entrada em camadas de índice menor ou igual a i (Braga, Ludemir e Carvalho, 2000).

2.2.1 Arquitetura *feedforward* de camada simples

A mais simples das arquiteturas de um RNA é a *feedforward* de camada simples. Essa arquitetura é, em geral, empregada em problemas de classificação de padrões e filtragens e é definida por uma camada de entrada diretamente conectada a uma camada de saída com um ou mais neurônios cuja quantidade de valores de saída equivale ao número de neurônios da camada de *output*. As redes Perceptron e Adaline são exemplos de redes neurais com arquitetura *feedforward* simples (da Silva et al., 2016).

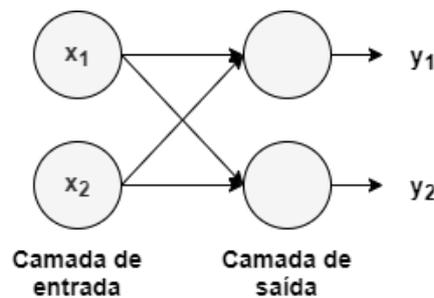


Figura 2.4: Esquemática de uma rede *feedforward* simples.

2.2.2 Arquitetura *feedforward* de camadas múltiplas

Diferentemente da anterior, a arquitetura *feedforward* de camadas múltiplas não só apresenta uma camada de entrada e uma camada de saída, bem como deve possuir uma ou mais camadas ocultas. Essa arquitetura já se enquadra na classificação de rede neural profunda (DNN), que são RNAs que apresentem mais de uma camada oculta. As redes *feedforward* de camadas múltiplas são utilizadas em problemas de classificações de padrões, aproximação de funções e otimização, tendo como exemplos mais comuns as redes neurais Perceptron multicamadas (MLP) e de Base Radial (RBF) (da Silva et al., 2016).

2.2.3 Arquitetura recorrente

A arquitetura recorrente (ou *feedback*) se difere das anteriores por possuir pelo menos um loop de realimentação, ou seja, a saída de um neurônio é utilizada como entrada de uma

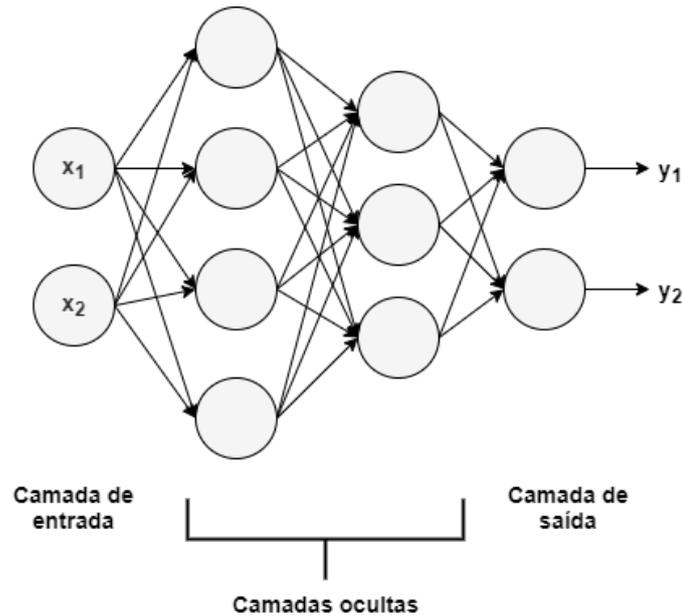


Figura 2.5: Esquemática de uma rede *feedforward* de camadas múltiplas.

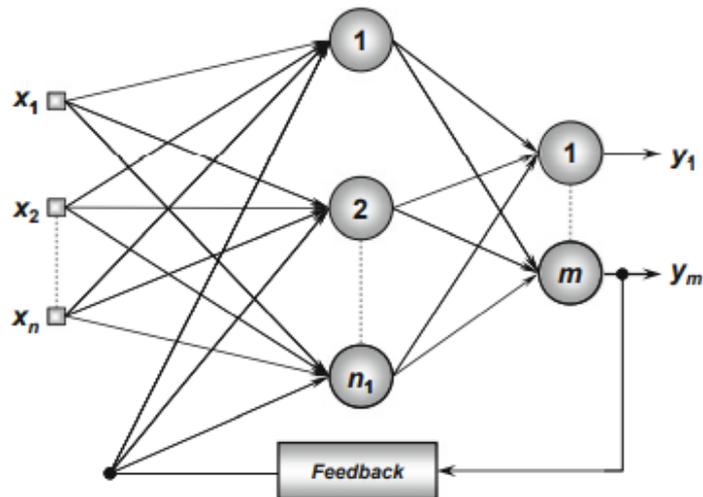


Figura 2.6: Esquemática de uma rede recorrente. Fonte: da Silva et al. (2016).

camada anterior. Para da Silva et al. (2016), o recurso de realimentação dessas redes as torna aptas a processar informações dinâmicas, podendo empregá-las em previsões de séries temporais, sistemas de identificação e otimização, entre outros. Ainda segundo Vellasco (2007), as redes recorrentes podem ser ditas "com memória" por responderem a estímulos dinamicamente. Essa arquitetura pode englobar as redes MLP e Hopfield.

2.3 Redes neurais Perceptron

A rede neural Perceptron é considerado o primeiro modelo de redes neurais artificiais, elaborado por Rosenblatt (1958). O objetivo era implementar um modelo computacional baseado no funcionamento da retina para identificação de padrões geométricos.

O Perceptron é uma rede neural com arquitetura *feedforward* de camada simples que utiliza as Equações 2.1 e 2.2 para cálculo dos *outputs* dos neurônios. A Figura 2.4 pode ser considerada um modelo Perceptron.

Uma limitação para as redes Perceptron foi identificada em 1969 por Minsky e Papert (1969), que perceberam que a rede Perceptron não era capaz de resolver problemas linearmente não-separáveis, isto é, problemas cuja divisão de duas ou mais classes não pode ser representada por uma reta. Concluiu-se que a rede Perceptron devia ser aplicada para casos onde as classes a serem mapeadas fossem linearmente separáveis.

2.3.1 Treinamento da rede Perceptron

O treinamento da rede Perceptron é feito de forma supervisionada, ou seja, é apresentado para a rede a resposta desejada para os dados de entrada, e, a função ativação utilizada na camada de saída da rede é a degrau ou degrau bipolar. Além disso, a rede Perceptron utiliza a regra de aprendizado de Hebb (1949) para ajustar seus parâmetros (pesos e *bias*). Segundo Hebb, a conexão entre dois neurônios se fortalecem quando os mesmos são disparados simultaneamente.

O treinamento consiste em ajustar os valores dos pesos e *bias* caso o *output* varie do valor desejado. O processo é repetido para todos os dados de treinamento até que o resultado seja similar ao *output* esperado. Matematicamente, o ajuste do peso w_i e do *bias* θ é expresso pelas Equações 2.9 e 2.10:

$$w_i^{atual} = w_i^{anterior} + (d^{(k)} - y^{(k)}) \cdot x_i^{(k)} \quad (2.9)$$

$$\theta^{atual} = \theta^{anterior} + (d^{(k)} - y^{(k)}) \cdot (-1) \quad (2.10)$$

onde, w e θ são o peso e *bias*, respectivamente; $x^{(k)}$ é o valor de *input*, $d^{(k)}$ é a resposta desejada e $y^{(k)}$ é o *output* para o k -ésimo neurônio. Geralmente, os valores iniciais do vetor w são pequenos e aleatórios e do vetor θ é -1 ou 1.

Como a rede Perceptron possui apenas um conjunto de *bias*, pode-se adicioná-lo ao vetor dos pesos e reescrever a equação 2.9 na forma vetorial, acrescentando uma taxa de aprendizado α , de forma que a atualização dos coeficientes da rede pode ser dada por (Eq. 2.11):

$$\mathbf{w}^{atual} = \mathbf{w}^{anterior} + \alpha \cdot (d^{(k)} - y) \cdot \mathbf{x}^{(k)} \quad (2.11)$$

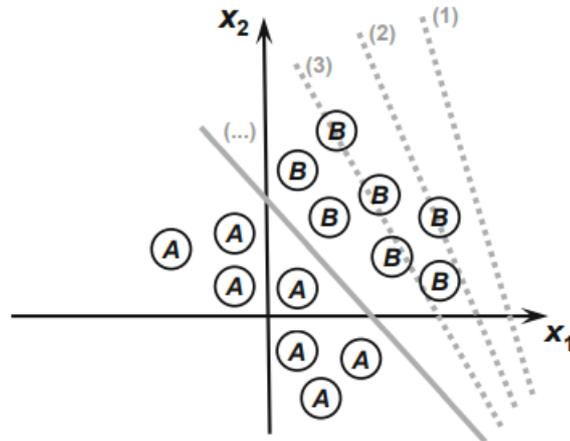


Figura 2.7: Ilustração do processo de treinamento de uma rede Perceptron. Fonte: da Silva et al. (2016).

2.4 Redes neurais Perceptron multicamadas

Como dito na seção anterior, Minsky e Papert concluíram que a rede Perceptron não era capaz de solucionar problemas de classes linearmente não-separáveis. Eles propuseram, então, a inclusão de múltiplas camadas intermediárias, modelo que ficou conhecido como rede neural Perceptron multicamadas (MLP do inglês, *Multi-layer Perceptron*). Em geral, as redes MLP possuem, no mínimo, uma camada oculta que funciona como um detector de características (Braga, Ludemir e Carvalho, 2000). As redes Perceptron multicamadas, diferentemente da Perceptron simples, possuem inúmeras aplicações tanto em problemas de classificação quanto de regressão, sendo empregadas para aproximações de funções, reconhecimento de padrões, previsões de séries temporais e até mesmo, otimização de sistemas (da Silva, Spatti, Flauzino, Liboni e Alves, 2016).

Segundo Haykin (2008), as três principais características de uma rede MLP são:

- Cada neurônio da rede inclui uma função de ativação não-linear e diferenciável;
- A rede contém uma ou mais camadas ocultas; e
- A rede possui uma alta taxa de conectividade.

2.4.1 Treinamento da rede Perceptron multicamadas

Em geral, as redes MLP possuem uma arquitetura *feedforward* com camadas múltiplas e seu treinamento é feito de forma supervisionada dividido em duas etapas (Haykin, 2008):

- Etapa de ida (*forward propagation*): os valores dos pesos são inicializados de forma aleatória dentro de um intervalo de valores pequenos e o dado de entrada é propagado pela rede, camada por camada, até a camada de saída; e
- Etapa de volta (*backward propagation*): um erro é produzido ao comparar o valor do *output* com o desejado. O erro resultante é propagado pela rede, também camada por camada, só que na direção contrária, partindo da última camada até a inicial. Nessa fase, ajustes são feitos nos parâmetros da rede através de técnicas de otimização.

2.4.2 Algoritmo de retropropagação de erros (*backpropagation*)

Como dito anteriormente, o treinamento de uma rede MLP é feito em duas etapas principais, sendo a segunda e mais importante, a retropropagação de erros pela rede neural, também conhecida como *backpropagation*.

O treinamento se inicia pela etapa de ida, onde os pesos sinápticos da rede são iniciados de maneira aleatória dentro de um intervalo de $[-\epsilon, \epsilon]$ e, o *bias*, em geral, é iniciado como -1 ou 1, até que seja calculado um output. Após isso, um erro é obtido através do cálculo da função custo J em função dos parâmetros da rede para cada um dos neurônios da camada de saída. Por simplificação, é assumido que o *bias* (θ) está incluído no vetor dos pesos \mathbf{w} , de forma que a função custo é $J(\mathbf{w})$.

Para minimizar a função custo utilizando um otimizador, é necessário o gradiente da mesma em função dos parâmetros \mathbf{w} , para tal, utiliza-se a regra de derivação da cadeia, de forma que (Deisenroth, Faisal e Ong, 2020):

$$\frac{\partial J}{\partial w_i} = \frac{\partial J}{\partial f_L} \frac{\partial f_L}{\partial f_{L-1}} \cdots \frac{\partial f_{i+2}}{\partial f_{i+1}} \frac{\partial f_{i+1}}{\partial w_i} \quad (2.12)$$

onde, $i = 0, 1, \dots, L-1$, L representando a camada e f representando o resultado da função ativação $g(u)$.

Essa derivação ocorre da camada de saída L , até a camada de entrada 0, logo a primeira derivada calculada é:

$$\frac{\partial J}{\partial w_{L-1}} = \frac{\partial J}{\partial f_L} \frac{\partial f_L}{\partial w_{L-1}} \quad (2.13)$$

Por fim, ao encontrar o gradiente da função custo, é feita a atualização dos parâmetros através da função otimização e uma nova iteração é iniciada com os parâmetros atualizados.

2.4.3 O otimizador Adam

A etapa de otimização de uma rede neural têm como objetivo minimizar sua função custo. A função custo mede o quão próximo o valor estimado está do valor esperado como saída da rede neural. Como dito anteriormente, as variáveis da função custo são os parâmetros da rede (pesos e *bias*), logo, a etapa de otimização busca os valores de peso e *bias* capazes de minimizar a função custo.

A maioria dos otimizadores se baseia no método de otimização do gradiente descendente (GD). Considerando α como uma taxa de aprendizado, o ajuste dos pesos, no GD, seria expresso por:

$$w_{t+1} = w_t - \alpha \frac{\partial J}{\partial w_t} \quad (2.14)$$

onde, α é a taxa de aprendizado e o termo $\frac{\partial J}{\partial w_t}$ representa o gradiente da função custo J , e também, pode ser representado por g_t .

De acordo com Karim (2018), os otimizadores que são variações do método GD se diferenciam em três grupos:

- Os que adaptam o componente gradiente, como os otimizadores Momentum e Nesterov;
- Aqueles que adaptam a taxa de aprendizado, como o AdaGrad e o RMSProp; e
- Aqueles que agem das duas formas. Neste grupo se encontram o otimizador Adam e suas variações, como o AdaMax e o Nadam.

O otimizador Adam (*Adaptive Moment Estimation*) foi elaborado em 2015 por Kingma e Ba (2015) e, segundo os próprios autores, combina as vantagens de dois outros métodos: o AdaGrad e o RMSProp. Atualmente, é um dos métodos de otimização mais populares e utilizados em RNAs.

O funcionamento do Adam utiliza duas médias móveis, uma média móvel exponencial dos gradientes m (ou primeiro momento) e uma média móvel exponencial do quadrado dos gradientes v (ou segundo momento), expressas por:

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t \quad (2.15)$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2 \quad (2.16)$$

onde, β_1 e β_2 são constantes que controlam a taxa de decaimento.

Os termos m e v iniciam como vetores nulos fazendo com que as estimativas iniciais tenham tendência a ser zero, especialmente se as constantes β_1 e β_2 forem valores próximos de 1, sendo necessária uma correção de *bias*, de forma que:

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t} \quad (2.17)$$

$$\hat{v}_t = \frac{v_t}{1 - \beta_2^t} \quad (2.18)$$

Assim, a atualização dos parâmetros pelo Adam é feita de acordo com a seguinte equação:

$$w_{t+1} = w_t - \frac{\alpha}{\sqrt{\hat{v}_t} + \epsilon} \cdot \hat{m}_t \quad (2.19)$$

Capítulo 3

Metodologia

A metodologia adotada para a implementação de uma rede neural que recebe como dado de entrada um traço sísmico e retorna uma *wavelet* sísmica de fase zero é apresentada no fluxograma mostrado na Figura 3.1. Cada passo mencionado no fluxograma será explicado nas próximas subseções. Todas as implementações foram executadas em um notebook com 16 GB de memória RAM, processador Intel i7-7700HQ e sistema operacional *64 bits Windows 10 Home*, utilizando a linguagem de programação *Python 3.9* (Van Rossum e Drake, 1995) por meio do Jupyter Notebook e, em particular, a rede neural foi desenvolvida através do pacote *Keras/TensorFlow* (Chollet et al., 2015).

3.1 Criação dos modelos sintéticos

Primeiramente, foram gerados modelos sintéticos de camadas plano-paralelas de densidade (ρ) e velocidade (v) com o objetivo de convolver os coeficientes de reflexão desses modelos com *wavelets* idealizadas de forma a gerar os traços sísmicos sintéticos utilizados como dados de entrada da rede neural. No total, foram gerados 70.000 traços sísmicos sintéticos através do modelo convolucional explicado na Seção 1.1.

Inicialmente, foram definidas as profundidades Z de início (0 m) e término dos modelos (3000 m) e a taxa de amostragem da sísmica que foi definida como 4 ms. A partir daí, o algoritmo define aleatoriamente um número de camadas $N_{camadas}$ que pode variar entre 80 e 250, com diferentes espessuras, as espessuras são valores sorteados aleatoriamente pelo algoritmo dentro do intervalo de [0,3000] organizados de forma crescente.

Também, de forma aleatória, é definido o intervalo numérico da velocidade mínima (v_{min}) e máxima (v_{max}), de forma que as velocidades escolhidas aleatoriamente sempre fiquem dentro do intervalo pré-estabelecido, que varia de $v_{min} = [1600, 2300]$ a $v_{max} = [4500, 6000]$.

As velocidades das camadas aumentam gradativamente com a profundidade, de forma

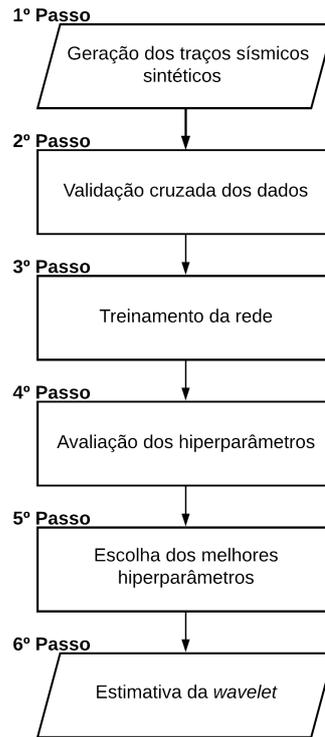


Figura 3.1: Fluxograma da metodologia implementada no trabalho.

que a velocidade de uma camada i é definida de forma aleatória em um intervalo de:

$$[v_{i-1} - (\mu \cdot \Delta v), v_{i-1} + (\mu \cdot \Delta v)]$$

onde μ é uma constante que aumenta a janela de escolha das velocidades e Δv é dada por:

$$\Delta v = \frac{(v_{max} - v_{min})}{N_{camadas}} \quad (3.1)$$

O conjunto de densidades do modelo foi definido entre os valores de 1,9 e 3 kg/m³, seguindo um critério de escolha da propriedade de forma semelhante à proposta para a velocidade, com um incremento de 0.05 entre cada valor de densidade. Para cada camada do modelo, é atribuído um dos valores de densidade desse conjunto de forma aleatória pelo algoritmo. Uma vez construído os modelos de camadas plano-paralelas, é calculada a impedância acústica de cada camada a partir da seguinte expressão (Eq. 3.2):

$$AI = v \cdot \rho \quad (3.2)$$

Note que as camadas estão definidas em termos de sua espessura, ou seja, em profundidade. Para obter o traço sísmico sintético é necessário realizar a conversão de tempo-profundidade, para que seja possível representar o dado em tempo duplo (TWT). De forma

<i>Wavelet</i>	Nº de traços
Ricker	13.907
Gabor	13.940
Ormsby	14.038
Klauder	13.994
Sinc	14.121
Total	70.000

Tabela 3.1: A tabela indica o número das *wavelets* sísmicas utilizadas para a criação dos 70 mil traços sintéticos.

análoga ao procedimento adotado na amarração sísmica-poço, pode-se calcular a vagarosidade Δt , a partir da velocidade sísmica das camadas, e logo após é feita a conversão através das seguintes equações (3.3 e 3.4):

$$\Delta t = \frac{1}{v} \quad (3.3)$$

e

$$t = 2 \cdot (Z_i - Z_{i-1}) \cdot \sum \Delta t_i. \quad (3.4)$$

onde, Z_i é a profundidade da i -ésima camada.

Em seguida, é feita uma interpolação para reamostrar o dado em tempo duplo para a mesma taxa de amostragem de um traço sísmico típico de exploração de petróleo - que neste trabalho foi de 4 ms — e em seguida é calculado o coeficiente de reflexão através da Eq. 1.2, para se obter o modelo de refletividade.

O próximo passo para geração do traço sísmico sintético é a convolução do modelo de refletividade com uma *wavelet* sísmica. As *wavelets* utilizadas foram a Ricker, Gabor, Ormsby, Klauder e Sinc, apresentadas na Seção 1.3, todas com o tempo de duração de 0.388 s, logo, todas as *wavelets* possuem 97 amostras.

Para garantir que houvesse uma variedade na banda de frequência e dos tipos de *wavelet*, foram selecionadas, para cada assinatura sísmica, frequências, de forma aleatória, de tal forma que as *wavelets* Ricker, Gabor e Sinc tivessem suas frequências dominantes no intervalo entre 5 e 60 Hz. Enquanto para as *wavelets* Klauder e Ormsby, o intervalo de frequência é entre 5 e 125 Hz. Como a Klauder e a Ormsby necessitam de mais de uma frequência, foi definido uma restrição, onde a diferença entre as frequências posterior e anterior não poderia ser inferior a 20 Hz. A Tabela 3.1 mostra a quantidade das *wavelets* sísmicas que foram convolvidas com um determinado modelo de refletividade sísmica.

Por fim, é feita a convolução da série de coeficientes de reflexão com o pulso sísmico gerado e se obtém o traço sísmico sintético. Foram produzidos dois conjuntos de dados: um sem adição de ruído, para verificar se uma rede neural Perceptron multicamadas é capaz de extrair uma *wavelet* em um traço sísmico sintético, e um outro conjunto de dados com adição de ruído, afim de se analisar a estabilidade da rede neural.

O ruído adicionado foi do tipo gaussiano e o desvio padrão foi definido a partir de diferentes porcentagens da amplitude máxima do traço sísmico sintético, para assim atribuir o desvio padrão do ruído normal gaussiano a ser distribuído no conjunto de dados, conforme a Tabela 3.2. Em outras palavras, um conjunto de valores aleatórios extraídos a partir de função gaussiana é somado aos traços sísmicos sintéticos, e os parâmetros da gaussiana são definidos a partir do valor máximo absoluto das amostras do traço sísmico avaliado. A Figura 3.2 ilustra um modelo gerado pela metodologia.

Porcentagem da amplitude máxima	Nº de traços	% do total de traços
0	10.000	14,5
10%	20.000	28,5
15%	20.000	28,5
20%	20.000	28,5
Total	70.000	100

Tabela 3.2: Tabela que apresenta o percentual adotado para adição de ruído nos traços sísmicos sintéticos, os números absolutos dos traços em que os ruídos foram adicionados e sua porcentagem.

3.2 Validação cruzada

Os testes de validação dos dados são aplicados para avaliar se as predições realizadas por cada algoritmo desempenham de forma satisfatória o papel de prever novas amostras, além de testar a estabilidade do algoritmo de aprendizado. De forma geral, isto é feito separando uma porção das amostras do conjunto de dados para compor um subconjunto de treino e outro de teste e assim avaliar o desempenho do treinamento realizado por cada algoritmo.

O método de validação cruzada utilizado foi o *Holdout*. A validação cruzada *Holdout* é a mais simples e consiste em dividir o conjunto de dados aleatoriamente em dois subconjuntos, um de treino e um de teste, de forma que sigam uma proporção p para o treinamento e $(1 - p)$ para o teste, sendo, em geral, $p > 0,5$. Nos experimentos executados foi utilizado o valor de $p = 0,8$, logo, 80% do conjunto de dados foi utilizado para treinamento da rede neural e 20% para teste.

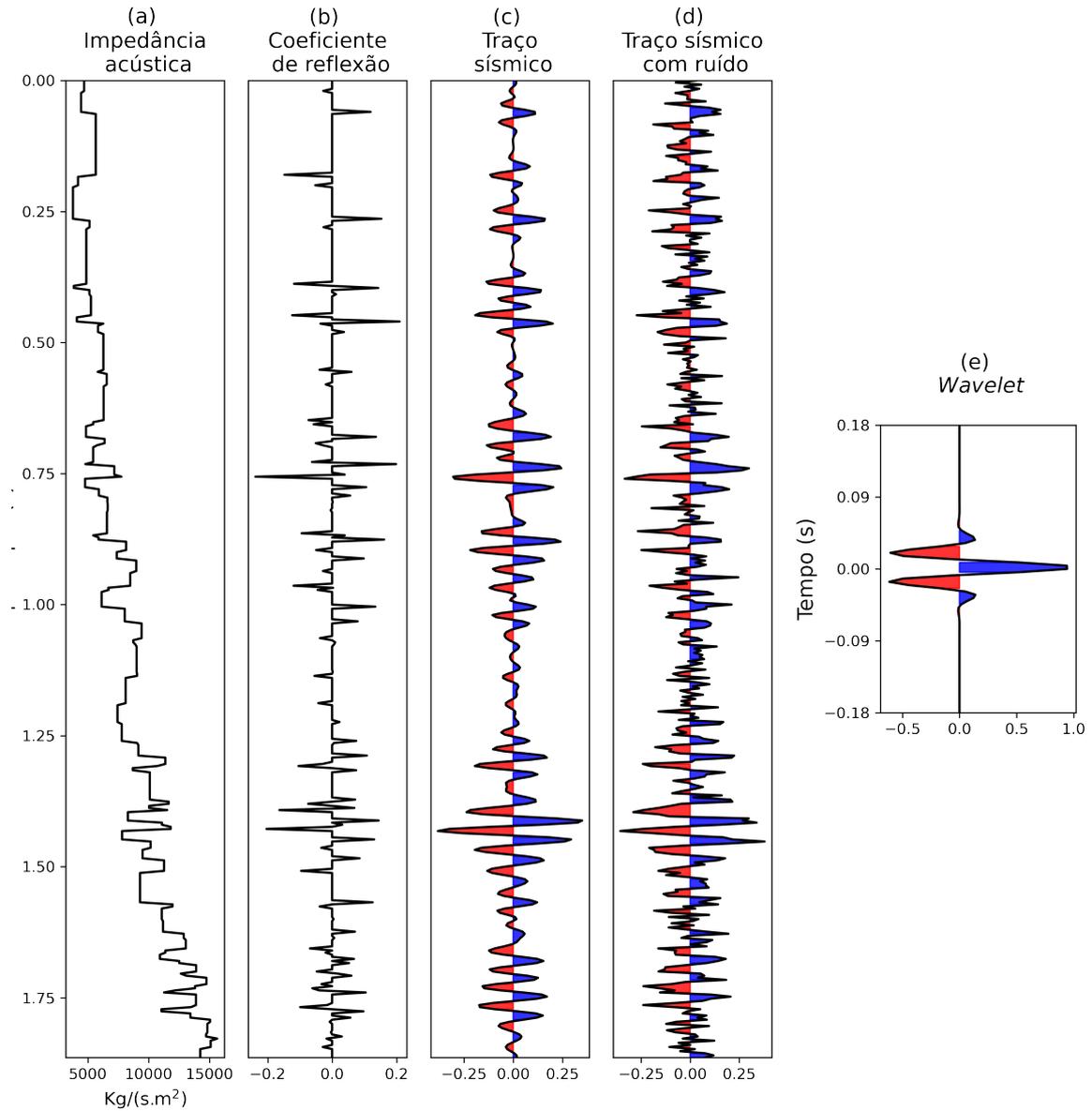


Figura 3.2: Exemplo de modelo e traços sísmicos sintéticos gerado pela metodologia: (a) é o modelo de impedância acústicas, (b) o modelo da função refletividade, (c) o traço sísmico sintético, (d) o traço sísmico sintético com a adição de ruído e (e) a wavelet utilizada na geração dos traços sintéticos.

3.3 Treinamento da rede neural

3.3.1 Configuração

A rede neural construída possui cinco camadas: um vetor de entrada, três camadas ocultas e uma de saída. A camada de entrada possui 300 amostras do traço sísmico como dado de entrada, já para as camadas ocultas, a primeira e segunda tem 300 neurônios e a terceira tem 200. Por fim, a camada de saída possui 97 neurônios, mesmo valor de amostras das *wavelets* utilizadas no treinamento. No total, a rede construída possui 260.297 parâmetros.

A função custo utilizada foi a logaritmo do cosseno hiperbólico (*logcosh*) que tem funcionamento semelhante ao erro médio quadrático (MSE), mas não é tão afetada por ocasionais predições incorretas. O logaritmo do cosseno hiperbólico pode ser definido como (Eq. 3.5):

$$\text{logcosh} = \log \left(\frac{e^x + e^{-x}}{2} \right). \quad (3.5)$$

Todas as camadas da rede possuem a mesma função ativação, que é a tangente hiperbólica (Seção 2.1.1) e o método de otimização utilizado foi o Adam (Seção 2.4.3). A Figura 3.3 ilustra a arquitetura da rede construída.

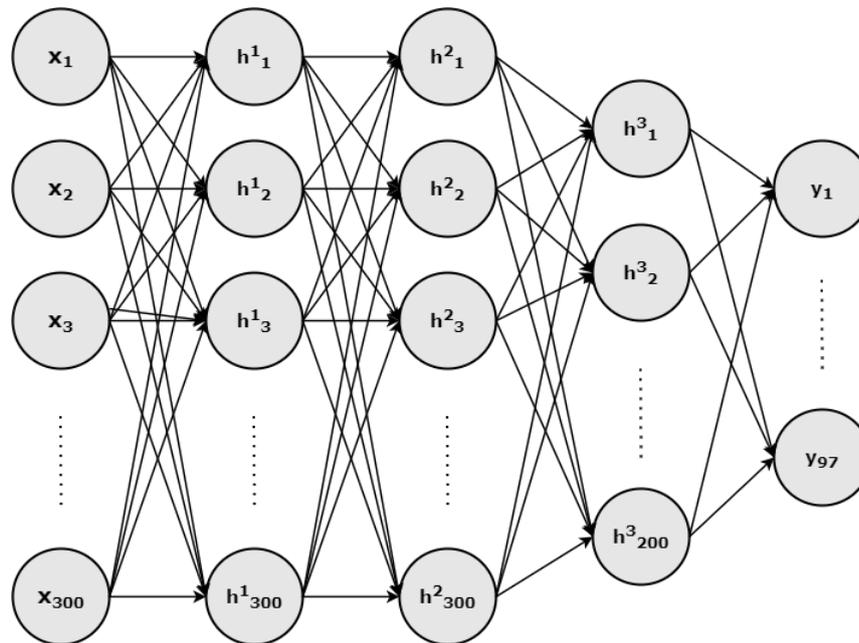


Figura 3.3: Ilustração da rede neural utilizada para predição das *wavelets* sísmicas. A camada de *input* é composta por 300 amostras do traço sísmicos, a primeira, segunda e terceira camada oculta têm 300, 300 e 200 neurônios respectivamente, e o *output* é a *wavelet* sísmica com 97 amostras.

3.3.2 Avaliação dos hiperparâmetros

Para iniciar o treinamento da rede neural, é necessário definir os seus hiperparâmetros. Os hiperparâmetros são parâmetros definidos pelo usuário que controlam a forma de aprendizado da rede e, conseqüentemente, seu resultado. Os hiperparâmetros considerados neste trabalho são:

- **Taxa de aprendizado (α):** é o valor que controla a velocidade de otimização do dado. Um valor de α muito grande pode fazer com que a rede neural não consiga convergir para o mínimo da função custo, por outro lado, um valor muito pequeno pode demorar de convergir e ser computacionalmente custoso;
- **Número de épocas:** o número de épocas define a quantidade de vezes que o conjunto de dados servirá de *input* para a rede. Valores pequenos podem fazer com que a rede neural não seja capaz de solucionar o problema, enquanto valores altos podem causar *overfitting*, ou seja, a rede funciona bem para o conjunto de treino mas não para o conjunto de teste; e
- **Tamanho de *batch*:** é o número de exemplos utilizados em cada iteração. O tamanho de *batch* pode se enquadrar em três modos:
 - Modo *batch*: onde o tamanho de *batch* é igual ao tamanho total do conjunto de dados. Esse modo garante uma convergência do gradiente com grande acurácia, entretanto, pode ser demorado e custoso computacionalmente.
 - Modo *mini-batch*: é o modo mais comum onde o tamanho de *batch* é maior que 1 e menor que o tamanho do input. Geralmente, são escolhidos valores relacionados a 2^i e, valores altos permitem a paralelização dos cálculos na GPU. As redes neurais costumam treinar mais rapidamente quando se usam *mini-batches*.
 - Modo estocástico: onde o tamanho de *batch* é igual a 1, ou seja, os parâmetros são atualizados para cada amostra. Assim como o modo *batch*, o modo estocástico pode ser demorado, além disso, possui uma variação muito grande nos valores de gradiente até a convergência.

A métrica da rede neural é o método de avaliação dos resultados da rede, servindo também, para avaliar os hiperparâmetros utilizados. Neste trabalho, a métrica utilizada para análise dos resultados foi a Similaridade dos Cossenos, expressa por (Eq. 3.6):

$$SimCos = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \|\mathbf{B}\|} = \frac{\sum_{i=1}^n A_i B_i}{\sqrt{\sum_{i=1}^n A_i^2} \sqrt{\sum_{i=1}^n B_i^2}} \quad (3.6)$$

Foram realizadas duas etapas de treinamentos da RNA: uma tendo como *input* os traços sísmicos sintéticos sem adição ruído — para avaliar se a rede neural é capaz de prever *wavelets* sísmicas de forma satisfatória —, e uma segunda etapa tendo uma adição de diferentes níveis de ruído nos traços sísmicos do conjunto de treinamento.

Para primeira etapa, foram realizados 3 conjuntos de treinamentos para avaliação dos resultados das variações de valores dos hiperparâmetros. Para o primeiro conjunto foi testado a taxa de aprendizado e fixados os valores do tamanho de batch e do número de épocas em 256 e 1000, respectivamente. Esses valores iniciais foram escolhidos pois permitiam um treinamento rápido para o tamanho do conjunto de dados utilizados. Foram testados 9 diferentes valores de taxas de aprendizado e avaliados seus desempenhos por meio do valor da função custo e da métrica. No segundo conjunto de treinamento, o hiperparâmetro testado foi o tamanho de batch com 7 valores testados. Para tal, foi mantido o valor fixo do número de épocas anterior (1000 épocas) e fixada a taxa de aprendizado com o valor de melhor desempenho no conjunto anterior. Por fim, no terceiro conjunto de treinamentos foram analisados 5 valores de números de épocas cuja taxa de aprendizado e tamanho de *batch* utilizadas foram as que obtiveram melhor desempenho nos experimentos anteriores. Nesse primeiro conjunto de experimentos, a RNA foi treinada 21 vezes.

Na segunda etapa, de forma a acelerar o treinamento, foram introduzidos os parâmetros obtidos no treinamento da rede neural com melhor desempenho geral e, apenas avaliados os resultados da variação do número de épocas. Logo, foram analisados 5 diferentes valores de números de épocas com o objetivo de avaliar a estabilidade do resultado da rede neural ao serem introduzidos dados ruidosos. Todos os resultados obtidos nos treinamentos são apresentados no Apêndice A.

3.3.3 Seleção dos melhores hiperparâmetros

Os melhores hiperparâmetros são aqueles que apresentam baixos valores da função custo, altos valores na métrica e que, a partir deles, não haja grandes variações nos resultados entre treinamento e teste. A Figura 3.4 apresenta os resultados dos treinamentos feitos na RNA sem o ruído. Os hiperparâmetros escolhidos como de melhor performance para a primeira rede neural — antes de atualizá-la com a adição de ruídos em algumas amostras do conjunto de dados — foram: 0,001 para a taxa de aprendizado (α), um tamanho de *batch* igual a 64 e 5000 para o número de épocas. O valor da função custo e a similaridade dos cossenos para a RNA com estes hiperparâmetros foram de $4,95 \times 10^{-4}$ e 0,988 para o treino e $3,5 \times 10^{-3}$ e 0,905 para o teste, respectivamente.

Para realizar o treinamento da segunda RNA — aquela com a adição de ruídos no conjunto de dados — foram importados os parâmetros da rede anterior, ou seja, os pesos e

bias da rede treinada com traços sísmicos sintéticos sem ruídos, e foi possível notar (Figura 3.5) que a rede neural se manteve estável com o aumento do número de épocas, ou seja, o aumento de iterações não influenciou de forma significativa o valor da função custo e a métrica de avaliação. A rede neural Perceptron multicamadas teve seus pesos atualizados para que fosse capaz de lidar com eventuais ruídos que pudessem ocorrer no sinal sísmico, e o número de atualizações adotado foi de 5000 épocas. A função custo e a similaridade dos cossenos foram, respectivamente, $7,8 \times 10^{-4}$ e 0,981 para o treino e $4,9 \times 10^{-3}$ e 0,867 para o teste.

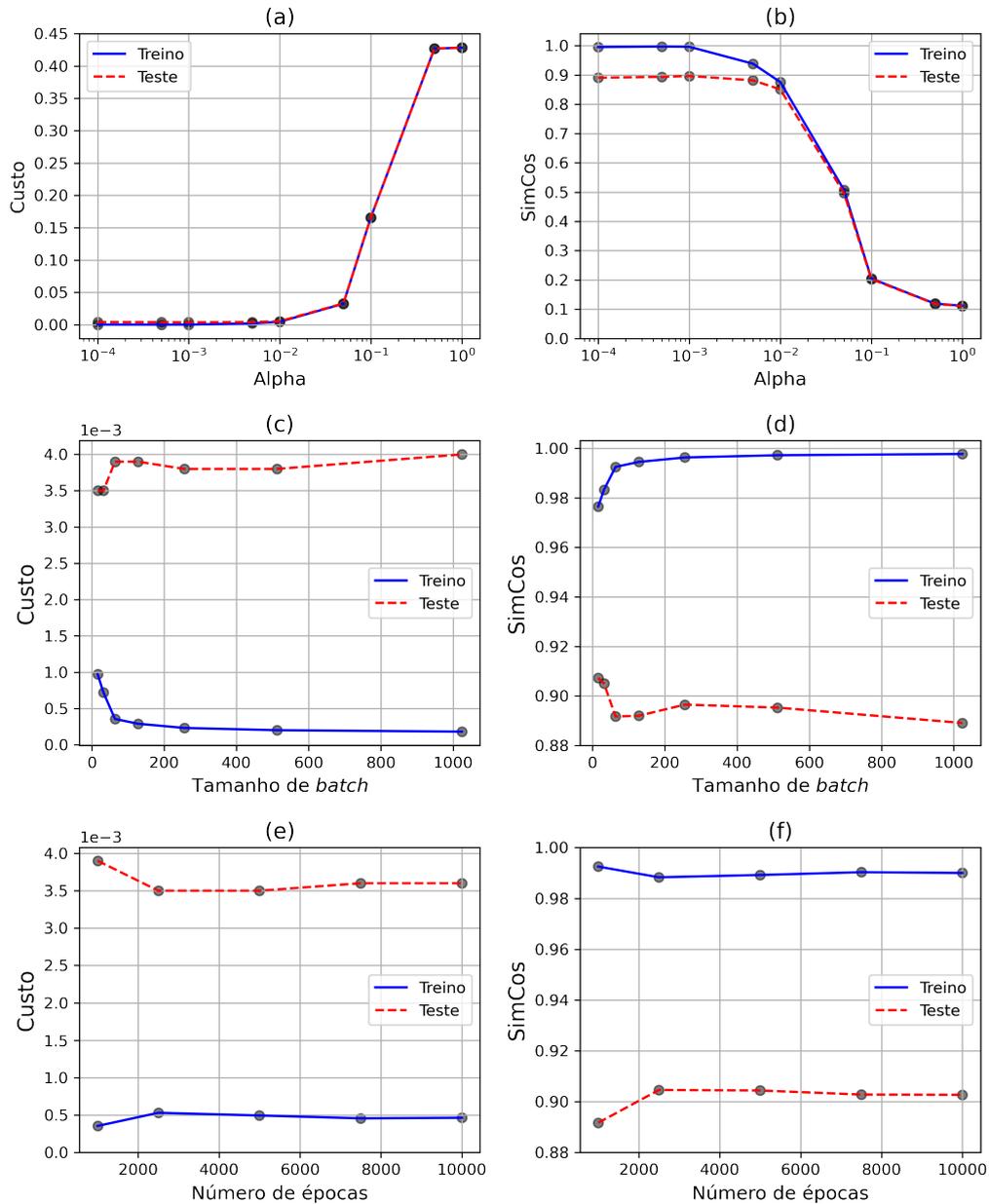


Figura 3.4: Análise dos hiperparâmetros utilizados para o treinamento da rede neural utilizada para a predição de *wavelets* utilizando traços sísmicos sintéticos sem ruídos: (a), (c) e (e) são os gráficos dos valores de função custo (*logcosh*) e (b), (d) e (f) são os gráficos de resultados da métrica utilizada para a análise do treinamento para os valores de α , tamanho de *batch* e número de épocas, respectivamente.

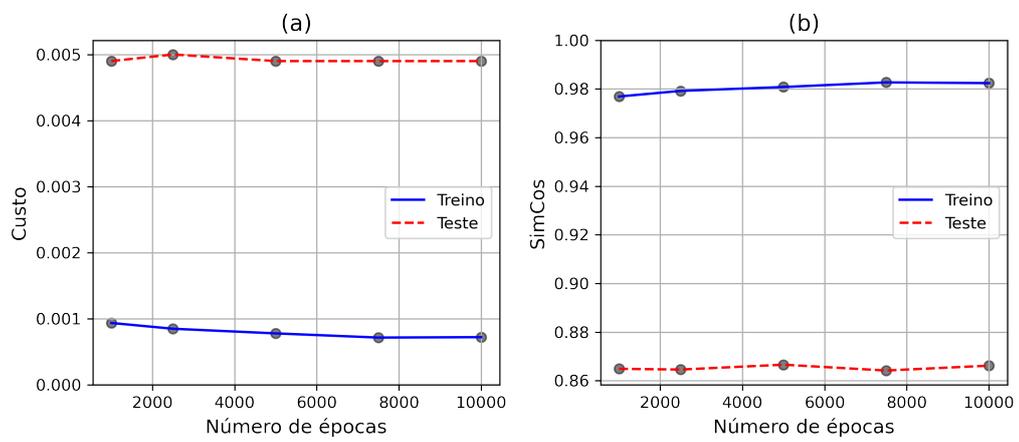


Figura 3.5: Resultados dos treinamentos feitos na segunda etapa da RNA — com adição de ruídos aos traços sísmicos: (a) é a variação dos valores da função custo e (b) os valores da métrica de cada treinamento para o número de épocas.

Capítulo 4

Resultados e Discussão

Este capítulo traz uma discussão sobre o treinamento da rede neural Perceptron multicamadas e os resultados obtidos para a predição de traços sísmicos sintéticos sem ruídos (Seção 4.1) e com ruídos (Seção 4.2). Além disso, na Seção 4.3, é feita uma breve comparação dos resultados obtidos pela RNA e pelo método da autocorrelação utilizando os traços sísmicos sintéticos ruidosos.

4.1 Análise dos resultados obtidos no primeiro treinamento

Para analisar os resultados obtidos pela rede neural Perceptron multicamadas treinada com traços sísmicos sintéticos sem adição de ruído normal, são apresentadas as predições de cinco *wavelets* idealizadas — uma de cada tipo — presentes no conjunto de treinamento, como mostram as Figuras 4.1, 4.2, 4.3, 4.4, 4.5, e são apresentadas as *wavelets* observadas e calculadas — indicadas nas subfiguras (a) e (c) —, a diferença entre elas (subfigura (e)) e os seus respectivos espectros de frequência (subfiguras (b), (d) e (f)). A Figura 4.6 mostra os traços sísmicos do conjunto de teste utilizados para as análises dos resultados obtidos pela RNA.

Començando a análise da predição a partir de uma *wavelet* Ricker, cujo o conteúdo de frequência dominante é de 33 Hz. Visualmente, a *wavelet* predita (Figura 4.1 (c)) manteve as características principais do lobo central, e deformou os lobos laterais, adicionando informações que não eram previstas de acordo com o pulso de fase zero observado na Figura 4.1 (a). O espectro de frequência da *wavelet* predita mostra que foi adicionado um conteúdo de baixa frequência, nos intervalos de 15 a 30 Hz, e de mais alta frequência no intervalo de 40 a 60 Hz, aproximadamente, o que faz com que o espectro se assemelhe ao espectro de frequências de uma *wavelet* Ormsby. O coeficiente de correlação entre os pulsos sísmicos

observado e calculado foi de aproximadamente 0,993.

A predição da *wavelet* Gabor (Figura 4.2 (c)) mostra um resultado muito semelhante a resposta esperada (Figura 4.2 (a)), porém levemente assimétrica, com uma correlação de 0,998. Houve um pouco de perda na banda de frequência da *wavelet* sísmica próximo à frequência dominante de 21 Hz. O espectro de frequências calculado (Figura 4.2 (d)) assemelhasse muito com o espectro do pulso sísmico observado (Figura 4.2 (b)).

O pulso sísmico predito para a *wavelet* Ormsby (Figura 4.3 (c)), cujo conjunto de frequências f_1 , f_2 , f_3 e f_4 é dado por [9,12,79,114] Hz, tem características assimétricas, porém mantém bastante semelhança com o sinal observado (Figura 4.3 (a)). Era esperado um espectro de frequência trapezoidal, como mostra a Figura 4.3 (b), porém houve uma perda de amplitude nas frequências centrais, como visto no espectro calculado na Figura 4.3 (d). O espectro calculado se assemelha ao espectro de frequências de uma *wavelet* Klauder com picos de amplitude nas frequências laterais. No geral, a *wavelet* predita é satisfatória e possui um coeficiente de correlação de 0,993.

Para o sinal do tipo Klauder com frequências de $f_1 = 8$ e $f_2 = 64$ Hz, vê-se que a *wavelet* predita consegue estimar de forma adequada o lóbulo central mas não as variações laterais da *wavelet* observada, como mostram as Figuras 4.4 (c) e 4.4 (a). Ainda assim, é possível notar que há pouco resíduo entre as *wavelets* e a correlação entre as duas é de 0,989. Já para o espectro de frequências, o espectro calculado, Figura 4.4 (b), adiciona conteúdo de frequência no primeiro pico de amplitude entre 20 e 45 Hz mas não no segundo, entre 45 e 65 Hz, se diferenciando do espectro observado (Figura 4.4 (d)).

Por fim, foi escolhida uma *wavelet* Sinc predita, como aponta a Figura 4.4 (c), cujo pulso sísmico calculado é muito similar ao observado na Figura 4.4 (a), de frequência dominante igual a 56 Hz. No espectro de frequências, nota-se que o espectro observado (Figura 4.4 (b)) possui uma tendência plana até, aproximadamente, 50 Hz, apresentando uma diminuição do conteúdo de frequências altas a partir desse ponto. Essa tendência é mantida pelo espectro predito (Figura 4.4 (d)) onde há regiões com pequenas adições de frequência, de 0 a 20 Hz e de 30 a 45 Hz, e pequenas perdas nos intervalos de 20 a 30 Hz e de 45 a 60 Hz. O coeficiente de correlação entre as *wavelets* observada e predita é de 0,997.

4.2 Análise dos resultados obtidos no segundo treinamento

Da mesma forma que na seção anterior, são apresentadas as predições para cada um dos 5 tipos de *wavelets* idealizadas presentes no conjunto de dados para análise dos resultados do segundo treinamento da RNA com adição do ruído normal. Os traços sísmicos utilizados

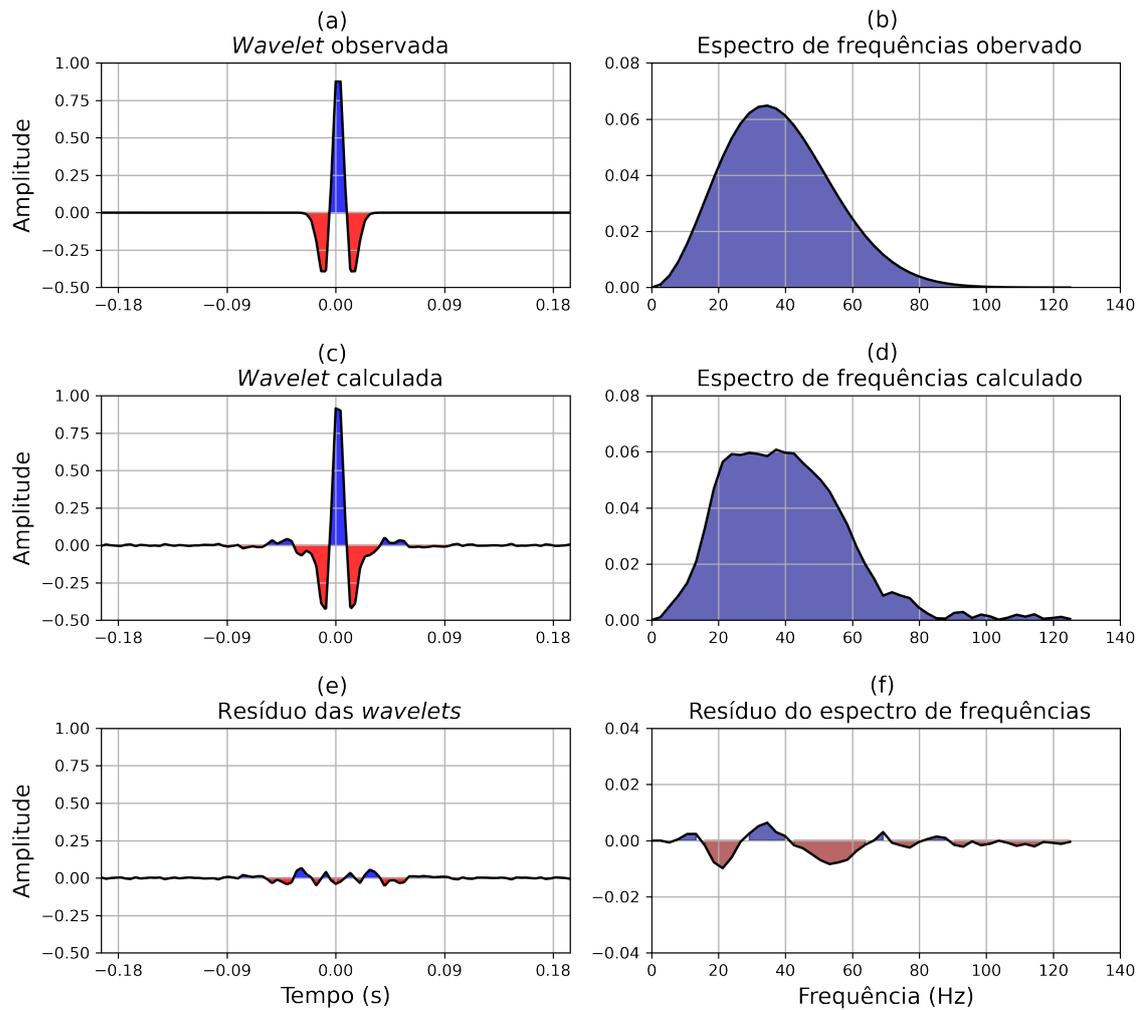


Figura 4.1: Resultado da predição de uma *wavelet* Ricker pela primeira RNA: (a), (c) e (e) são as *wavelets* observada, calculada e o resíduo; enquanto (b), (d) e (f) são os respectivos espectros de frequência. O coeficiente de correlação entre as *wavelets* observada e calcula é de 0,993.

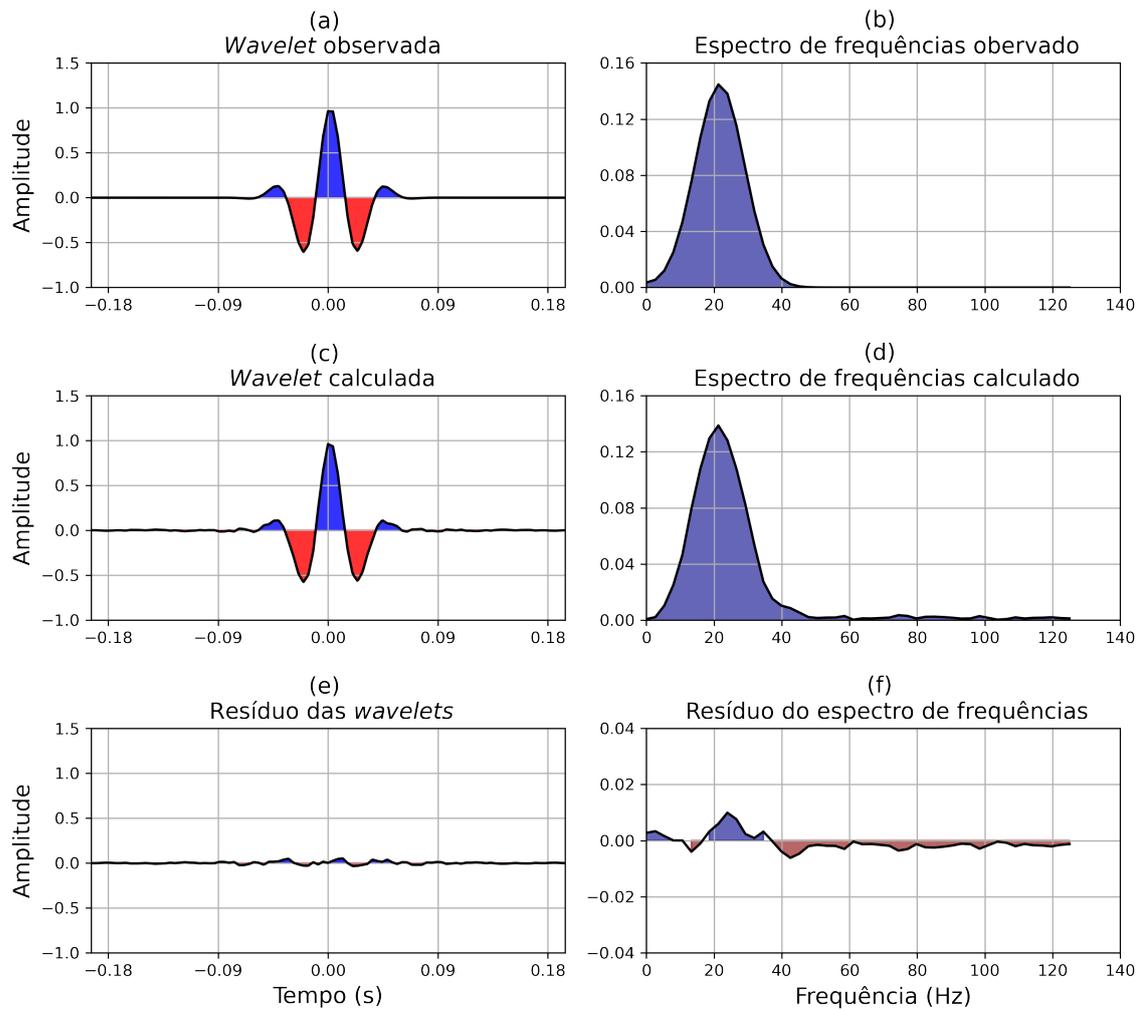


Figura 4.2: Resultado da predição de uma *wavelet* Gabor pela primeira RNA: (a), (c) e (e) são as *wavelets* observada, calculada e o resíduo; enquanto (b), (d) e (f) são os respectivos espectros de frequência. O coeficiente de correlação entre as *wavelets* observada e calcula é de 0,998.

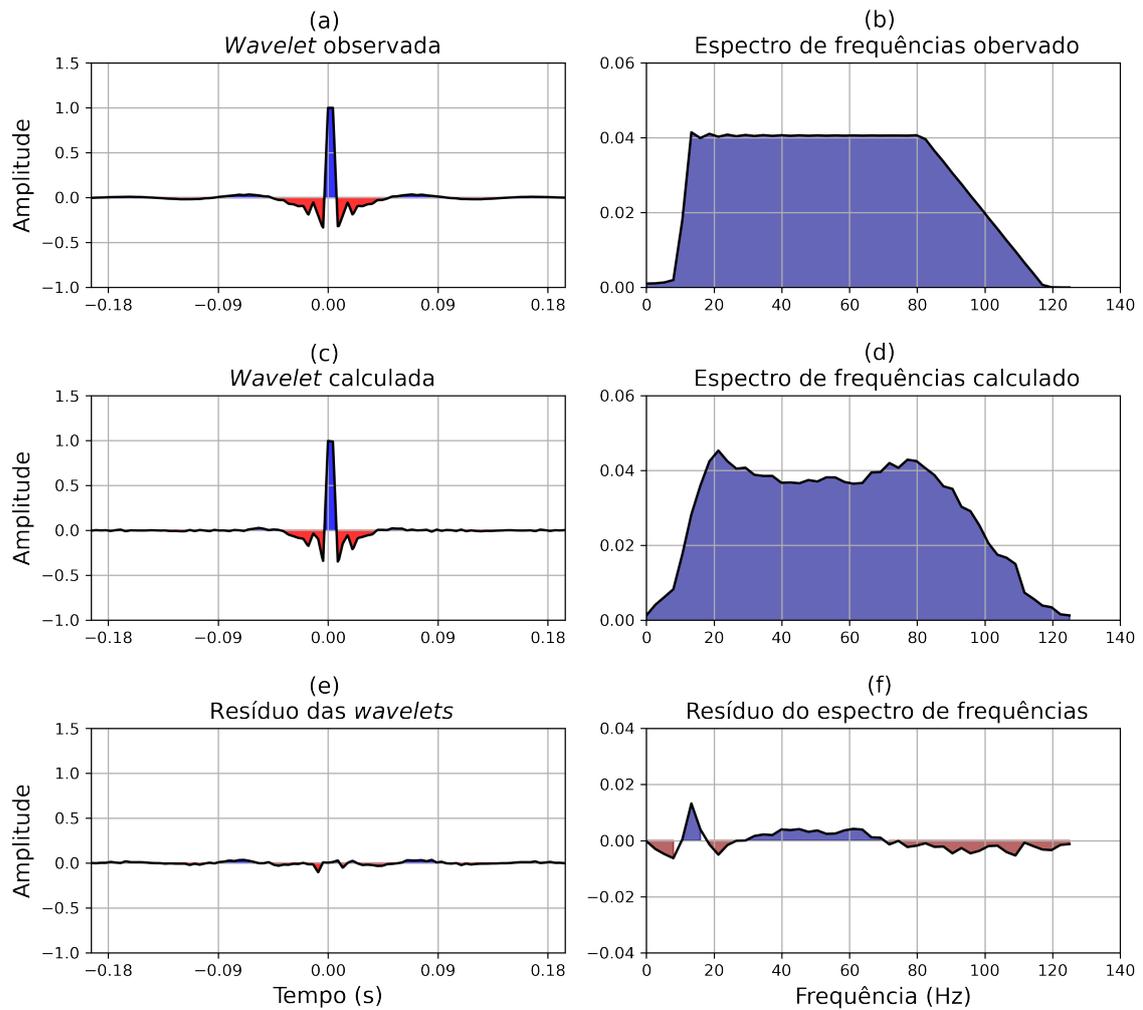


Figura 4.3: Resultado da predição de uma *wavelet* Ormsby pela primeira RNA: (a), (c) e (e) são as *wavelets* observada, calculada e o resíduo; enquanto (b), (d) e (f) são os respectivos espectros de frequência. O coeficiente de correlação entre as *wavelets* observada e calcula é de 0,993.

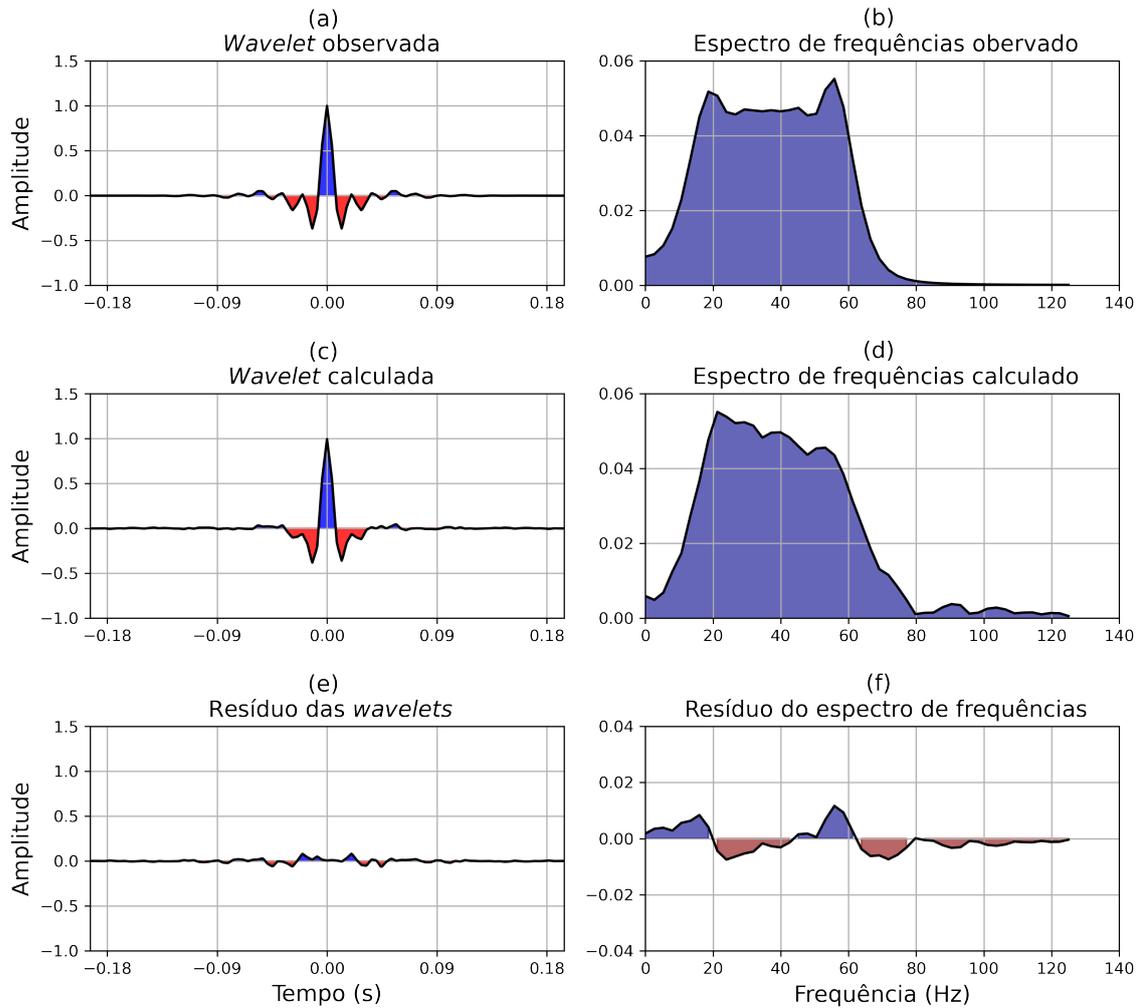


Figura 4.4: Resultado da predição de uma *wavelet* Klauder pela primeira RNA: (a), (c) e (e) são as *wavelets* observada, calculada e o resíduo; enquanto (b), (d) e (f) são os respectivos espectros de frequência. O coeficiente de correlação entre as *wavelets* observada e calcula é de 0,989.

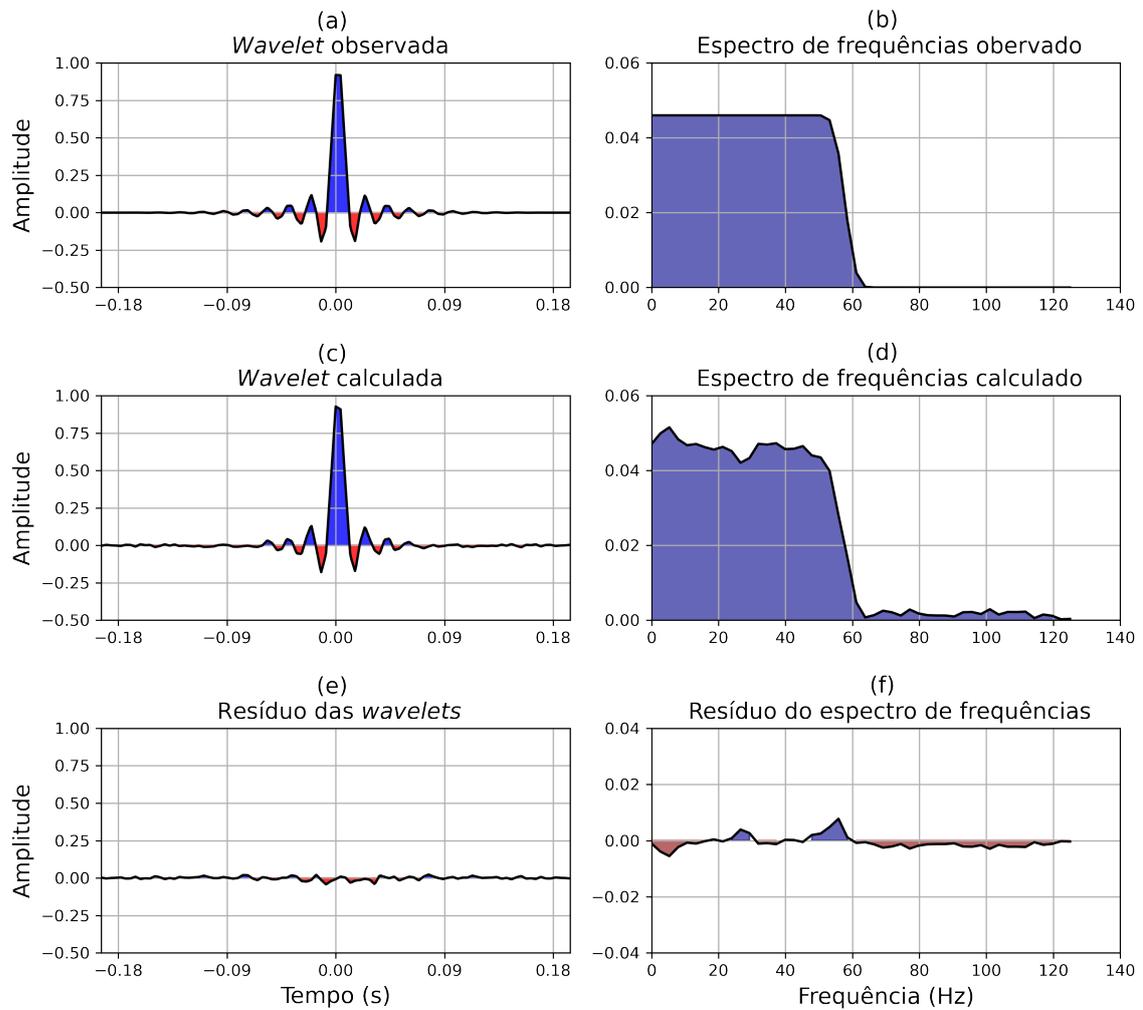


Figura 4.5: Resultado da predição de uma *wavelet* Sinc pela primeira RNA: (a), (c) e (e) são as *wavelets* observada, calculada e o resíduo; enquanto (b), (d) e (f) são os respectivos espectros de frequência. O coeficiente de correlação entre as *wavelets* observada e calcula é de 0,997.

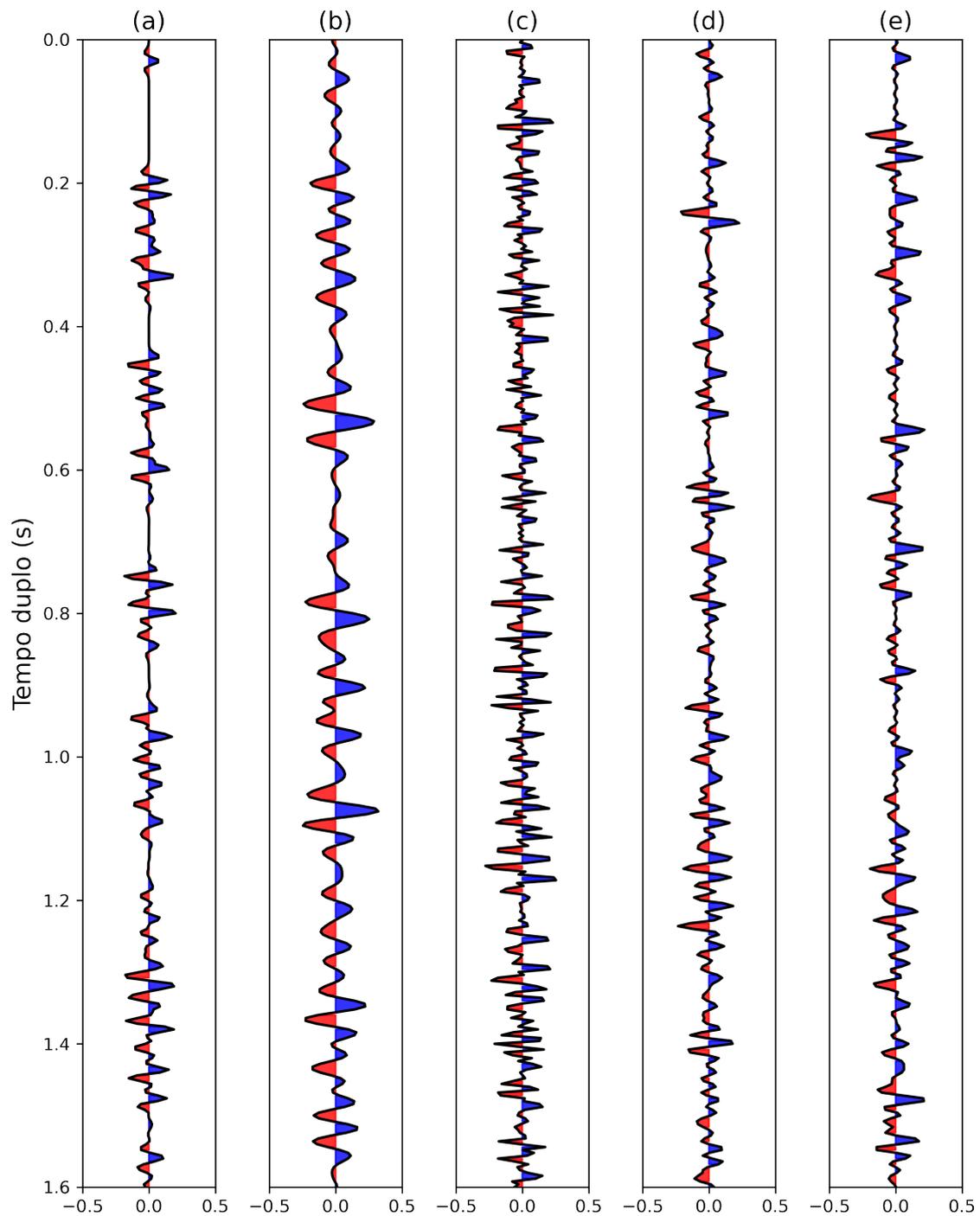


Figura 4.6: Traços sísmicos dos resultados apresentados para as *wavelets*: (a) Ricker (Figura 4.1), (b) Gabor (Figura 4.2), (c) Ormsby (Figura 4.3), (d) Klauder (Figura 4.4), e (e) Sinc (Figura 4.5).

serão os mesmos utilizados na seção anterior, porém foi adicionado um ruído normal ao seu conteúdo de amplitude. As Figuras 4.7, 4.8, 4.9, 4.10 e 4.11, apresentam as *wavelets* observadas e calculadas (Subfiguras (a) e (c)), o resíduo entre elas (Subfigura (e)) e seus respectivos espectros de frequência (Subfiguras (b), (d) e (f)). Os traços sísmicos sintéticos utilizados na predição foram os mesmos utilizados para os resultados anteriores, porém, foi adicionado um ruído normal no seu conteúdo de amplitude. A Figura 4.12 apresenta os traços sísmicos sintéticos utilizados na predição das *wavelets* apresentadas nessa seção de resultados.

Para a *wavelet* Ricker com frequência dominante de 33 Hz, assim como no primeiro treinamento, o pulso sísmico predito (Figura 4.7 (c)) manteve as características principais do lobo central, apresentando uma pequena variação no pico da wavelet. Além disso, seus lobos laterais se apresentam levemente ruidosos e deformados. O espectro de frequência da *wavelet* predita (Figura 4.7 (d)), assim como no caso sem ruído, se assemelha a um espectro de *wavelet* Ormsby, visualmente mostrando uma tendência trapezoidal. Por meio do espectro do resíduo (Figura 4.7 (f)) vê-se que o espectro calculado adiciona um conteúdo de frequências de 10 a 30 Hz e entre 45 a 65 Hz, por outro lado, nos intervalos entre 30 a 45 Hz e de 55 a 70 Hz tem-se uma amplitude menor que a desejada. O coeficiente de correlação entre as *wavelets* calculada e observada é de 0,989.

A *wavelet* Gabor de frequência dominante de 21 Hz (Figura 4.8 (c)) apresenta um resultado que se assemelha a *wavelet* observada (Figura 4.8 (a)), com lobos laterais de menor amplitude e uma correlação de 0,993. No espectro de frequências calculado (Figura 4.8 (d)) é notável um acréscimo de conteúdo nas frequências iniciais, até aproximadamente 20 Hz, e um decréscimo de 20 a 40 Hz, entretanto, o espectro calculado aparenta ser muito semelhante ao espectro da wavelet observada (Figura 4.8 (b)).

Para a predição a partir de uma *wavelet* Ormsby, vê-se que o lobo central do sinal calculado (Figura 4.9 (c)) possui menor amplitude que o observado (Figura 4.9 (a)), além disso, as amplitudes laterais negativas foram evidenciadas na *wavelet* calculada e não há as variações laterais vistas na observada. O espectro de frequências calculado (Figura 4.9 (d)) possui um direcionamento curvo, ao invés do trapezoidal visto no espectro observado (Figura 4.9 (b)), tendo um menor conteúdo de frequências entre 10 a 35 Hz e entre 70 a 115 Hz. O coeficiente de correlação entre as *wavelets* observada e calculado é de 0,944.

A *wavelet* predita para a Klauder (Figura 4.10 (c)) conseguiu estimar de forma satisfatória o lobo central, apresentando um desvio e sendo mais estreita em relação a observada. Por outro lado, a variação dos lobos laterais não foi reproduzida e a correlação entre as *wavelets* foi de 0,952. O espectro de frequências calculado (Figura 4.10 (d)) se assemelha a um espectro Ormsby, tendo maior amplitude na região central entre 25 a 50 Hz e de 60 a

80 Hz e menor amplitude nas laterais, de 0 a 25 Hz e de 50 a 60 Hz.

Por último, a predição da *wavelet* Sinc predita (Figura 4.11 (c)) se assemelha ao sinal observado (Figura 4.11 (a)) no lobo central, no entanto, as variações laterais apresentam menor amplitude ou são inexistentes na calculada, outrossim, o coeficiente de correlação entre as duas *wavelets* é de 0,987. Para o espectro de frequências, tem-se que o espectro calculado (Figura 4.11 (d)) tenta apresentar uma tendência linear, assim como acontece com o observado (Figura 4.11 (b)), mas possui variações de amplitude de forma que, em sua maioria, possui um conteúdo de frequências maior q o observado. Junto a isso, o espectro calculado apresenta uma queda a partir da frequência de 45 Hz, enquanto que, para o observado, isso ocorre em, aproximadamente, 55 Hz.

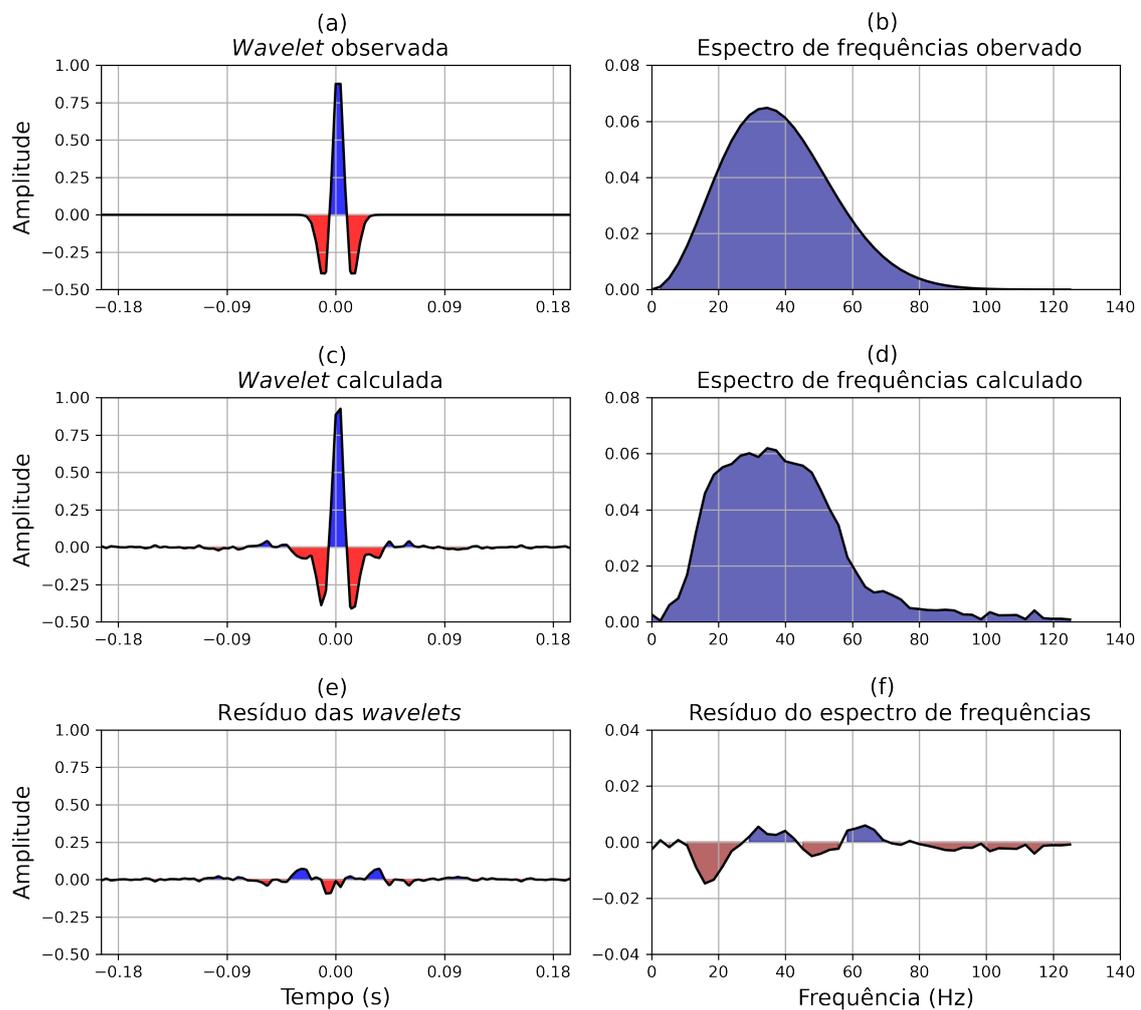


Figura 4.7: Resultado da predição de uma *wavelet* Ricker pela segunda RNA: (a), (c) e (e) são as *wavelets* observada, calculada e o resíduo; enquanto (b), (d) e (f) são os respectivos espectros de frequência. O coeficiente de correlação entre as *wavelets* observada e calcula é de 0,989.

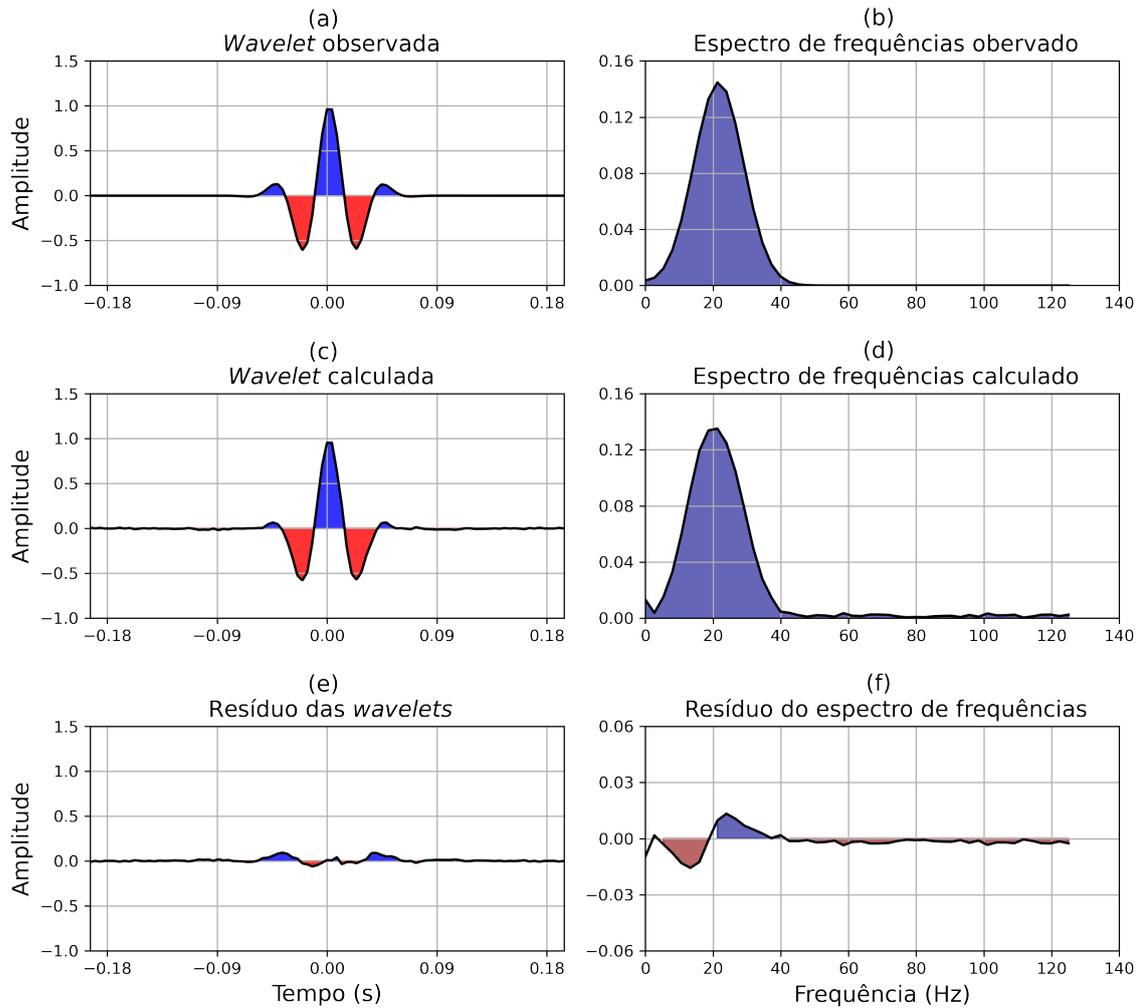


Figura 4.8: Resultado da predição de uma *wavelet* Gabor pela segunda RNA: (a), (c) e (e) são as *wavelets* observada, calculada e o resíduo; enquanto (b), (d) e (f) são os respectivos espectros de frequência. O coeficiente de correlação entre as *wavelets* observada e calcula é de 0,993.

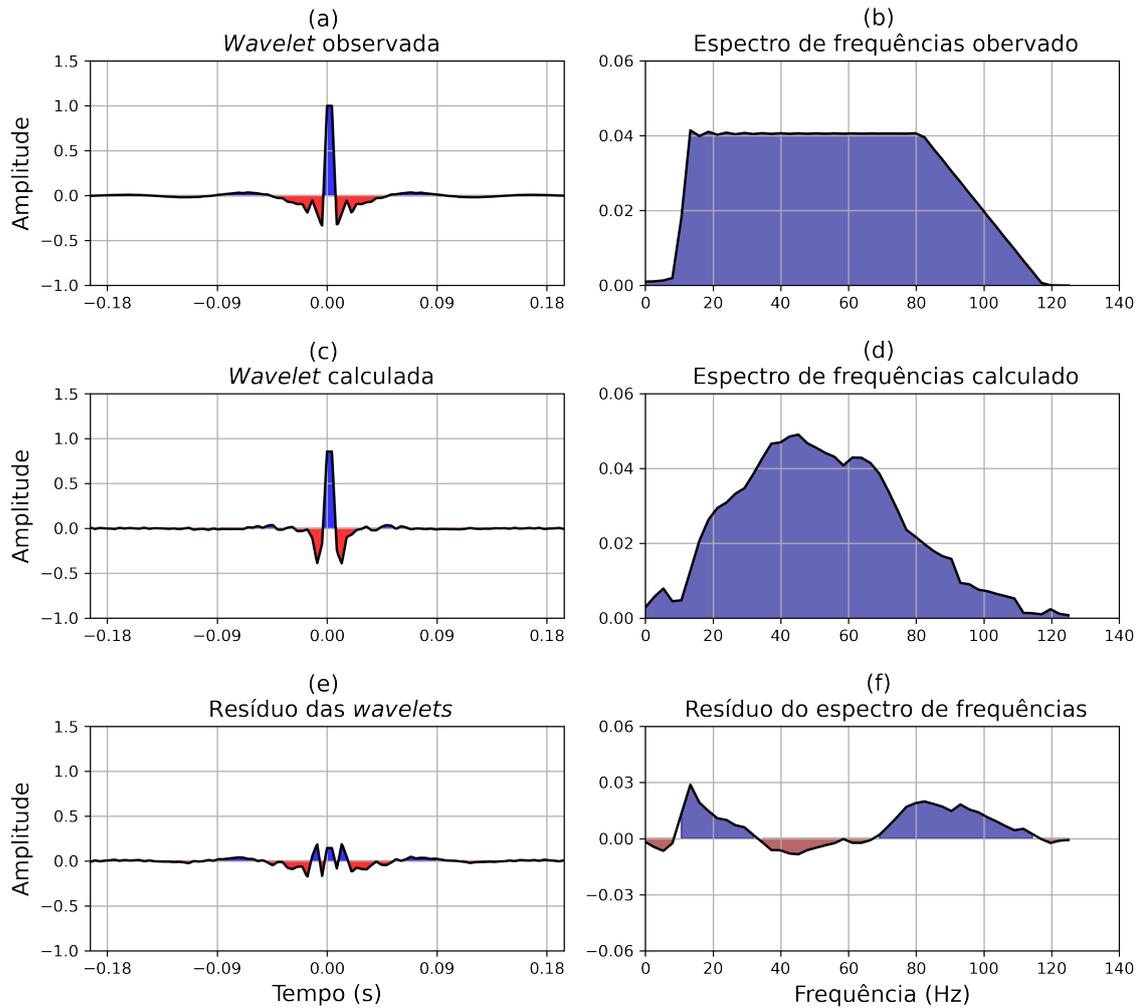


Figura 4.9: Resultado da predição de uma *wavelet* Ormsby pela segunda RNA: (a), (c) e (e) são as *wavelets* observada, calculada e o resíduo; enquanto (b), (d) e (f) são os respectivos espectros de frequência. O coeficiente de correlação entre as *wavelets* observada e calcula é de 0,944.

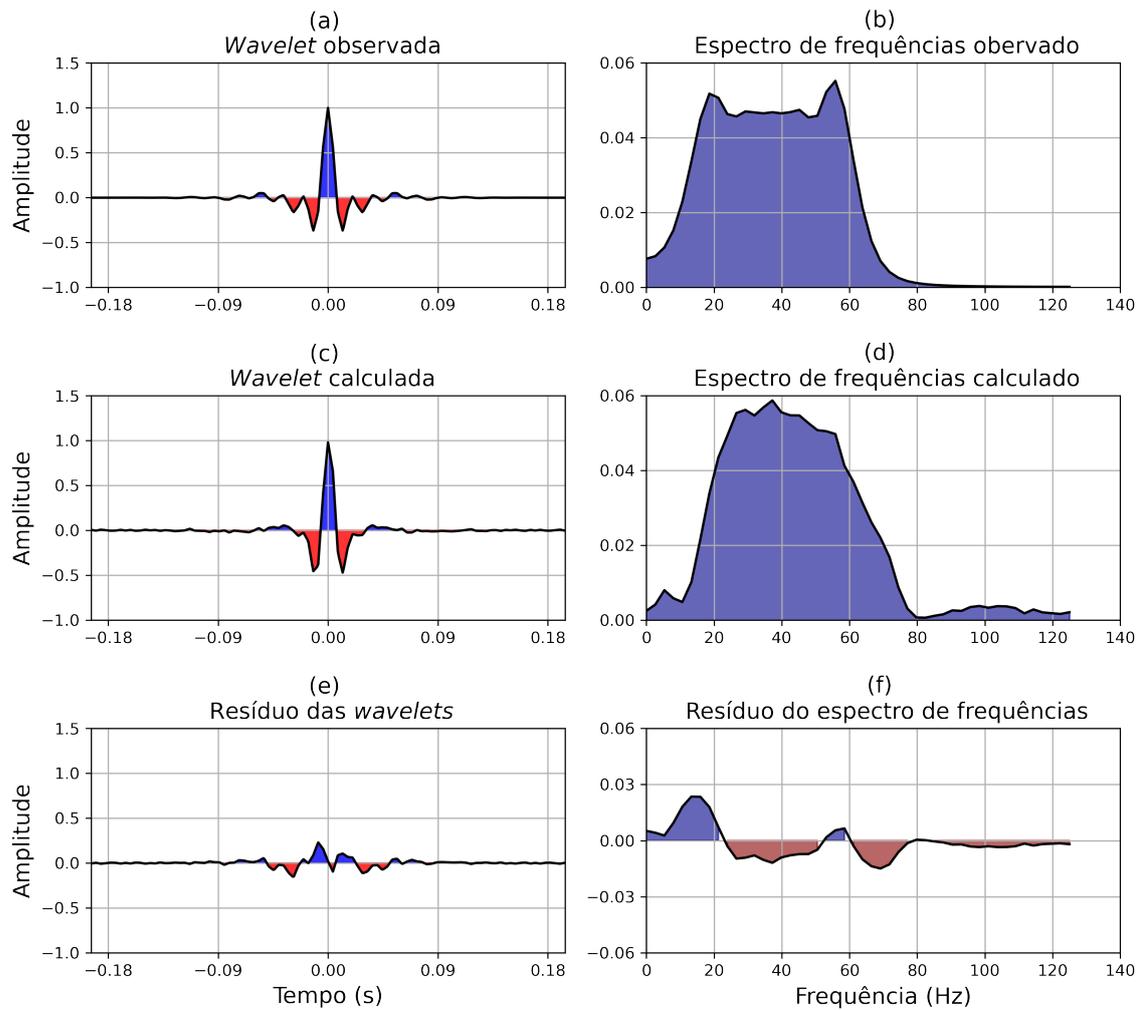


Figura 4.10: Resultado da predição de uma *wavelet* Klauder pela segunda RNA: (a), (c) e (e) são as *wavelets* observada, calculada e o resíduo; enquanto (b), (d) e (f) são os respectivos espectros de frequência. O coeficiente de correlação entre as *wavelets* observada e calcula é de 0,952.

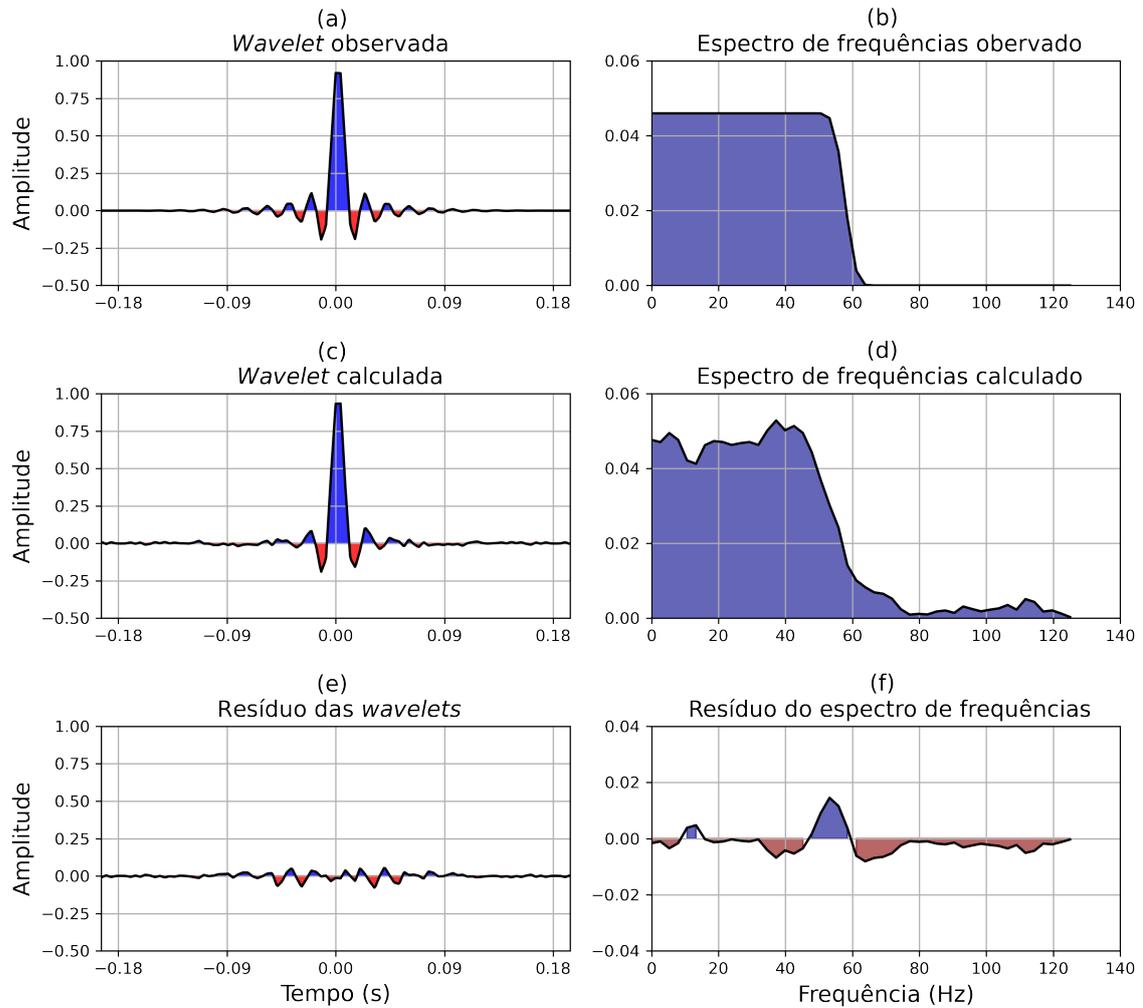


Figura 4.11: Resultado da predição de uma *wavelet* Sinc pela segunda RNA: (a), (c) e (e) são as *wavelets* observada, calculada e o resíduo; enquanto (b), (d) e (f) são os respectivos espectros de frequência. O coeficiente de correlação entre as *wavelets* observada e calcula é de 0,987.

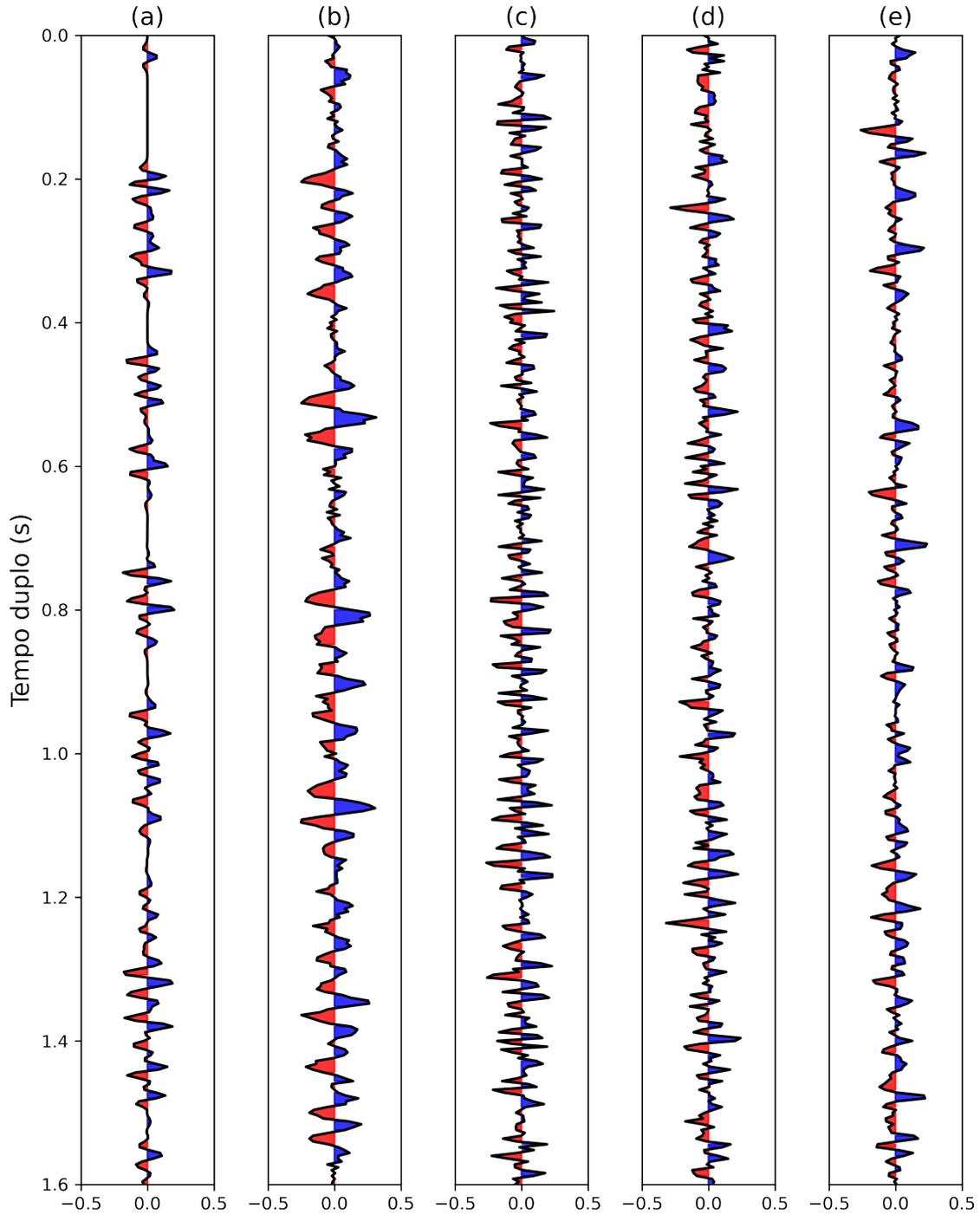


Figura 4.12: Traços sísmicos ruidosos utilizados na predição dos resultados apresentados para as *wavelets*: (a) Ricker (Figura 4.7) com 0% de ruído, (b) Gabor (Figura 4.8) com 10% de ruído, (c) Ormsby (Figura 4.9) com 10% de ruído, (d) Klauder (Figura 4.10) com 10% de ruído, e (e) Sinc (Figura 4.11) com 10% de ruído.

4.3 Comparações com o método de autocorrelação

Para verificar a viabilidade dos resultados obtidos no treinamento da rede neural Perceptron multicamadas frente aos métodos convencionais de extração de *wavelets* mais populares, foram feitas comparações entre os sinais observados, as estimativas da RNA e, também, as *wavelets* calculadas através o método da autocorrelação. Para tal comparação, apenas foram considerados os resultados obtidos na segunda etapa do treinamento da RNA — com a adição do ruído normal no conjunto de dados — e, da mesma forma, a autocorrelação foi feita utilizando os mesmos traços sísmicos sintéticos (Figura 4.12). As Figuras 4.13, 4.14, 4.15, 4.16 e 4.17 apresentam os resultados, de forma que são mostradas as comparações entre as *wavelets* calculada pela RNA e observada nas subfiguras (a), as comparações entre as *wavelets* calculada pelo método da autocorrelação e observada nas subfiguras (b), as *wavelets* observada e calculadas (subfiguras (c),(e) e (g)) e seus respectivos espectros de frequência (subfiguras (d), (f) e (h)).

Os resultados obtidos pela autocorrelação para os pulsos sísmicos mostram que o lobo central consegue ser calculado de forma satisfatória, apresentando um pico acentuado em todos os casos. Entretanto, as *wavelets* possuem grandes variações nos lobos laterais, prejudicando sua similaridade com o sinal observado quando comparado aos pulsos sísmicos preditos pela RNA.

Além disso, é possível notar que os espectros de frequência calculados pela autocorrelação, em geral, possuem amplitudes pontualmente maiores e mal distribuídas em relação aos espectros desejados, possuindo uma disposição de se apresentar em picos de conteúdos de alta amplitude por todo o espectro de frequência, diferentemente dos espectros da rede neural que tentam simular o espectro de frequências desejado. Ainda assim, o espectro de frequências da autocorrelação acompanha o formato e o conteúdo de frequências do espectro observado.

As Tabelas 4.1 e 4.2 apresentam os resultados mínimo, máximo e médio dos coeficientes de correlação de Pearson obtidos pela predição das *wavelets* utilizando a RNA e o método da autocorrelação para o conjunto de treino e teste, respectivamente. As médias de treino e teste mostram que a rede neural foi capaz de estimar as *wavelets* de forma satisfatória sem apresentar *overfitting*, ou seja, manteve os resultados similares tanto para o conjunto de treinamento quanto para o conjunto de teste. Em sua maioria, os resultados da RNA com os menores valores do coeficiente de correlação se encontram próximos de zero, enquanto os resultados com os maiores valores de coeficiente de correlação se encontram próximos de 1. Os valores mínimos representam casos pontuais onde a rede neural não foi capaz de prever a *wavelet* corretamente, sendo que coeficientes de correlação abaixo de 0,5 correspondem a 2,5% do conjunto de treino e 4,2% do conjunto de teste. Por outro lado, na autocorrelação,

resultados cujo coeficiente de correlação está abaixo de 0,5 simbolizam 5,5% e 5,3% dos conjuntos de treino e teste, respectivamente. Da mesma forma, predições da RNA com coeficiente de correlação acima de 0,8 representam 83,7% do conjunto de treinamento e 77,3% do conjunto de teste, já para a autocorrelação, estimativas com coeficiente de Pearson maior que 0,8 compõem cerca de 46,3% e 46,7% do conjunto de treino e teste, respectivamente.

Da mesma forma, a Tabela 4.3 apresenta a média obtida pelos coeficientes de correlação calculados utilizando todo o conjunto de dados entre as *wavelets* observada e calculada pelo treinamento da segunda RNA — com adição de ruído — e, entre as *wavelets* observada e calculada pela autocorrelação utilizando o traço ruidoso. A correlação média da rede neural Perceptron multicamadas é de 0,89, enquanto a média apresentada para a autocorrelação é de 0,76, o que mostra que a rede é mais eficiente para as *wavelets* sísmicas. A autocorrelação teve um valor médio superior para a *wavelet* Klauder. A média obtida para as *wavelets* do tipo Ormsby pela autocorrelação pode ter sido influenciada pela característica desse sinal utilizar um conjunto de frequências e possuir um espectro com tendência de forma quadrada ou trapezoidal. Essa característica pode não ser estimada com eficiência pela autocorrelação, já que o método possui propensão a apresentar picos no espectro, o que deve excluir importantes conteúdos de frequência na faixa central da Ormsby. Os resultados mostram que a rede neural consegue estimar a *wavelet* e o espectro de frequência com relativa coerência.

		Coeficiente de correlação de Pearson					
		RNA com ruído			Autocorrelação		
<i>Wavelets</i>	Nº de traços	Média	Mín.	Máx.	Média	Mín.	Máx.
Ricker	11.171	0,9405	0,0040	0,9995	0,7721	0,5159	0,9915
Gabor	11.175	0,9441	-0,0088	0,9995	0,7895	0,5315	0,9951
Ormsby	11.220	0,8607	-0,0008	0,9965	0,5754	0,1303	0,9041
Klauder	11.103	0,7871	-0,3614	0,9933	0,8774	0,6468	0,9812
Sinc	11.331	0,9508	0,2329	0,9997	0,7984	0,4053	0,9980
Total	56.000	0,8913	-0,3614	0,9997	0,7622	0,1303	0,9980

Tabela 4.1: Coeficientes de correlação de Pearson médio, mínimo e máximo dos resultados obtidos pela predição de wavelets utilizando a RNA com ruído e o método da autocorrelação para cada *wavelet* e para o conjunto total de treinamento.

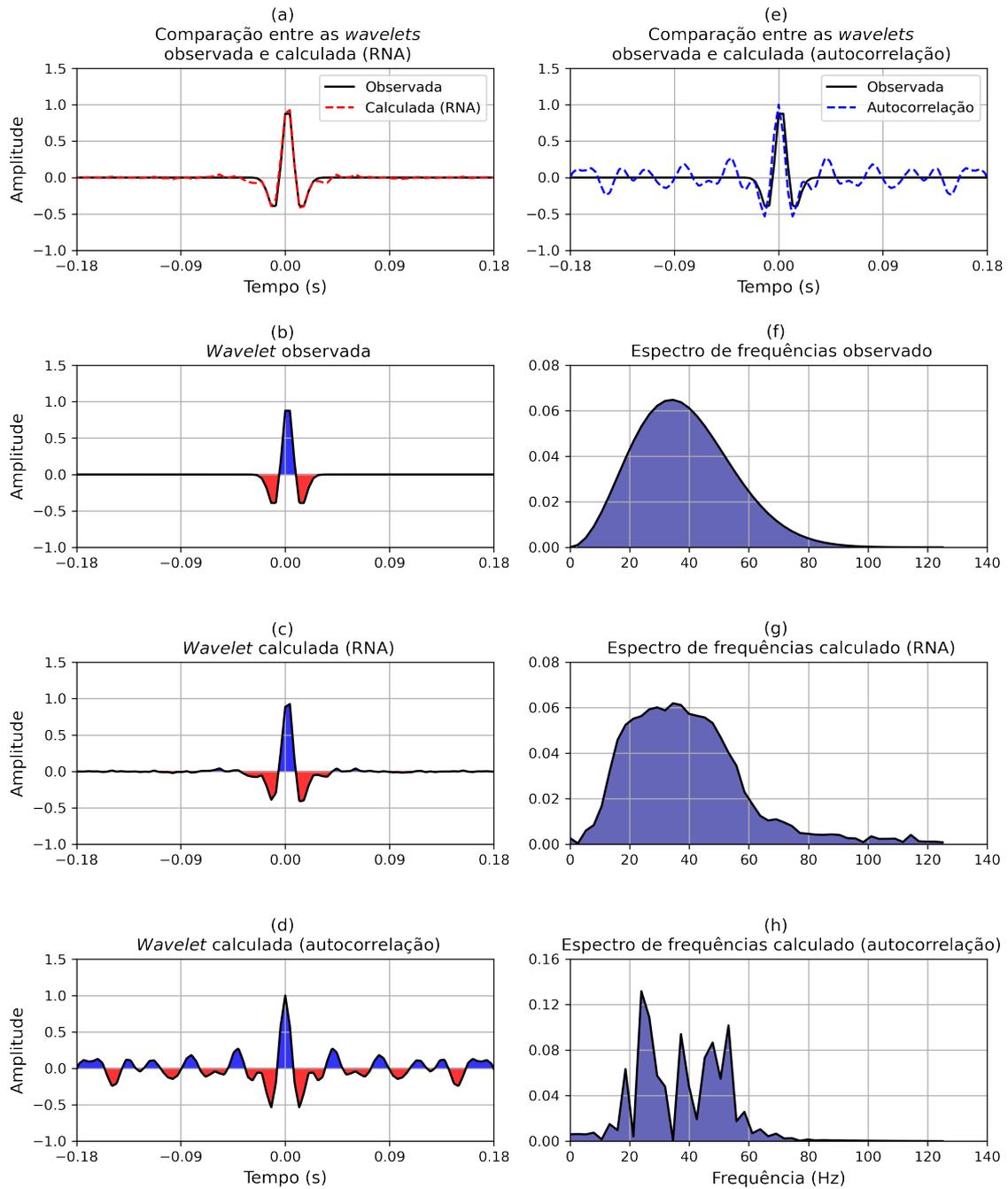


Figura 4.13: Comparação de resultados de uma *wavelet* sísmica do tipo Ricker: (a) Comparação entre a *wavelet* observada e calculada utilizando a rede neural Perceptron multicamadas, (b) *wavelet* sísmica observada, (c) *wavelet* sísmica extraída a partir da RNA, (d) *wavelet* sísmica extraída a partir da técnica de autocorrelação, (e) Comparação entre a *wavelet* observada e calculada através o método de autocorrelação, (f) espectro de frequência da *wavelet* observada, (g) espectro de frequência da *wavelet* calculada pela RNA e (h) espectro de frequência calculada a partir da técnica de autocorrelação.

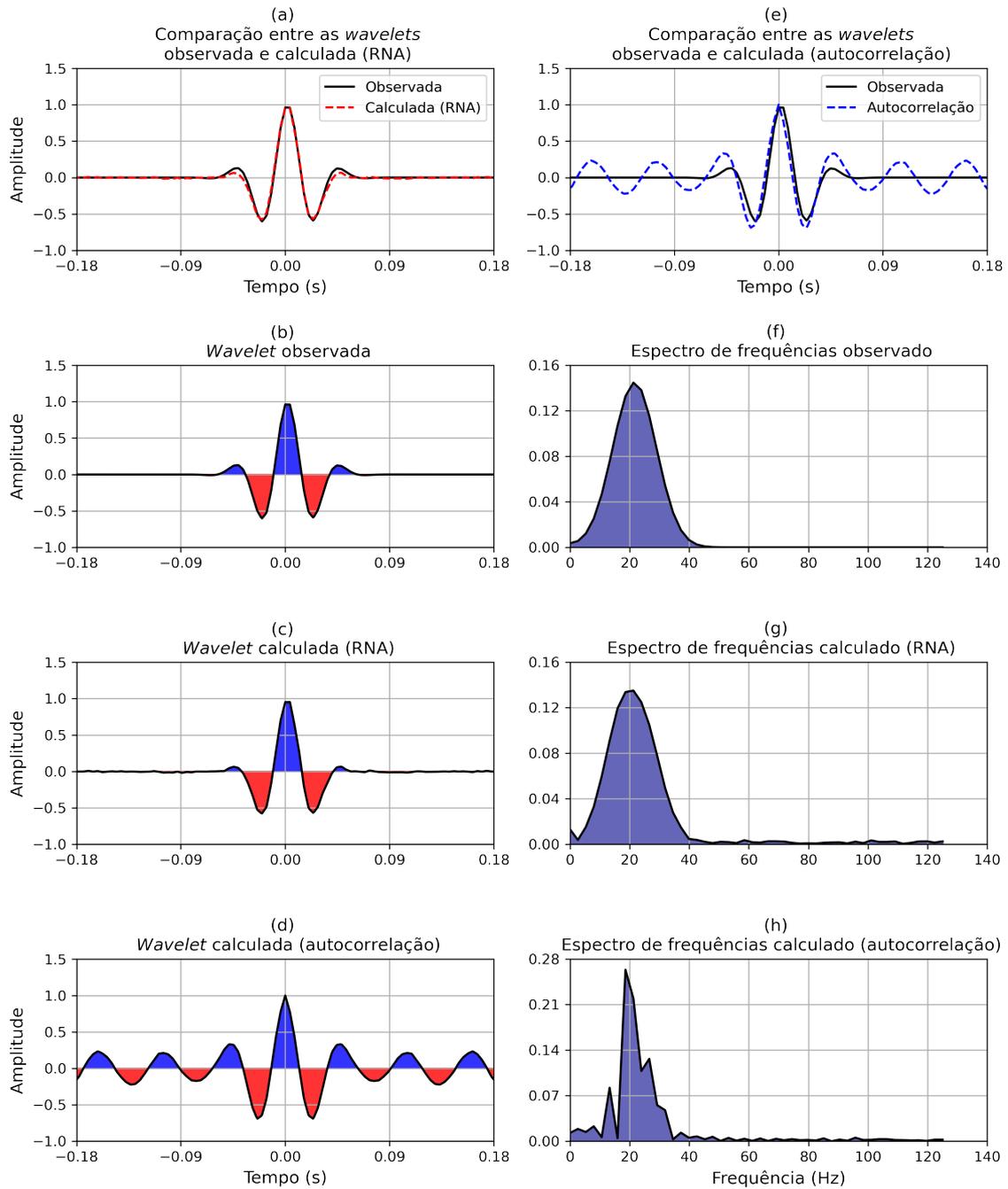


Figura 4.14: Comparação de resultados de uma *wavelet* sísmica do tipo Gabor: (a) Comparação entre a *wavelet* observada e calculada utilizando a rede neural Perceptron multicamadas, (b) *wavelet* sísmica observada, (c) *wavelet* sísmica extraída a partir da RNA, (d) *wavelet* sísmica extraída a partir da técnica de autocorrelação, (e) Comparação entre a *wavelet* observada e calculada através o método de autocorrelação, (f) espectro de frequência da *wavelet* observada, (g) espectro de frequência da *wavelet* calculada pela RNA e (h) espectro de frequência calculada a partir da técnica de autocorrelação.

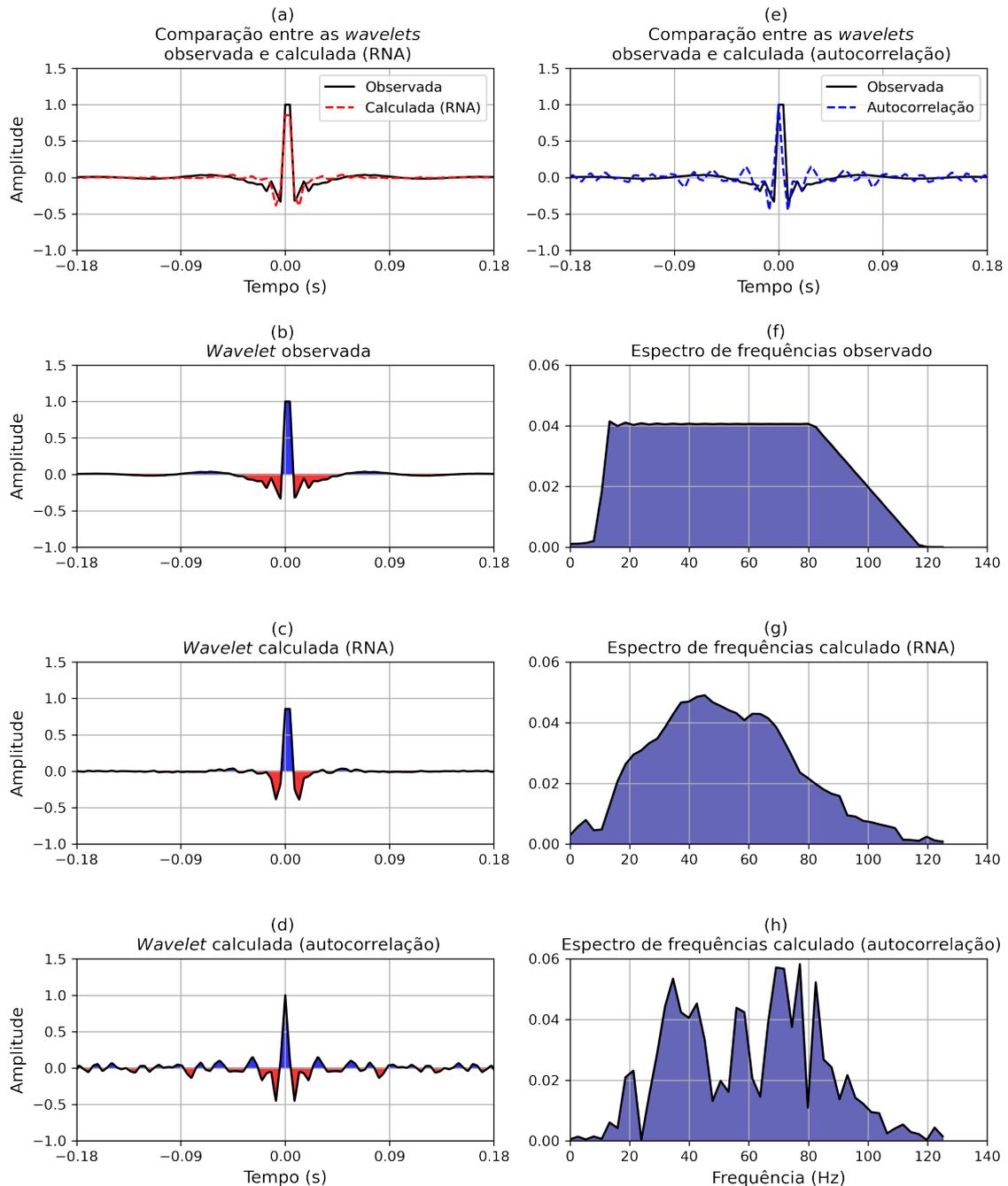


Figura 4.15: Comparação de resultados de uma *wavelet* sísmica do tipo Ormsby: (a) Comparação entre a *wavelet* observada e calculada utilizando a rede neural Perceptron multicamadas, (b) *wavelet* sísmica observada, (c) *wavelet* sísmica extraída a partir da RNA, (d) *wavelet* sísmica extraída a partir da técnica de autocorrelação, (e) Comparação entre a *wavelet* observada e calculada através o método de autocorrelação, (f) espectro de frequência da *wavelet* observada, (g) espectro de frequência da *wavelet* calculada pela RNA e (h) espectro de frequência calculada a partir da técnica de autocorrelação.

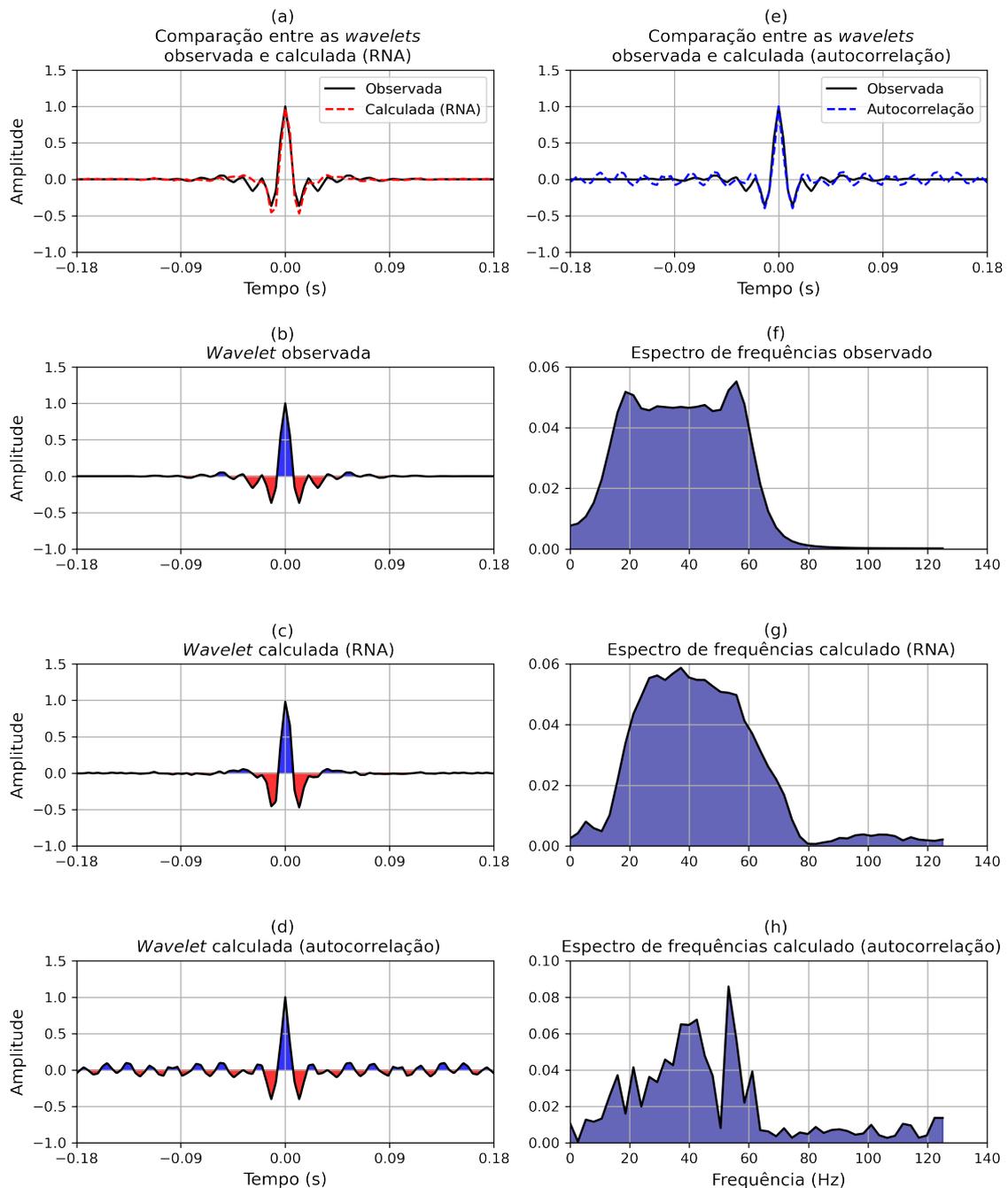


Figura 4.16: Comparação de resultados de uma *wavelet* sísmica do tipo Klaunder: (a) Comparação entre a *wavelet* observada e calculada utilizando a rede neural Perceptron multicamadas, (b) *wavelet* sísmica observada, (c) *wavelet* sísmica extraída a partir da RNA, (d) *wavelet* sísmica extraída a partir da técnica de autocorrelação, (e) Comparação entre a *wavelet* observada e calculada através o método de autocorrelação, (f) espectro de frequência da *wavelet* observada, (g) espectro de frequência da *wavelet* calculada pela RNA e (h) espectro de frequência calculada a partir da técnica de autocorrelação.

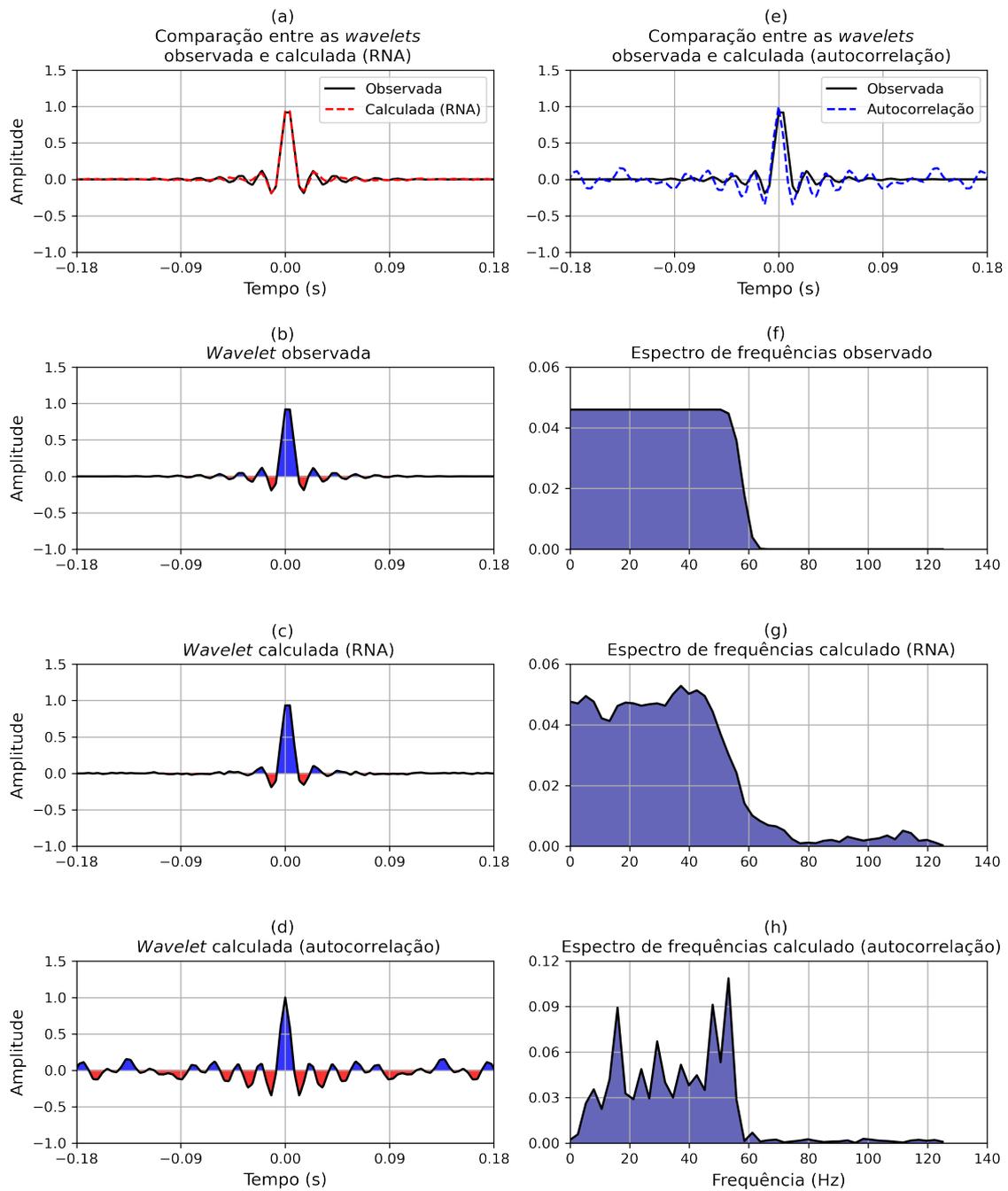


Figura 4.17: Comparação de resultados de uma *wavelet* sísmica do tipo Sinc: (a) Comparação entre a *wavelet* observada e calculada utilizando a rede neural Perceptron multicamadas, (b) *wavelet* sísmica observada, (c) *wavelet* sísmica extraída a partir da RNA, (d) *wavelet* sísmica extraída a partir da técnica de autocorrelação, (e) Comparação entre a *wavelet* observada e calculada através o método de autocorrelação, (f) espectro de frequência da *wavelet* observada, (g) espectro de frequência da *wavelet* calculada pela RNA e (h) espectro de frequência calculada a partir da técnica de autocorrelação.

		Coeficiente de correlação de Pearson					
		RNA com ruído			Autocorrelação		
<i>Wavelets</i>	Nº de traços	Média	Mín.	Máx.	Média	Mín.	Máx.
Ricker	2736	0,9284	0,0815	0,9994	0,7741	0,5471	0,9898
Gabor	2765	0,9323	0,2221	0,9994	0,7889	0,5555	0,9932
Ormsby	2818	0,8153	0,0332	0,9953	0,5759	0,1899	0,8882
Klauder	2891	0,7289	0,0007	0,9895	0,8761	0,7043	0,9837
Sinc	2790	0,9408	0,3214	0,9997	0,7976	0,5435	0,9986
Total	14.000	0,8899	0,0007	0,9997	0,7633	0,1899	0,9986

Tabela 4.2: Coeficientes de correlação de Pearson médio, mínimo e máximo dos resultados obtidos pela predição de wavelets utilizando a RNA com ruído e o método da autocorrelação para cada *wavelet* e para o conjunto total de teste.

		Coeficiente de correlação médio		
Wavelet	Nº de traços	RNA sem ruído	RNA com ruído	Autocorrelação
Ricker	13.907	0,9862	0,9385	0,7722
Gabor	13.940	0,9885	0,9419	0,7893
Ormsby	14.038	0,9547	0,8517	0,5755
Klauder	13.994	0,9390	0,7760	0,8766
Sinc	14.121	0,9935	0,9494	0,7982
Total	70.000	0,9724	0,8911	0,7624

Tabela 4.3: Desempenho dos resultados da rede neural para predição de *wavelets* em traços sintéticos sem ruídos, da rede neural que extrai *wavelets* sísmicas considerando o ruído e da autocorrelação, também considerando os traços ruidosos, para cada uma das *wavelets* e para todo o conjunto de dados.

Capítulo 5

Conclusões

Neste trabalho foi desenvolvida e treinada uma rede neural Perceptron multicamadas para extrair *wavelets* sísmicas a partir de traços sísmicos sintéticos. Para realizar tal atividade, um conjunto de 70.000 traços sintéticos foram gerados e utilizado no treinamento das duas RNAs, uma que teve como conjunto de treinamento os dados sem ruído e outra com diferentes porcentagens de ruído normais distribuídos pelo traços sísmico para testar a estabilidade da rede neural.

No primeiro treinamento, a rede neural mostrou-se capaz de fazer a predição de *wavelets* sísmicas em traços sísmicos sintéticos sem ruídos. Após realizar diversos treinamentos, chegou-se a conclusão que os melhores hiperparâmetros para o treinamento da rede foram de 0,001 para a taxa de aprendizado (α), o número de *batches* igual a 64 e o número de épocas igual a 5000. Os valores do cosseno hiperbólico (função custo) e da similaridade dos cossenos foram de $4,95 \times 10^{-4}$ e 0,988 para o treino e $3,5 \times 10^{-3}$ e 0,905 para o teste, respectivamente.

É possível observar que a RNA treinada com dados sem adição de ruídos apresenta uma correlação média entre *wavelets* calculadas e observadas é de aproximadamente 0,9724, e apresenta seu melhor desempenho na predição de *wavelets* do tipo Gabor, com correlação média igual a 0,9885, e seu pior desempenho na predição de *wavelets* do tipo Klauder, com correlação média de 0,9390. Além disso, é possível notar nos espectros de frequência das *wavelets* calculadas que, em geral, pouco conteúdo de frequência é perdido ou adicionado nas *wavelets* preditas pela RNA.

No segundo treinamento, após adicionar ruídos normais em alguns traços sísmicos sintéticos, é possível notar que não há grande melhorias no aumento do número de épocas ao se importar os coeficientes da rede neural treinada com um conjunto de dados que não tinham ruídos adicionados no traços sísmicos, e os valores da função custo e da similaridade dos cossenos foram de $7,8 \times 10^{-4}$ e 0,981 para o treino e $4,9 \times 10^{-3}$ e 0,867 para o teste,

respectivamente. A rede neural teve uma correlação média entre as *wavelets* observadas e calculadas de aproximadamente 0,8911, e apresentou um melhor desempenho na predição das *wavelets* Sinc — aproximadamente 0,9494 — e o pior desempenho na predição das *wavelets* do tipo Klauder (aproximadamente 0,7760).

Por fim, por se tratar da predição de *wavelets* sísmicas de fase zero, foram extraídas *wavelets* nos traços sísmicos ruidosos utilizando o método de autocorrelação. Em geral, a rede neural mostrou um valor de correlação entre a *wavelet* calculada e observada superior ao método de correlação — que teve um valor médio igual a 0,7624 —, mostrando-se ser mais eficaz na determinação de pulsos sísmicos em dados sintéticos com adição de ruído normal. Exceto para *wavelet* do tipo Klauder, o desempenho da RNA foi muito superior que os resultados obtidos pela técnica de autocorrelação do traço sísmico.

Por fim, para trabalhos futuros, pretende-se aprimorar os resultados obtidos apartir da rede neural treinada neste trabalho, de forma a contemplar *wavelets* sísmicas de fase mínima e/ou de fase mista. Além disso, mais ajustes serão realizados para aperfeiçoar os resultados da rede treinada com dados ruidosos. Comparações com outros métodos de estimativa serão feitas para consolidar os resultados obtidos pela rede neural e espera-se utilizar RNA em aplicações envolvendo dados sísmicos reais - tanto para o processamento quanto para a interpretação de dados sísmicos.

Agradecimentos

Em primeiro lugar, agradeço aos meus pais, Luciano Miranda dos Santos e Cristiane da Silva Oliveira Santos, por serem meus principais apoiadores e incentivadores, e por nunca medirem esforços para me ajudar a alcançar o que desejo. Agradeço por todo o seu amor incondicional.

Agradeço a toda minha família. Em especial, minha avó Joana, meu tio Lourival e minha tia Alessandra, que partiram esse ano e deixam muita saudade, mas me motivaram a nunca desistir dos meus objetivos.

À minhas amigas Vaneska e Maria Mariana que me acompanharam ao longo da minha graduação, por me apoiarem e estarem sempre dispostas a escutar minhas lamentações. À todos os meus amigos da época do colégio e UFBA.

Agradeço, também, ao meu orientador Alexsandro Guerra Cerqueira por todas as palavras de incentivo e por sempre estar disposto a me atender e ajudar em tudo que eu precisasse.

Aos meus colegas de 2017.1 e do grupo GAIA.

Aos professores do curso de geofísica, Milton Porsani, Wilson Figueiró, Joelson Batista, Suzan Vasconcelos, Susana Cavalcanti, Marcos Vasconcelos, Michelângelo Gomes e todos os outros, pelo conhecimento compartilhado e pelas aulas ofertadas. Em especial, ao Prof. Hedison Sato por todo o apoio durante o curso.

Por fim, agradeço a todos que contribuíram e me apoiaram ao longo de minha jornada na UFBA.

Apêndice A

Tabelas de resultados dos treinamentos da rede neural

α	Treino		Teste		Tempo (min)
	Custo	Métrica	Custo	Métrica	
1×10^{-4}	$2,5608 \times 10^{-4}$	0,9955	0,0042	0,8905	15
5×10^{-4}	$2,0541 \times 10^{-4}$	0,9971	0,0040	0,8936	15
1×10^{-3}	$2,3147 \times 10^{-4}$	0,9963	0,0038	0,8965	15
5×10^{-3}	0,0022	0,9385	0,0042	0,8816	15
1×10^{-2}	0,0046	0,8758	0,0054	0,8509	15
5×10^{-2}	0,0325	0,5074	0,0330	0,4965	15
1×10^{-1}	0,1656	0,2046	0,1656	0,2027	15
5×10^{-1}	0,4268	0,1190	0,4269	0,1182	15
1	0,4281	0,1118	0,4283	0,1104	16

Tabela A.1: Treinamento da primeira da rede neural Perceptron multicamadas (sem ruídos) para obtenção do melhor valor para a taxa de aprendizado (α). São apresentados os resultados para a função custo (cosseno hiperbólico) e métrica (similaridade dos cossenos), além do tempo de processamento gasto em cada um dos treinamentos. Os valores de *batch* e épocas foram de 256 e 1000.

Tamanho de <i>batch</i>	Treino		Teste		Tempo (min)
	Custo	Métrica	Custo	Métrica	
16	$9,7085 \times 10^{-4}$	0,9765	0,0035	0,9073	149
32	$7,2005 \times 10^{-4}$	0,9833	0,0035	0,9050	85
64	$3,5508 \times 10^{-4}$	0,9925	0,0039	0,8917	40
128	$2,8942 \times 10^{-4}$	0,9945	0,0039	0,8920	28
256	$2,3147 \times 10^{-4}$	0,9963	0,0038	0,8965	15
512	$2,0055 \times 10^{-4}$	0,9972	0,0038	0,8953	7
1024	$1,8066 \times 10^{-4}$	0,9977	0,0040	0,8891	4

Tabela A.2: Treinamento da primeira da rede neural Perceptron multicamadas (sem ruídos) para obtenção do melhor valor para o número de (*batchs*). São apresentados os resultados para a função custo (cosseno hiperbólico) e métrica (similaridade dos cossenos), além do tempo de processamento gasto em cada um dos treinamentos. Os valores de α e de épocas foram de 0.001 e 1000.

N° de épocas	Treino		Teste		Tempo (min)
	Custo	Métrica	Custo	Métrica	
1000	$3,5508 \times 10^{-4}$	0,9925	0,0039	0,8917	40
2500	$5,3089 \times 10^{-4}$	0,9883	0,0035	0,9046	161
5000	$4,9485 \times 10^{-4}$	0,9892	0,0035	0,9044	320
7500	$4,5677 \times 10^{-4}$	0,9903	0,0036	0,9028	481
10.000	$4,6426 \times 10^{-4}$	0,9900	0,0036	0,9027	632

Tabela A.3: Treinamento da primeira da rede neural Perceptron multicamadas (sem ruídos) para obtenção do melhor valor para o número de épocas. São apresentados os resultados para a função custo (cosseno hiperbólico) e métrica (similaridade dos cossenos), além do tempo de processamento gasto em cada um dos treinamentos. Os valores de α e do número de *batchs* foram de 0.001 e 64.

N° de épocas	Treino		Teste		Tempo (min)
	Custo	Métrica	Custo	Métrica	
1000	$9,3786 \times 10^{-4}$	0,9769	0,0049	0,8649	34
2500	$8,5007 \times 10^{-4}$	0,9792	0,0050	0,8646	88
5000	$7,7990 \times 10^{-4}$	0,9808	0,0049	0,8666	177
7500	$7,1655 \times 10^{-4}$	0,9827	0,0049	0,8642	263
10.000	$7,2280 \times 10^{-4}$	0,9824	0,0049	0,8662	349

Tabela A.4: Treinamento da segunda rede neural Perceptron multicamadas (com ruídos) para obtenção do melhor valor para o número de épocas. Foram mantidos os valores para o α e o tamanho de *batch* de melhor desempenho do treinamento da RNA com traços sísmicos sintéticos sem ruído. São apresentados os resultados para a função custo (cosseno hiperbólico) e métrica (similaridade dos cossenos), além do tempo de processamento gasto em cada um dos treinamentos.

Referências

- Acquaviva, A. A.; Nicácio, R.; Silva, E. e Portugal, B. (2015) Análise comparativa entre as wavelets ricker e extraída para amarração de poços, 14th International Congress of the Brazilian Geophysics Society & EXPOGEOF, pp. 1150–1154.
- Bianco, E. (2016) Tutorial: Wavelet estimation for well ties, *The Leading Edge*, **35**(6):541–543.
- Bishop, C. M. (2007) *Pattern recognition and machine learning*, Springer International Publishing.
- Braga, A. P.; Ludemir, T. B. e Carvalho, A. C. (2000) *Redes neurais artificiais: Teoria e aplicações*, LTC.
- Brown, R. L.; McElhattan, W. e Santiago, D. J. (1988) Wavelet estimation: An interpretative approach, *The Leading Edge*, **7**(12):16–19.
- Calderón-Macías, C.; Sen, M. K. e Stoffa, P. L. (2000) Artificial neural networks for parameter estimation in geophysics, *Geophysical Prospecting*, **48**(1):21–47.
- Chollet, F. et al. (2015) Keras, <https://github.com/fchollet/keras>.
- Cui, T. e Margrave, G. F. (2014) Seismic wavelet estimation, *CREWES Research Report*, **26**:1–16.
- Deisenroth, M. P.; Faisal, A. A. e Ong, C. S. (2020) *Mathematics for machine learning*, Cambridge University Press.
- Dubois, M. K.; Bohling, G. C. e Chakrabarti, S. (2007) Comparison of four approaches to a rock facies classification problem, *Computers & Geosciences*, **33**(5):599–617.
- Goutorbe, B.; Lucazeau, F. e Bonneville, A. (2006) Using neural networks to predict thermal conductivity from geophysical well logs, *Geophysical Journal International*, **166**(1):115–125.
- Graupe, D. (2007) *Principles of artificial neural networks*, World Scientific, 2^o edic..
- Haykin, S. (2008) *Neural networks and learning machines*, Prentice Hall, 3^o edic..

-
- Hebb, D. O. (1949) *The organization of behavior: A neuropsychological theory*, Wiley.
- James, G.; Witten, D.; Hastie, T. e Tibshirani, R. (2013) *An introduction to statistical learning: with Applications in R*, Springer International Publishing.
- Ji, Z. e Yan, S. (2017) Properties of an improved gabor wavelet transform and its applications to seismic signal processing and interpretation, *Applied Geophysics*, **14**(4):529–542.
- Karim, R. (2018) 10 stochastic gradient descent optimisation algorithms cheatsheet, <https://towardsdatascience.com/10-gradient-descent-optimisation-algorithms-86989510b5e9>. Acessado em 15-set-2021.
- Kingma, D. P. e Ba, J. (2015) Adam: A method for stochastic optimization, *International Conference on Learning Representations*, pp. 1–15.
- Li, H.; Yang, W. e Yong, X. (2018) Deep learning for ground-roll noise attenuation, *SEG Technical Program Expanded Abstracts*, pp. 1981–1985.
- McCulloch, W. S. e Pitts, W. (1943) A logical calculus of the ideas immanent in nervous activity, *Bulletin of mathematical biophysics*, **5**:115–133.
- Mejri, M. e Bekara, M. (2020) Application of machine learning for the automation of the quality control of noise filtering processes in seismic data imaging, *Geosciences*, **10**(12).
- Minsky, M. e Papert, S. A. (1969) *Perceptrons: An introduction to computational geometry*, MIT Press.
- Morlet, J.; Arens, G.; Fourgeau, E. e Giard, D. (1982) Wave propagation and sampling theory—part II: Sampling theory and complex waves, *Geophysics*, **47**(2):222–236.
- Oldenburg, D. W.; Levy, S. e Whittall, K. P. (1981) Wavelet estimation and deconvolution, *Geophysics*, **46**(11):1528–1542.
- Peters, B.; Haber, E. e Granek, J. (2019) Neural networks for geophysicists and their application to seismic data interpretation, *The Leading Edge*, **38**(7):534–540.
- Poulton, M. M., ed. (2001) *Computational neural networks for geophysical data processing*, vol. 30, Pergamon.
- Poulton, M. M.; Sternberg, B. K. e Glass, C. E. (1992) Location of subsurface targets in geophysical data using neural networks, *Geophysics*, **57**(12):1534–1544.
- Rosa, A. L. R. (2018) *Análise do sinal sísmico*, Sociedade Brasileira de Geofísica.
- Rosenblatt, F. (1958) The perceptron: A probabilistic model for information storage and organization in the brain, *Psychological Review*, **65**(6):386–408.
- Ryan, H. (1994) Ricker, ormsby, klander, butterworth – a choice of wavelets, *CSEG Recorder*, **19**(7):8–9.

- da Silva, I. N.; Spatti, D. H.; Flauzino, R. A.; Liboni, L. H. e Alves, S. F. (2016) Artificial neural networks: A practical course, Springer International Publishing.
- Simm, R. e Bacon, M. (2014) Seismic amplitude: An interpreter's handbook, Cambridge University Press.
- Van Rossum, G. e Drake, F. L. (1995) Python tutorial, <https://www.python.org/>.
- Velasco, M. M. (2007) Redes neurais artificiais, Laboratório de Inteligência Computacional Aplicada, PUC, Rio de Janeiro.
- Wrona, T.; Pan, I.; Gawthorpe, R. L. e Fossen, H. (2018) Seismic facies analysis using machine learning, *Geophysics*, **83**(5):O83–O95.
- Yilmaz, O. (1987) Seismic data processing, Society of Exploration Geophysicists.
- Yilmaz, O. (2001) Seismic data analysis: Processing, inversion, and interpretation of seismic data, Society of Exploration Geophysicists.
- Zheng, Y.; Zhang, Q.; Yusifov, A. e Shi, Y. (2019) Applications of supervised deep learning for seismic interpretation and inversion, *The Leading Edge*, **38**(7):526–533.